Jan-May 2023

Report: A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees

Author : Jai Priyadarshi

Abstract

This project aims to establish a hybrid model namely the"Logit Leaf model" wherein this model gets carried out in two phases: In the first phase decision tree is used for customer segmentation followed by the second phase wherein logistic regression is applied at each leaf node, followed by combining the result in order to obtain the final result. An improvement to the Pseudo code is also proposed, wherein we put the pseudo-code inside a for loop and iterate the code for different values of depth of the decision trees and thereby selecting the optimum depth for maximum accuracy without underfitting or overfitting.

1 Introduction:

This project deals with the problem of Customer churn. Industries such as telecom, E-commerce, Ed-tech, etc want to know their customer churning rate as in today's time with a wide variety of options available it is very important to retain customers. So, the question is how is this project useful in that: basically, every customer has certain attributes or features attached to them, we use these features/attributes to create a model and predict whether a given customer will churn or not.
We have various models available for this purpose such as logistic regression, random forest, decision trees, etc, but all these models come with a trade-off between predictive performance and comprehensibility.
Thus here in this paper, we introduce another method known as "logit leaf model" (LLM) for the purpose of customer churn prediction. The tradeoff for various models is shown in below:

Classifier\ criteria	Pred. perf.	Comprehensibility

Logistic regression (LR)	+	+ +
Random forests (RF)	+ +	–
Logistic model trees (LMT)	+ +	–
Decision tree (DT)	–	+ +
Logistic Leaf model (LLM)	+ +	+

In this paper we have :

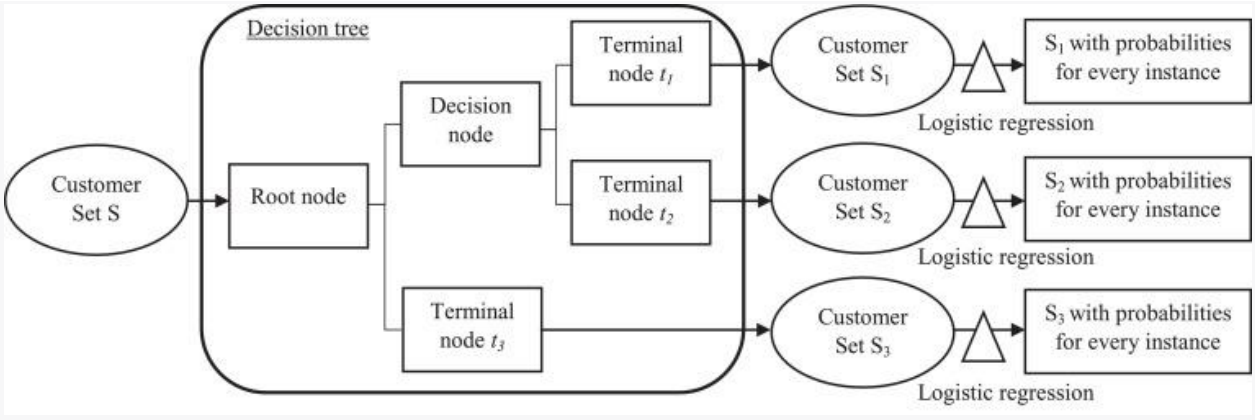
***Section 2.Literature Survey*:** It contains the details of the project from the research paper point of view, it explains in brief the mechanism of "logit leaf model".
Section 3. Methods and Approaches:It gives detail of the project and explains how is it different from the other models. We have also described the working mechanism of "logit leaf model".It also contains the modification that we introduced in the pseudo-code.
Section 4. Data set Details: It contains an explanation on the dataset used for the project and an understanding of the various customer features. It also contains details on data size, its nature, and type and also illustrates the pre-processing techniques. The data for the project was collected from Kaggle.
Section 5. Experiments: It contains a description of accuracy of decision tree, logistic regression and the logit leaf model. It also contains algorithm for the logit leaf model along with its pseudo code and the proposed improvement. Training procedures and optimisations are also discussed here.
Section 6. Results: It contains various graphs and tables obtained during the execution of the code, with an explanation.
Section 7. Future work: the work that needs to be done or further methods that could have been explored are discussed here.
Section 8. Conclusion
References

Literature Survey:

This algorithm namely "the logit leaf model is a two-step hybrid approach that constructs a decision tree in the first step to identify homogenous customer segments, and in the second step applies logistic regressions to each of these segments.

A decision tree starts with a root node which is basically a feature on which it divides and this division continues after each layer on layer. At each step, Information gain is

calculated based on which the node gets divided further. Finally, at the terminal node, we get our result. But there are some issues with this method, decision trees tend to have problems handling linear relations between variables. Another method that can be used for this purpose is logistic regression but even this has its own problem, logistic regression has difficulties with interaction effects between variables.
Thus, we come up with a mixed hybrid approach wherein customer segmentation is done in the first part followed by applying logistic regression to the root nodes in the second part.
The figure below shows a conceptual representation of the Logit leaf model displaying the flow of the data. In this representation, the entire customer set S has been divided into three subsets S_1, S_2 and S_3 by the decision tree. A logistic regression is fitted for every subset separately resulting in probabilities for every instance in the subsets.



Methods and Approaches:

We basically need to predict customer churn. The question that needs to be answered is "Will the customer continue with the services/company or discontinue? ".
Herein we will be using a hybrid method logit leaf model method wherein, the model carries out the predictive task in 2 steps. In the first step decision tree is used customer segmentation, followed by this logistic regression is applied at each leaf node.

So, let's try to understand how we proceeded with "logit leaf model method" :
First we split the data in training data and validation dataset.
The input that we provide is the training dataset $D = \{(X_i, Y_i)\}$ (i=0 to N)

We start by making a decision tree model on the training data which spans to a total space S, where S includes all the training dataset.
Now, we define subspaces S_t based on the number of terminal nodes T.

$S = \cup S_t$, tϵT

Now we run a loop for T times.
For i=1 to T:
    Here we define the logistic regression model
    Now we apply the logistic model to each node here to get output M_k.
End the for loop;

Now we combine M_k to give M as output.
This is the basic principle of how this algorithm works.

Modification to the Pseudo Code:

So, the modification that we made to the code was basically choosing an optimum value of depth of the decision tree in order to get maximum accuracy, without underfitting or overfitting.So, now basically the entire above pseudo-code goes inside a for loop running in range let's say (2,20). Now, we record the accuracy for the training dataset as well as the test dataset and plot it on a graph with the x-axis representing the depth of the decision tree. From close observation, we can select an optimum value for the depth of the decision tree such that the accuracy is maximized without introducing overfitting or underfitting.

Data set Details:

The data used for this project has been collected from Kaggle.
Each row in this dataset represents various attributes of customer. The Churn column represents the customers who left within the last month.

We get to know some personal details of the customer such as gender, are they seneior citizens (rough approximation of age), do they have partners and dependents.

We also get to know various account details of the person such as:how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges

We also get to know about various services that they have signed up for: phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies
The data frame is of shape (7043, 21)
The various datatypes present in this dataset are: float64(1), int64(2), object(18)
The size of the dataset is 1.1+ MB
The target we will use to guide the exploration is Churn
We have also manipulated our data to try to work only with the useful data that provide some valuable basis for analysis. For instance, the “customer ID “ column does not provide any valuable information thus we have dropped this column.
On deep analysis, we can find some indirect missingness in our data (which can be in the form of blank spaces).
Here we see that the “TotalCharges” column has 11 missing values.
It can also be noted that the Tenure column is 0 for these entries even though the MonthlyCharges column is not empty. Let's see if there are any other 0 values in the tenure column.
There are no additional missing values in the Tenure column. Let's delete the rows with missing values in Tenure columns since there are only 11 rows and deleting them will not affect the data.
To solve the problem of missing values in the TotalCharges column, I decided to fill it with the mean of TotalCharges values.
Now, moving further lets talk about data pre-processing.
First, we split the data into training and test set.
Since the numerical features are distributed over different value ranges, we will use a standard scalar to scale them down to the same range.

Experiment:

We basically split the data into the training set and a test set. The training set was used to create the model followed by testing it on the test set.
We first carried out the “Decision tree model” from which we obtained an accuracy of “0.7341232227488151”
Next, we used “Logistic Regression model” from which we obtained an accuracy of “0.8090047393364929”
Now, we build our “logit Leaf model”. We basically ran a loop for finding an optimum value of depth of the decision tree, inside which we carried out the procedure in 2 steps. The first step was customer segmentation using a decision tree, followed by the second step which was applying logistic regression on the leaf nodes. The Pseudo code is:

The input that we provide is the training dataset $D = \{(X_i, Y_i)\}$ (i=0 to N)

We start by making a decision tree model on the training data which is spanning to a total space S, where S includes all the training dataset.

Now, we define subspaces S_t based on the number of terminal nodes T.

$S = \cup S_t$, tϵT

Now we run a loop for T times.

For i=1 to T:

Here we define the logistic regression model

Now we apply logistic model to each node here to get output M_k.

End the for loop;

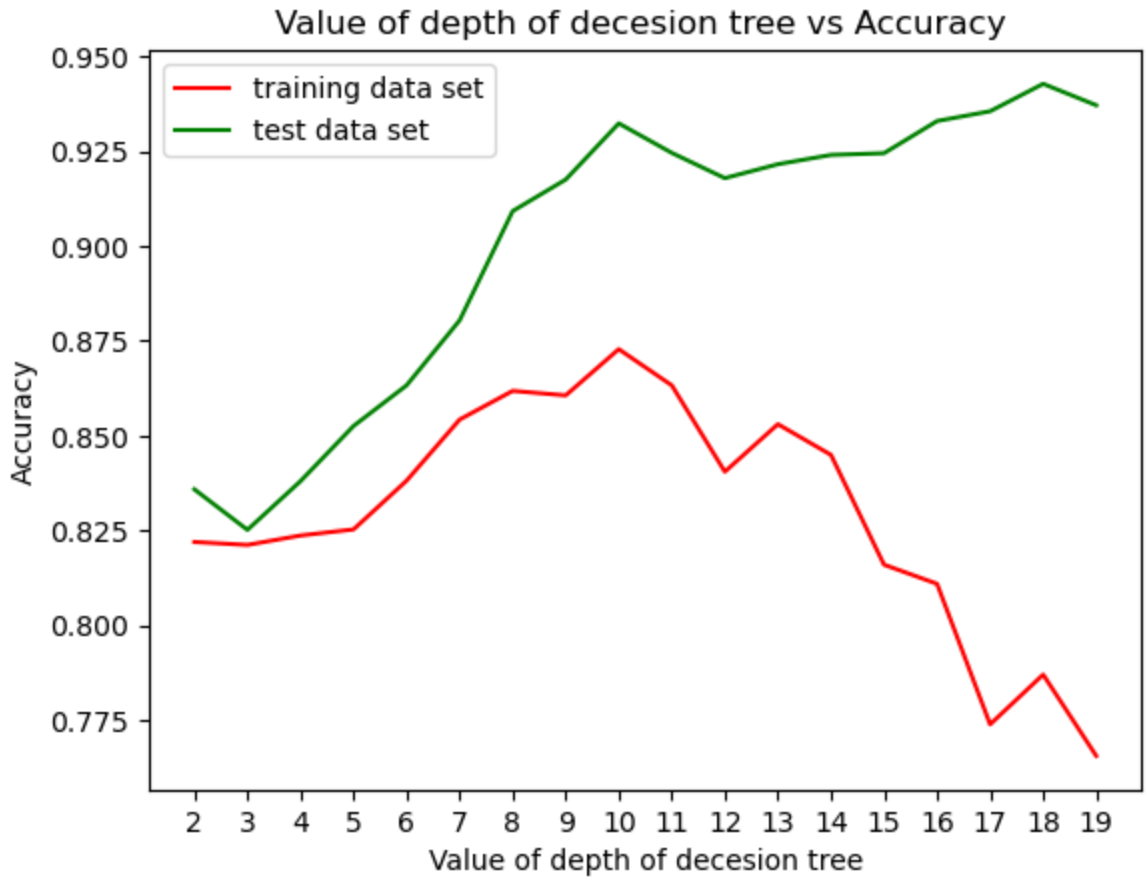
Now we combine M_k to give M as output.

For training the dataset we first made it to fit in a decision tree(whose layer depth was variable, but constant during a single iteration). Next, we used these datasets obtained from the decision trees at nodes to train a logistic regression model at each node and later combine the result of all the nodes. For, the accuracy of "logit leaf model" we took the mean of all accuracy occurring at different leaf nodes.

For optimization we ran the code for different values of depth of the decision tree and plotted the graph of accuracy of the training dataset and test dataset against the depth of the decision tree, from where we can obtain an optimum layer depth level from the analysis of the graph; such that the accuracy is maximum without introducing underfitting or overfitting. Here, we got the optimum layer depth as 10. This model gave us an accuracy of "0.9323949997142614", which comes out to be far better than the previous two models described above namely "decision tree model" and "logistic regression model".

Result:

When we ran the code for "logit leaf model" inside the for loop in order to obtain the optimum depth level we came up with the following graph. Wherein we selected layer depth as 10 because it gives the maximum accuracy and seems not to overfit or underfit.



We basically ran the model onto 3 different models namely - “decision tree”, “logistic regression” and the “logit left model’ the accuracy score obtained on these 3 models is given in the table below:

Model name:	Accuracy score:
Decision tree	0.7341232227488151
Logistic regression	0.8090047393364929
Logit left model	0.9323949997142614

We can clearly see the difference in accuracy score of these models, Thus our model of logit leaf performs much better than the other two model described above.

Future Work:

We used decision tree depth as the parameter for optimising Logit leaf model, this could have been even done in different ways such as deciding on the number of examples in a node, or when the information gain is below a threshold. We could check the logit leaf model made with these features to check whether it increases the accuracy or decreases it.

Conclusion:

The problem that was stated is customer churn prediction using a hybrid method namely “Logit Leaf Model”, wherein we carry out the model in two phases: In the first phase we apply a decision tree for customer segmentation, followed by applying logistic regression at the leaf nodes to obtain the result, in the second phase. We found the accuracy score for the decision tree, logistic regression and logit leaf model, from which we can clearly conclude that this method has improved the accuracy score. Now to forget to mention, we modified the pseudo-code such that we put the entire code inside a for loop and iterated for different values of depth of the layer of the decision tree. It helped us to maximize our accuracy score without underfitting or overfitting as evident from the graph above. Talking about future work, we may optimise a decision tree based on the number of examples in a node, or when the information gain is below a threshold.

References:

https://www.sciencedirect.com/science/article/pii/S0377221718301243#fig0002