Title of Case Study: Customer Churn Analysis

Case Problem

In this case study, you will investigate a telecommunication customer churn data set to predict a customer's situation (churn or not).

Data Set

Telco Customer Churn Dataset:
https://www.kaggle.com/blastchar/telco-customer-churn

The dataset is provided by IBM which has information on Telco consumers including the following:

- Customers who left within the last month – the column is called Churn.
- Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies.
- Customer account information – how long they had been a customer, contract, payment method, paperless billing, monthly charges, and total charges.
- Demographic info about customers – gender, age range, and if they have partners and dependents.

Each row represents a single customer, whereas the columns hold information regarding churn status, services, account and demographic data. The data is stored as a classic information feel thus it is easier to understand. Generally, it can be observed that it's easier for the company to retain current customers than to attract new ones leading to the churn analysis for this data in order to predict the customer who will be staying.

The dataset contains 7043 rows, 21 columns and 11 missing values in TotalCharges column. Tenure, MonthlyCharges and TotalCharges are in continuous variables where as Seniorcitizen is in int format while the rest are character. The ratio of females to males is equal, number of senior citizens is 1142, around 48% customers don't have multiple phone lines, 44% of the consumers use Fiber Optic while 34% use DSL, a whopping 50% of the customers don't have online security.

Methods and Models

The following methods and models have been discussed during OPIM 407 course:

Regression-based Forecasting

The method is used to predict the future by using regression today. There is a relationship between independent and dependent variables. Dependent variable is an output variable. Independents are inputs. You can predict the output by using input values. Independent variables are also named Predictive, they have some coefficients, by looking at the coefficients you comment the impact of the outputs, and then predictions about outputs. We have simple and multiple linear regression methods.
In simple linear regression the idea is to use a predictor in order to find out some average value of the responses.
The relationship is defined as: $y = a + bX + \epsilon$ where:

$$b = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$a = \bar{y} - b$$

While preparing the data, first we look at having a linear assumption. This is an assumption that the input and output relations are linear. At this point you might need to transform your data for making the relation linear between input and output.
Removing noise is another point to correct. In linear regression your input and output are assumed to be not noisy. One can consider to use the data cleaning operations, which is essential if you want to remove the outliers from the output variables.
Collinearity is another point to consider, in the data preparation process highly correlated variables should be removed, because when your input variables are highly correlated, linear regression will over-fit the data.

After data preparation and modeling in LR model representation part there are a couple of essential models to pay attention to, such as p-values and residuals. P-value is an indicator of if you can accept or reject the hypothesis. Residuals are basically testing the quality of the model fit.

In order to understand the models performance, we look at the R^2, which is an explanation of the proportion of the total variability. If R^2 is near 1, it means the model fits the data well. If R^2 is near 0, then the model poorly fits the data.

KNN

k-Nearest Neighbors Algorithm (KNN) is a supervised predictive algorithm that tries to identify k records from the training set that are similar to the target value we wish to predict. KNN can be used for both classification and regression problems, however, during the course, we used KNN for classification only. KNN is a lazy learning algorithm and a non-parametric method. It uses only the training data provided to it along with the hyperparameter K, which is the number of points that we need to consider in order to predict, so takes zero time to learn but the calculation time may vary depending on the K. During the training phase the model stores the data provided. In the test phase distance of the point under observation from the points in the training, phase is calculated in order to classify. Various distance parameters (Manhattan, Hamming) can be calculated. The one we used in the course was the Euclidean distance. The value of K indicates the number of nearest neighbors into account while classifying the dataset points. A very high K value will underfit the data whereas a very low value will increase the noise. There is no preset method for finding the optimal k value. We usually use an odd number for k since it's possible that the frequencies of the class are equal. To sum up, KNN classifies efficiently on low dimension datasets but with high dimensions, we tend to face problems (calculation time).

Clustering

Clustering is a form of unsupervised learning which segregates the data based on the similarities of the data points. Basically, similar samples are grouped together in the same cluster, while different samples are grouped in distinct clusters. Among different clustering methods such as k-Means Clustering, Hierarchical Clustering, Partitioning Clustering, and so on, the k-Means Clustering is captured in class. This method partitions the dataset to k-distinct clusters/ groups which do not overlap, as it can be seen in Figure X. In this case, 'k' represents the number of clusters and when determining the ideal value of K, the distances and the similarities of the samples in the dataset are checked. These distances and similarities are called within-cluster variations which are desired to be as small as possible, and between cluster distances. Different distance functions for numeric variables and categorical variables are also introduced in class. The relevant clustering method consists of an iterative process. The first step is to assign an initial number, thus a random number for 'k' is selected as cluster center/centroid. Secondly, the distance between each data point and centroid is calculated. Thirdly, the data point is assigned to the closer centroid, whose distance from the centroid is the minimum of all the centroids. Fourthly, the centroids of clusters are recalculated, thus in this step, new cluster centers are obtained. Finally, the distances between each data point and newly obtained centroids are recalculated and if there is no data point that needs to be reassigned to the closest centroid, then the method is finalized; if not, it is repeated from the third step.

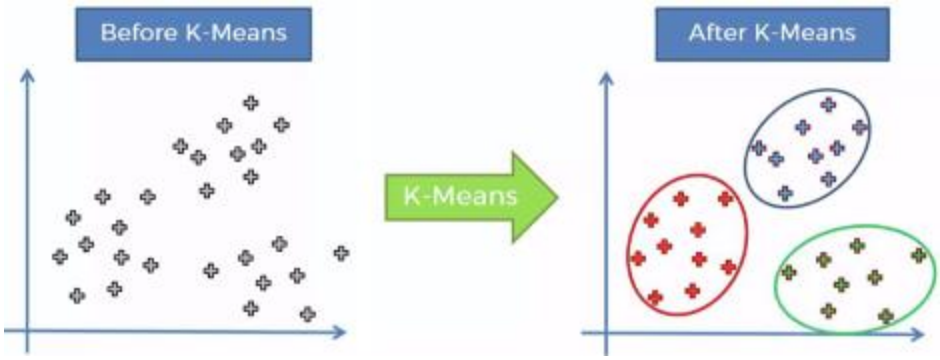


Figure X: Visual representation of k-Means Clustering

Naive Bayes Algorithm

Naive Bayes model is a type of classification machine learning algorithm. It is based on a statistical classification technique called 'Bayes Theorem'. It enables an easy way to create accurate models with a very good performance given their simplicity. Algorithms implement this by using an approach to calculate the 'posterior' probability of a certain event A to happen, with some 'prior' events and their probabilities.

To be able to use the Naive Bayes Algorithm for solving classification problems, the following steps must be done:

1. Data set should be converted into a frequency table

2. By finding the probabilities of the events to occur, a likelihood table should be built.

3. The Naive Bayes equation is used to compute the posterior probability of each class.

4. The class with the higher posterior probability is the outcome of the prediction.

Advantages of Naive Bayes:

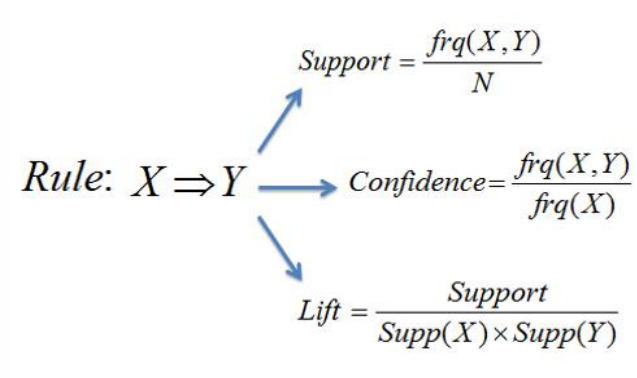
An easy and quick way to predict classes, both in binary and multiclass classification problems.

If the independence assumption fits in the problem, the algorithm performs better compared to other classification models.

The decoupling of the class conditional feature distributions actually means that each distribution can be independently estimated as a one-dimensional distribution. This helps with problems derived because of the dimensionality and develops the performance.

Association Rules & Apriori Algorithm

Association rules in data mining focus on relationship probability between data elements. These sometimes called "if-then" statements as well. In order to find out the association among items, we have measurement concepts. These concepts are actually constraints in order to find rules according to them. In other words, most of the time the task defines minimum values that the data set should pass this value for being eligible. First of them is *Support*, with the help of it we are able to find out how many times the selected item appeared in the data set. The figure below shows all the concepts for Association rules from ***X to Y***. **N** is the number of transactions that helps to find Support value.



The second concept is *Confidence,* which indicates the frequency of rules on true occasions. It's actually the measure of the "if-then" rule's uncertainty.
There is also one more concept which is called *Lift*, with the help of it we can evaluate performance of Association rule at predicting or classifying occasions.
Apriori Algorithm is a type of algorithm used for data mining when frequency of item sets are measured with the help of Association Rule. Apriori has 2 stages where in the first stage there should be generated frequent Item sets (stage is also called "Frequent Itemset Generation"). As it mentioned earlier, minimum support is actually provided in this case, where item sets should have minimum support value. In order to be eligible for next "phases". Second stage - Rule Generation, this stage goes around "minimum confidence" value to check whether confidence value satisfied or not.

Decision Tree

Decision Tree is a datamining algorithm used for supervised learning problems based upon either classification or regression. In a decision tree structure, each node is labelled with an input feature and the arcs to the nodes are labelled with possible values of the features.
The basic algorithm for decision tree involves splitting the data into multiple sets and each set is then further split into subsets in order to come up with a tree like structure and make a conclusion. The splits are made homogenously, the attribute that makes the most homogenous split is used. The process repeats itself for each subset. Gini Index, Entropy and Information Gain are used to measure this homogeneity of the tree. The concepts and there formulas are explained below:

Entropy: It is used to measure the impurity or randomness of a dataset.

$$Entropy(x) = -\sum(P(x=k) * log_2(P(x=k))$$

Information Gain: To find the best feature which serves as a root node in terms of information gain, we first use each descriptive feature and split the dataset along the values of these descriptive features and then calculate the entropy of the dataset. This gives us the remaining entropy once we have split the dataset along the feature values. Then, we subtract this value from the originally calculated entropy of the dataset to see how much this feature splitting reduces the original entropy which gives the information gain of a feature and is calculated as:

$$InformationGain(feature) = Entropy(Dataset) - Entropy(feature)$$

- The feature with the largest information gain should be used as the root node to start building the decision tree.

Gini Index: It is calculated by subtracting the sum of squared probabilities of each class from one. It favors larger partitions and easy to implement whereas information gain favors smaller partitions with distinct values.

$$Gini\ Index = 1 - \sum(P(x=k))^2$$

A feature with a lower Gini index is chosen for a split.

No one can decide for sure which of the above methods is the best one for any dataset. The decision for the best model varies from dataset to dataset and the values of accuracy, error, sensitivity. The values might change after pre-processing is done on data, so we need to consider each model for the newly produced dataset and then reach a conclusion.

Assessment

At first glance we realize that Customer Id is not a very useful attribute, so we don't use it in our processing and algorithm. We remove the 11 rows which have missing TotalCharges. After having a complete dataset with no NA values we move onto visualizing the data and figuring out the relation between all attributes and churn (target variable). The code is available in the R file our findings include the following:

1. For continuous variables we look for distributions, check outliers and then correlations:

    a. We observe that the number of customers with MonthlyCharges < 25 is very high => the distribution graph are similar between the people who churned and not
    b. no outliers were found all values are inside whiskers
    c. the TotalCharges distribution shows a high positive skew regardless of the churn status (binwidth test = 150,200,250)
    d. the tenure distribution is very different for the customers who churned and who did not . The skew is high and positive for customers who churned indicating that they are likely to end the subscription in early months. If customers don't churn intially we can see a huge rise in the distibution indicating their loyalty
    e. the attribute totalcharges has a positive correlation with both monthly charges and tenure

2. For factor variables we can observe that churn rate:

    a. is almost same for both genders
    b. is higher for senior citizens
    c. is lower for consumers who have a partner or dependent
    d. For lines and phone we can do a more in depth analysis
    e. is higher for people utilizing Fiber Optics services
    f. is higher for the people with no online (secuirty) services since they have left more

After this we prepare the data in order to apply decision tree algorithm by changing all character to factors. We divide the original data into a 75% training and a 25% test ratio for our model. The first decision tree model uses raw data as it's and predicts the churn status. The following are observed:

1. Accuracy for training data: 0.789
2. AUC value for training data: 0.793

3. Accuracy for test data: 0.794
4. AUC value for test data: 0.801
5. Other values are available in the R file

We observe a 79.2% for training and 80.2% for test accuracy rate with our data set and decision tree model. The AUC values are performance parameters which show that our model is performing at a very good successful classification rate.

The second Decision Tree uses the dataset excluding TotalCharges as well since the attribute was highly correlated to two others. Procedures for Decision Tree 1 is repeated but this time using the new data and the following observations are made:

1. Accuracy for training data: 0.804
2. AUC value for training data: 0.817
3. Accuracy for test data: 0.776
4. AUC value for test data: 0.798
5. Other values are available in the R file

We observe a higher accuracy for training data 80.4% but a lower accuracy for test data 77.6% in the second decision tree. Another major observation value is that Specificity values for both trees are very low. However, with the limited amount of data this is the best Decision tree that can be produced. For a better result we can try to use other predictive models and pre-processing methods.

Interpretation of Results

After pre-processing, visualizing and cleaning the data we applied Decision Tree algorithm from the one's taught in class in order to predict churn status for the customers. Our initial observations included the correlation between TotalCharges and both monthly charges and tenure. We observed an average accuracy rate of around 79% using our decision tree model for the predicted outcomes. The observed AUC values are high as well indicating a good performance model. We can try using other models and pre-processing techniques if we are hoping to find a more accurately predicting method.