Machine Learning Algorithms for Churn Prediction

By: Tim Hollinger

04/18/2021

Table of Contents

I. Introduction

The ability to accurately predict future results is a crucial aspect within business. By using machine learning algorithms, predicting future outcomes is made easier through the analysis of historical data. The goal and process of machine learning is teaching a machine how to correctly predict a specified outcome such as customer churn based on the customer's previous behavior. Once the machine has 'learned' these patterns for prediction, the machine is then able to apply this learned knowledge to new data where there is no known outcome to come up with a prediction. The more difficult as well as tedious part of machine learning is selecting the most appropriate algorithm for the given use case, then fine tuning the parameters to get the most accurate result out of the model. The types of models used in application ultimately depends on the data that the model is being applied to as well as the end goal. The models covered will be one's that are applicable for predicting the churn probability for customers of the telecom company Orange. The data in question consists of 232 variables including the churn prediction and 10,000 customer observations.

II. Logistic Regression

When predicting probability, one of the most common methods to doing so is through Logistic Regression. Due to the nature of the data where customer churn is a binary value (0 or 1), logistic regression ultimately does the same after looking at all data that is provided for the model to learn from and assigns each ID a value between 0 and 1 as the probability. Given that a dataset is normalized, the threshold of acceptance is typically .5, so any value above .5 is rounded to 1 and any value below .5 is rounded to 0. Below is an example of a basic logistic regression formula:

$$\ell = \log_b \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

The main component of the logistic regression is the probability of success denoted as ‘p’ or failure which is denoted as 1- p; by applying a logarithmic transformation to the probability the graphical representation now looks like the graph below:

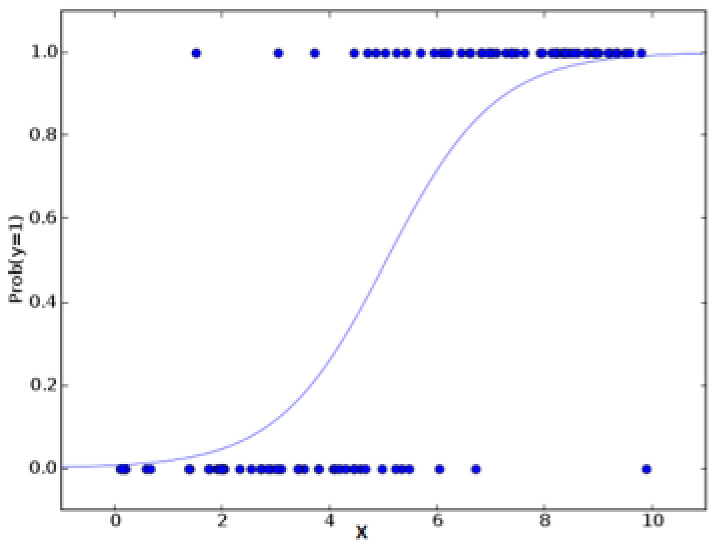


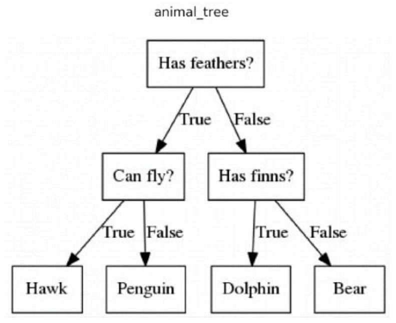
image source: https://www.analyticsvidhya.com/blog/2015/11/beginners-guide-on-logistic-regression-in-r/#:~:text=log(p%2F1%2Dp,association%20in%20a%20linear%20way.&text=This%20is%20the%20equation%20used%20in%20Logistic%20Regression.

As seen above, all data points lie at either 0 or 1 indicating the binary predictor. In order to get to the prediction process, the data must be prepared in such a way that the algorithm will be able to be fit to the data. As a basis, if there are any categorical variables they must be converted into a numerical variable that would be able to be used in prediction. Highly correlated variables must be removed in order to offset any overfitting of a prediction, the dataset must be separated from the output variable in this case the churn output of a customer which is valued at either 1 or 0.

There are many advantages with utilizing logistic regression, but one of the biggest reasons that the method is so popular in machine learning is the ease of implementation. As a result of logistic regression’s widespread use, logistic regression is often used as a sort of baseline for comparing the accuracy and AUC of other machine learning algorithms. The method is also extremely efficient in the time it takes to train, as well as when the data is distinctly separate and more easily differentiated. In a business setting the time efficiency of the model also results in a more efficient usage of company resources to implement their machine learning algorithm. With the advantages that logistic regression brings to machine learning applications, there are disadvantages that come with the model as well. In cases where there are a large amount of dimensions, the logistic model will have a tendency to overfit on the training data resulting in a greater variance in scores between the train and test sets. Logistic models typically require a sufficiently large dataset but the model is sensitive to outliers and complex datasets because of these factors, Logistic regression often does not produce the best possible results for prediction problems. Within the Orange prediction dataset, the logistic regression model produced an AUC of .5 and an accuracy score of .928

III. Decision Trees

When considering classification problems, a decision tree is an extremely effective method for distinguishing which category a specific item falls under. Essentially the decision tree decides classification by asking multiple layers of “yes” and “no” questions in order to properly classify or suggest the proper categorization for where a specific data observation falls under. When a basic decision tree is visualized such as the example below, it is easy to see where the tree portion of the naming convention is derived. As the decision tree begins with an initial ‘this’ or ‘that’ question and continues to drill down on specific features of the classification which at that point the model determines is the appropriate classification for that observation.



In this simple example, the decision tree is looking to determine what sort of animal the model is looking at by determining distinguishing features of the specific animal such as the ability to fly or whether the animal has fins or not. Depending on the answer the algorithm derives from the given data, the decision tree will be able to determine that the animal is either a hawk, penguin, dolphin or bear. When a dataset is more complex the decision tree gets more branches and layers to lead to the possible outcomes. If a tree has too many branches and is too large, there is a tendency for the model to overfit the data. To counteract overfitting of a decision tree, pruning can be used to reduce the amount of branches that may be highly correlated or have low importance. By pruning the a decision tree, the model's cost function is maximized by finding the branches that have similar splits as well as setting a minimum number of inputs for each branch. In order to utilize a decision tree for prediction and classification, the data must first undergo preparations as with any machine learning algorithm. Firstly identifying and splitting the input variables that will be used in the training of the model from the output variable that will be the target variable. When fitting the decision tree, a grid search is used in order to find the minimum number of samples per leaf, ideal number of leaves to be used in the decision tree that would result in the best AUC; in the case of the Orange dataset, the grid search found that the minimum sample size per leaf should be 1000 while the number of leaves should be 9. With these parameters the model returns an AUC of .53 and Accuracy score of .93.

One of the most advantageous aspects of using a decision tree is the interpretability, as seen above when a decision tree is visualized, understanding the outcome is easily understood by seeing each leaf that the data passed through in order to get to the predicted outcome. In addition to easy interpretation of results from a decision tree, the model also does not require much data preparation and is capable of easily handling categorical variables which is very advantageous from a time perspective. Decision trees also greatly benefit from very large datasets as there is more data to learn from, allowing the model to have more well defined splits leading up to the outcome for prediction. Among the many advantages of using a decision tree, there are aspects of the model to be wary of the potential for overfitting which is a common disadvantage of decision trees. In instances when data is adjusted or new data is introduced the model does not typically adjust very well and can result in highly variable outcomes. In the use case of Orange churn prediction, the decision tree produced an AUC of 1.0 and an accuracy score of 1.0, in this scenario the model clearly overfit the data and would improve through pruning and feature reduction.

IV. SVM

The third supervised machine learning algorithm used to evaluate customer churn prediction within the Orange data is the support vector machine or SVM. Much like the decision tree previously discussed, the SVM is commonly used in classification cases. The SVM is effective because of it's method of splitting data points by using a hyperplane. Simply put a hyperplane is the line that separates one classification of data from the other classification, the SVM will then evaluate the data points that lie near the dividing line to determine whether the classification is correct or not. For data points that are further away from the hyperplane the more confidence the model will have that the classification is correct, however for the data points that are very close to the hyperplane these data points are known as the support vectors in which the model derives it's name. Below is a simple visualization of the concept of using an SVM with a hyperplane splitting two classifications of data.

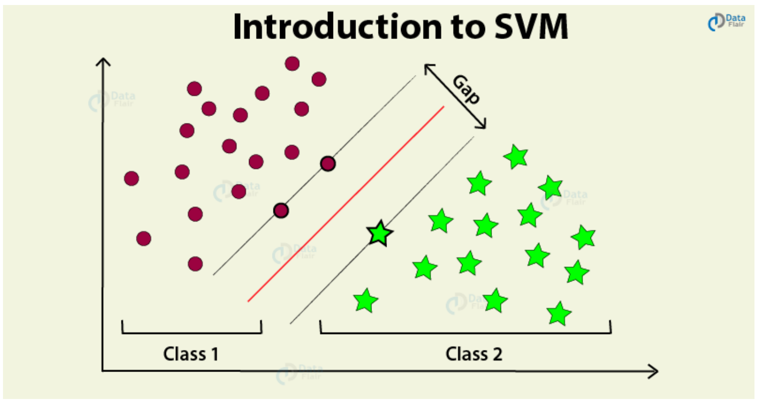


Image source: https://data-flair.training/blogs/svm-support-vector-machine-tutorial/

Of course, in more complex scenarios SVM's are able to have more than just one hyperplane to properly divide the dataset that is fit to the model. If the data is too complex and there is no distinguishable way to divide the data with a single hyperplane, changing the dimensions of the data from 2D to 3D can provide an alternate look at the data to allow for data division that correctly classifies the data provided. An example of this can be seen below:

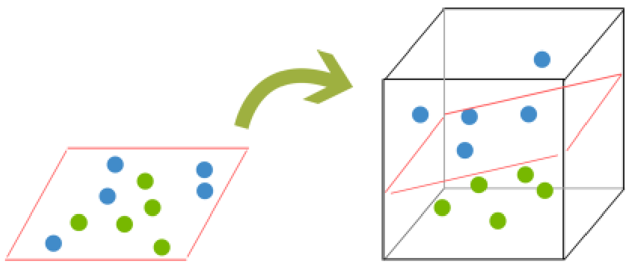


image source: https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html

By adding in another dimensionality into the data, the SVM can distinguish how to classify the blue vs green dots this can be done using SVM kernel. Which the formula to kernelize the data looks like this:

$$L_{\mathrm{d}} = \sum a_{\mathrm{i}} - \tfrac{1}{2}\sum a_i a_j y_i y_j K(x_{\mathrm{i}} \bullet x_{\mathrm{j}}).$$

Where "K" is what adds dimentionality to the maximization function of the SVM. The SVM is a more complex method than the previous two methods listed, however with the added complexity comes with better accuracy which the SVM certainly does. The high accuracy is one of the biggest advantages with choosing to utilize an SVM, the model works very well when classification within the dataset has clear distinctions between each other. Some of the cavaets of working with SVM's are that if the data does not have have very clear boundaries and has lots of noise surrounding the data, the model has a very difficult time with prediction so as a result SVM does not typically perform as well when dealing with larger datasets. However in this specific use case the SVM did perform better than most other methods although another drawback to running SVM is the long training times that can be exerperienced with large datasets and even the Orange dataset resulted in a lengthy training time for SVM. In this case SVM scored .928 in accuracy and .5 for AUC which was similar to Logistic Regression however when tested on the test data the SVM outperformed both Logistic Regression and Decision Tree methods.

V. Neural Network

Neural network models are a result of inspiration gained from observing the complexity of the human brain and how it allows humans to learn and function everyday. Neural networks are one of the more capable supervised machine learning algorithms as data is passed through multiple layers of neurons that learn how to classify the data that is given to provide an outcome. Below is an example of a basic neural network showing the input layer, the hidden layer(s) which are the layers where the neural network will learn from the input data then the output layer which is where the model will decide which output the data belongs to.

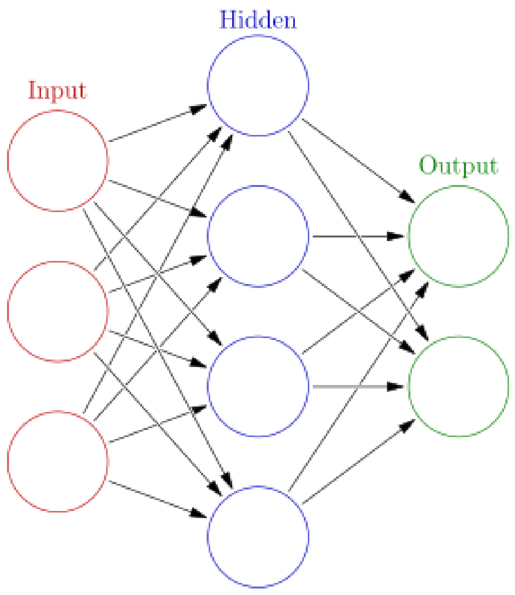


Image source: https://en.wikipedia.org/wiki/Artificial_neural_network

Neural network models learn through reinforced learning so the more data that the model is given the more familiar the model will become with the inputs that produce specific outputs. The visual above shows the inputs going into a hidden layer, which is vague as far as understanding how the hidden layer transforms the input values to the output values. When input values go into the hidden layer the inputs go to a neuron that has a weight associated with it depending on how important the neuron is, the sum of the weighted values for each neuron. The weighted sum of the neuron has a bias value added to it where bias is a constant value that improves the fit by adjusting the interception location of linear function. When fitting a neural network model to data, the data must first be split with the input values being separated from the output values. Following the split of the input and output values the model must then be back propagated which adjusts the weighted sums previously mentioned to account for erroneous values that were calculated, after the back propagation is completed the model produces the output prediction of regression, clustering or in this case classification. Through back propagation, the neural network is attempting to minimize the number of errors produced resulting in a lower RMSE which is ultimately the primary function of the neural network.

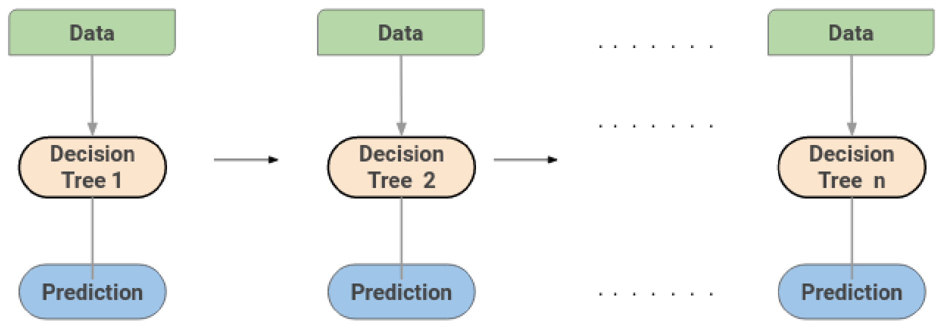
The neural network is best utilized for very large datasets, as previously mentioned neural networks utilize reinforced learning so the more data the model has to train and learn from the more accurate the model will produce. For the Orange dataset the neural network did perform well due to the large amount of variables and the complex nature of the data does not hinder the model's performance. In cases where the dataset is as large or complex as the Orange dataset used in this instance, the neural network is not utilized to it's full potential because the model isn't about to reinforce the values that result in classification. Another factor to be wary of when using a neural network is the fact that the formula and function can be so complex, that when an error occurs the ability to pinpoint the issue is very difficult and often the structure of the model cannot be explained. While the neural network was an effective model in this Orange dataset, the performance was not justified by the disadvantages that come with using the model.

VI. Gradient Boosting

The final machine learning algorithm evaluated within the context of the Orange dataset. Arguably one of the stronger performers amongst machine learning models. Gradient boosting is typically based on a decision tree and the model then identifies which branches of the decision tree are the weakest links of the tree and looks to optimize those branches by combining them into one stronger branch. The gradient boosting model focuses on optimizing the loss function by creating a new tree after each training to improve the weakest learners. Below is an example of the gradient boosting function

$$\hat{F} = \arg\min_{F} \mathbb{E}_{x,y}[L(y, F(x))].$$

The main component of the function is the last portion where L(y,F(x)) is the loss function that gradient boosting looks to optimize and F(x) is the function that continues to improve in each iteration as the overall function improves. By optimizing the loss function the model also mitigates the likelihood of overfitting the data. In preparing the gradient boosting model to fit data the most important aspect of the fitting process is that the model is able to learn from each iteration, using the previous decision tree as a foundation of knowledge on where the model can improve. As a result the fitting process of gradient boosting makes the model highly effective. Below is a visualized example of how gradient boosting works and learns to produce it’s predicted outcomes:



As seen above the gradient boosting model will iterate ‘n’ number of times to continue improving the model, this number of iterations should typically be specified to prevent overfitting which gradient boosting is prone to if a high number of iterations is given. In order to get the most out of using gradient boosting, parameters such as the learning rate which reduces complexity of the trees as the number of trees increase. In practice gradient boosting did result in the best AUC and accuracy within the Orange dataset with scores of .532 and .933 respectively.

In this instance gradient boosting was the best method for dealing with the dataset provided, the model performed well the high accuracy of gradient boosting is of course one of it’s biggest advantages and draws. The model is also highly robust as it can handle both numerical and categorical variables even if there are missing values. While gradient boosting models often look and perform well, as mentioned previously using a high number of iterations for the number of trees to learn from can easily result in overfitting. Gradient boosting’s complexity and intelligence does also come at a price in labor as training the model is often a long process and requires a lot of computing power that, in a real world scenario may not be worth a business’ time for a potentially marginal increase in performance in comparison to a much simpler method such as logistic regression.

VII. Conclusion

The evaluation of the aforementioned machine learning algorithms was contextual in relation to the Orange dataset, however this does not detract from the advantages and disadvantages of each method. Each model discussed has their use and place within machine learning. An important aspect to keep in mind while looking to solve prediction problems is not only the feasibility but also the aspect of real world implementation. In this scenario gradient boosting was a much better performer than the other 4 models tested against it however, the resources needed to implement a gradient boosting model in a business setting may not be as beneficial as taking a model such as logistic regression where accuracy is sacrificed in order reduce cost of time and labor. These models are also greatly affected by the data that is inputted into them so, the preprocessing of the data is just as important as choosing the right model. Choosing the appropriate model begins at the data processing stage but also does not end after initial model evaluation, as parameters can be fine tuned and experimentation can be conducted on the model afterwards.