Decision Trees: A Powerful and Intuitive Process to Predict Customer Churn

CASE STUDY

MAY 2020

Decision Trees: A Powerful and Intuitive Process to Predict Customer Churn

INTRODUCTION

Telco companies will always face the challenge of retaining as many customers as possible for their company. The most important asset in this challenge is the data telco companies already have. Cybiant can help Telco companies to identify costumers who are likely to end their contract in the near future. This opens possibilities for targeting those customers to keep them longer as a client. The variable of interest in this case is churn. It has value 1 if a customer ended his/her contract and 0 otherwise. There are various methods available to binary variables as churn on more attributes. Some of them focus on understanding which characteristics of the churn rate are the most important to predict the churn rate. Other methods deploy a powerful algorithm to predict which customers are going to churn. This second group has the drawback that they are often not intuitive and very difficult to understand for people without data-science background. We believe decision trees fit perfectly in the middle of these two. Decisions trees are intuitive and easy to visualize since they use a tree-like model of decisions. Furthermore, they have high predictive accuracy when optimized. The focus of this article is on the application of decision trees on customer data of a telco. The objective is to identify customers who are likely to end their contract within a short period.

THE DATA: TELCO CUSTOMER CHURN DATA

An example decision tree is shown in figure 1. This simple model is created using a sample dataset from a telco company. The target variable of our example is of course churn. The dataset consists of several attributes which provide characteristics of every customer. The twenty attributes included can be partitioned into three main categories: services that the customer signed up for, customer account information and demographic info about the customers. A few of the attributes consist of continuous variables, but most of them are categorical.

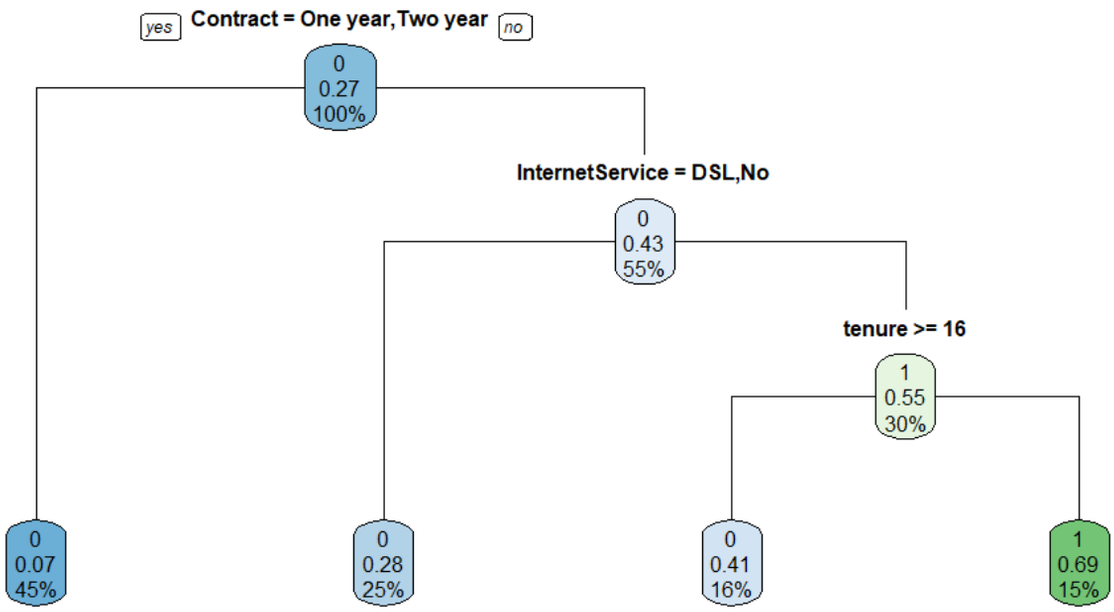

FIGURE 1

DECISION TREES EXPLAINED

'Decision tree' is a collective name for two different machine learning methods: a regression tree and a classification tree. A regression tree is used for numerical target variables. The churn problem requires a classification tree approach, which can have categorical or binary dependent variables. A modern and common-used abbreviation for decision tree is CART(classification and regression tree). The final decision tree we created for the sample data can be observed in figure 4. In figure 2 we made an example of a small decision tree. The tree has a root node, which contains the full set of customers in the data. One can see in the picture this node contains 100% of the customers. Furthermore there are more nodes, splits and buckets at the bottom of the tree. Now imagine some arbitrary customer of a Telco company. The tree can be understood as a kind of flow-chart. For each customer the questions in the tree are followed, such that the customer ends up in one of the buckets(or leaf nodes) at the bottom. The first split determines whether a customer has a one-year or a two-year contract or another contract. The only other form of contract for this company is a month-to-month contract. When a customer has a one- or two-year contract, it ends up immediately in the most left bucket. The 0 means that that customer is predicted not to churn.

The average value of churn in this bucket is 0.07. Since the churn is a binary variable, the interpretation is that customers in that bucket have an average churn probability of 7%. 45% of the customers in the dataset that is used to make the tree are in this bucket. In this way, it is quite easy and intuitive to understand how the model is used to identify customers who are likely to churn in the near future. Splits can be made on categorical as well as numerical data. For a numeric variable, there is chosen a value to split the data into two groups, one group smaller or equal to that value and one larger. This is displayed in figure 1 for the split on the tenure variable. The more complex tree in figure 4 works in exactly the same way as this small example. This larger model uses a larger amount of the information in the data in order to predict churn more accurately.

BUILDING THE DECISION TREE

The most complex part of a decision tree is determining how the tree should be built, although the main idea is quite intuitive. The first is to split the sample dataset into two subsets: a training set and a test set. The customers are randomly assigned to one of the two subsets. The training dataset is necessary to build our decision tree. We need the test set later on to examine the predictive accuracy of the model. An advantage of decision trees is that they can be applied to all kinds of data and datasets need relatively little cleansing and preparation before it can be used to build the model. Now the data is ready, we can start building the model. We used the programming language R to create a decision tree on this data. R is very flexible and includes great possibilities to visualize the tree and other aspects of the data. The main idea behind the algorithm used to build the tree is as follows. It starts with the whole training dataset. This dataset needs to be split into two smaller subsets to create branches of the tree. The main goal is to create buckets that are pure as possible with respect to the target variable ‘Churn’. In a perfectly pure bucket, all customers would churn or all customers would churn or all customers would not. Various measure for purity are available and could be used for decision trees. The measure we use on the telco data is entropy. The formula of entropy is as follows:

$$entropy = -p_1\log(p_1) - p_2\log(p_2) - \cdots = \sum_{i=1}^{n} p_i log(p_i)$$

Pi is the proportion of property i in the set. This formula is applied to the target variable 'Churn'. It has only two values 0 ('no churn') and 1 ('churn'). A perfectly pure bucket has entropy 0 and a bucket in which exactly half of the customers would churn and half would not has entropy 1. The aim of the tree is to obtain buckets which have an entropy which is as close to zero is possible. When the entropy is closer to zero, it becomes more clear whether a customer would be likely to churn or not, if it is placed in a certain bucket. Now we know how we can measure the purity of a bucket. The next question is how to determine the best way to split the data in two. Therefore, we need to calculate the information gain(IG) of each possible split. This is a measure of how much is gained in purity by a certain split on the data. The mathematical formula is:

$$\mathrm{IG} = entropy(root\ node) - [\mathrm{p}(c_1) * entropy(c_1) + \mathrm{p}(c_2) * entropy(c_2)]$$

c1 and c2 are the two resulting nodes after the split. p(c1) is the proportion of customers that end up in node c1 . When all possible splits are examined, the one which results in the highest information gain is chosen. Figure 2 visualizes an example of the difference in information gain for two possible splits on the same dataset.

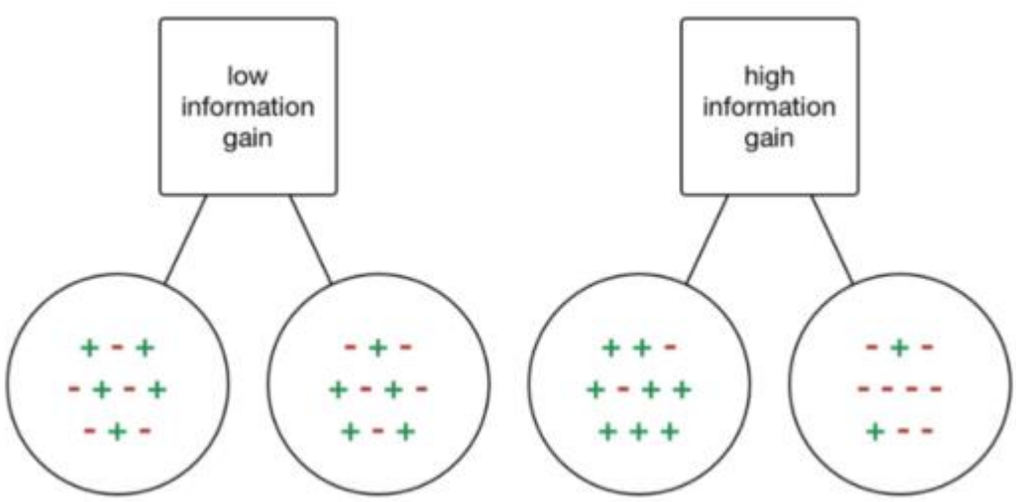

FIGURE 2

The right one is informative split, in the sense that it gives the purest subsets. In figure 1 you can see the first split is on the type of contract. The first question is whether a customer has a one- or two-year contract or another one. The only other one is a month-to-month contract. The probability of churn for customers who have a one or two year contract is only 7%, for customers with a month-to-month contract it is 43%.

This high difference confirms that it is useful to make a split on this attribute. This process of calculating information gain and splitting is repeated on each level of the tree on each different node. The information gain is calculated again for each possible split on the subset which includes only customers with a month-to-month-contract. The best attribute to split on for that subset appears to be the type of internet service. We could continue splitting further to gain information until almost each customer ends op in its own separate bucket. In the next section will be explained why this is not a good idea and which factors determine how complex the tree should be.

Now the model is built using the training set. The next step is to use the model the predict for any new costumers whether they are likely to churn or not. This is conducted by following the tree for each customer, such that they all fit in a bucket at the bottom of the tree. The customers in buckets with a 1 are predicted to churn, the customers who are in a bucket with a zero are not. It is also possible to attach the probabilities of churn of each bucket to the customers. Large quantities of customers can be processed through the tree in a reasonable amount of time in order to make predictions on a large scale. The values of ‘churn’ for each customer in the test dataset are known. We compare those values to the predicted values to examine the accuracy of the model in the next section. The accuracy of the model is the proportion of predictions that are correct. So if the accuracy is 60%, it predicts correctly whether customers are going to end their contract or not in six out of 10 of the cases.

OVERFITTING

The tree can continue to grow until no more splits are possible. The endpoint would be that all buckets are perfectly pure. Let’s call the tree with the maximum number of splits the ‘full model’. When we use the ‘full model’ to predict churn for the customers in the training set, it will give very high accuracy. It predicts for the same data as the model is built. The accuracy will be not that high on new data of new customers. This is because the full model is ‘overfitted’. The model is tailored too much to the training dataset. It will perform poorly in generalizing the model to new data. This is exactly the goal of our study of the churn data. Other disadvantages of a ‘full-model’ are that it is more difficult to visualize and that it is slower. Figure 3 shows the full model for the sample data.

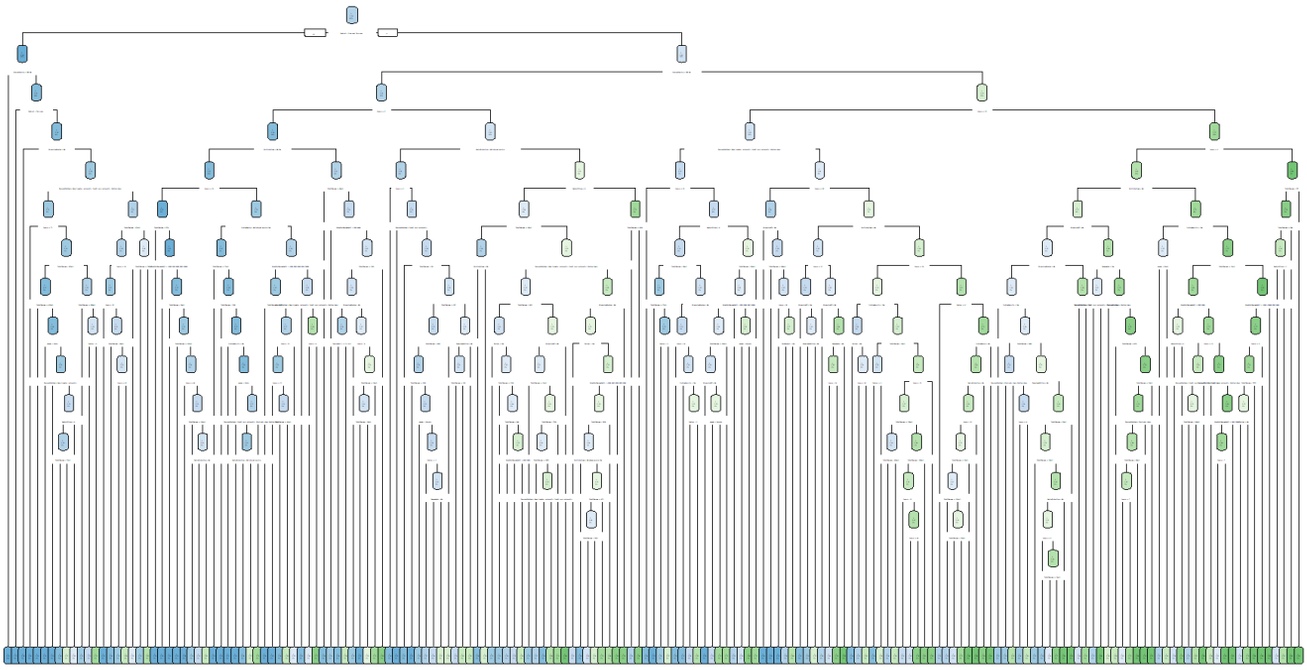

FIGURE 3

We should try not to overfit the model. Too little splits lead to buckets which are not specific enough. We should find a ‘sweet spot’ somewhere in between. We examine the accuracy of trees with different complexities to discover a model that has is highly generalizable.

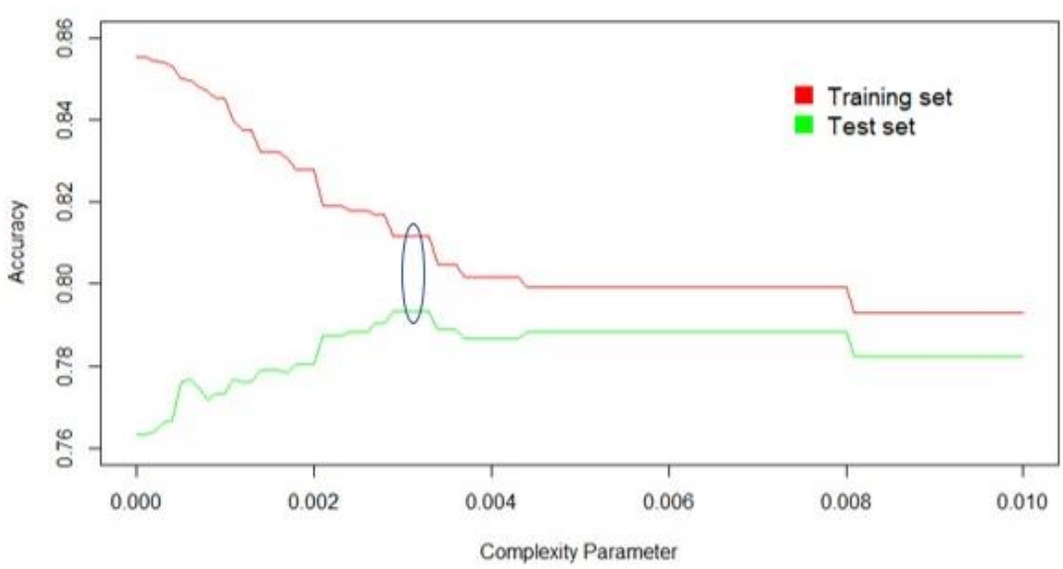

FIGURE 4

We plot the complexity of the models to the accuracy on the test and the training set in a fitting graph displayed in figure 4. The green line represents the performance on the test data, the red line is for the training data. The values on the x-axis correspond to the complexity parameter(cp). The mathematical derivation of this parameter is somewhat technical, but you can think of it as the minimum benefit a split must add to the tree.

So a split that results in an information gain lower than a certain threshold will not be performed. A lower value for the complexity parameter corresponds to a greater complexity of the tree. The performance on the training dataset increases always with the complexity as can be observed by the red line. The green line shows us the accuracy of the test data. The accuracy improves with the complexity until a maximum is reached. That “sweet spot’’ is represented marked in the plot by the blue oval and is somewhere around 0.003. We retrieved the cp that results. in the exact maximum(`0.0033) and used that to built the final tree. This tree is displayed in figure 5. This tree has more splits than the simple example in figure 1, but is way less complex than the ‘full model’, as can be observed in figure 3.

> This model reaches an accuracy of 79.3%. This means that it predicts 8 out of 10 times correctly whether a customer is likely to churn or not. This information will be highly valuable for a Telco company.

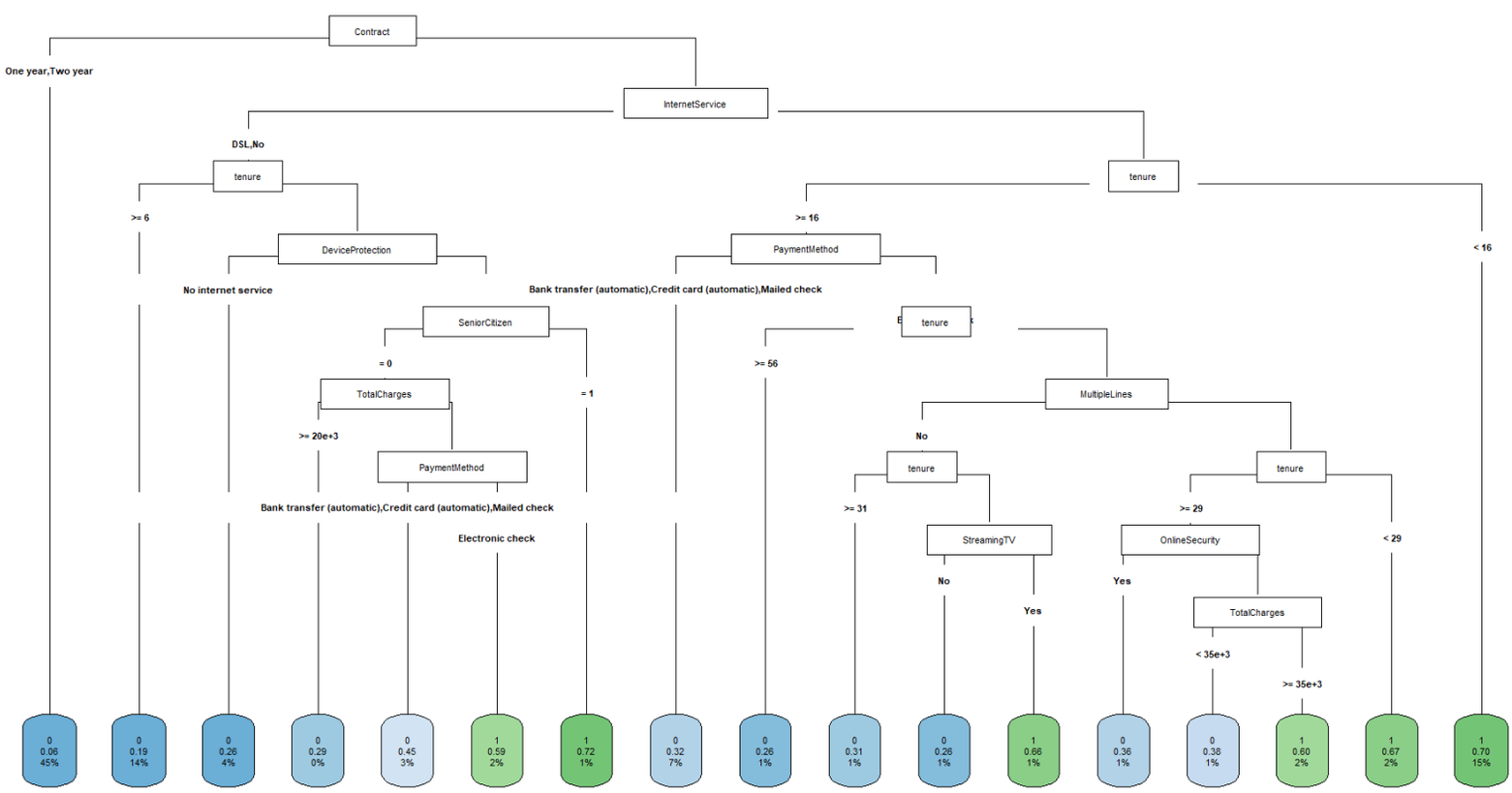

FIGURE 5

CONCLUSION

Retaining customers as long as possible is an important challenge for each Telco company. We used a sample dataset to built a decision tree model. This model can be used to predict whether a customer will end his/her contract soon or not. We, as Cybiant, have two important reasons to choose for this method. At first, we want to identify the customers in the client base which are likely to churn soon. Secondly, we want to create a model which is easy to follow for different people within a business. The decision tree combines both of these goals. As shown in the previous section, we reach a relatively high accuracy of almost 80% on a test dataset. It has to be noted that this accuracy depends on the customers in the dataset and even more on the available attributes for each customer. A major advantage of this method is that it can be visualized very clearly. This makes the model easy to understand. You can see which 'decision' the model is making for each step for each customer. The model is written in R, which provides excellent abilities to transform, perform calculations and visualize data(and models). After the tree is built and optimized it can even be integrated into Microsoft Power BI. This is a business intelligence tool, which is easy and intuitive in use and provides excellent opportunities for gathering all data in one place and updating the data. In this way, the model could be easily used and applied when new data becomes available.

Any questions?

Connect with the Cybiant Team:

+60 3 2724 7628 – Malaysia

+65 313 88 924 – Singapore

info@cybiant.com

References:

- Provost, F., & Fawcett, T. (2013). Data Science for Business (1st ed.). Sebastopol, California: O'Reilly.
- IBM Sample Data Sets. (2018). Telco Customer Churn [Dataset]. Retrieved from https://www.kaggle.com/blastchar/telco-customer-churn
- R (Version 3.6.3) [Software]. (2020). Retrieved from https://www.r-project.org
- Therneau, T.M. , & E.J. Atkinson. (2019). An Introduction to Recursive Partitioning Using the RPART Routines. Retrieved from https://cran.rproject.org/web/packages/rpart/vignettes/longintro.pdf