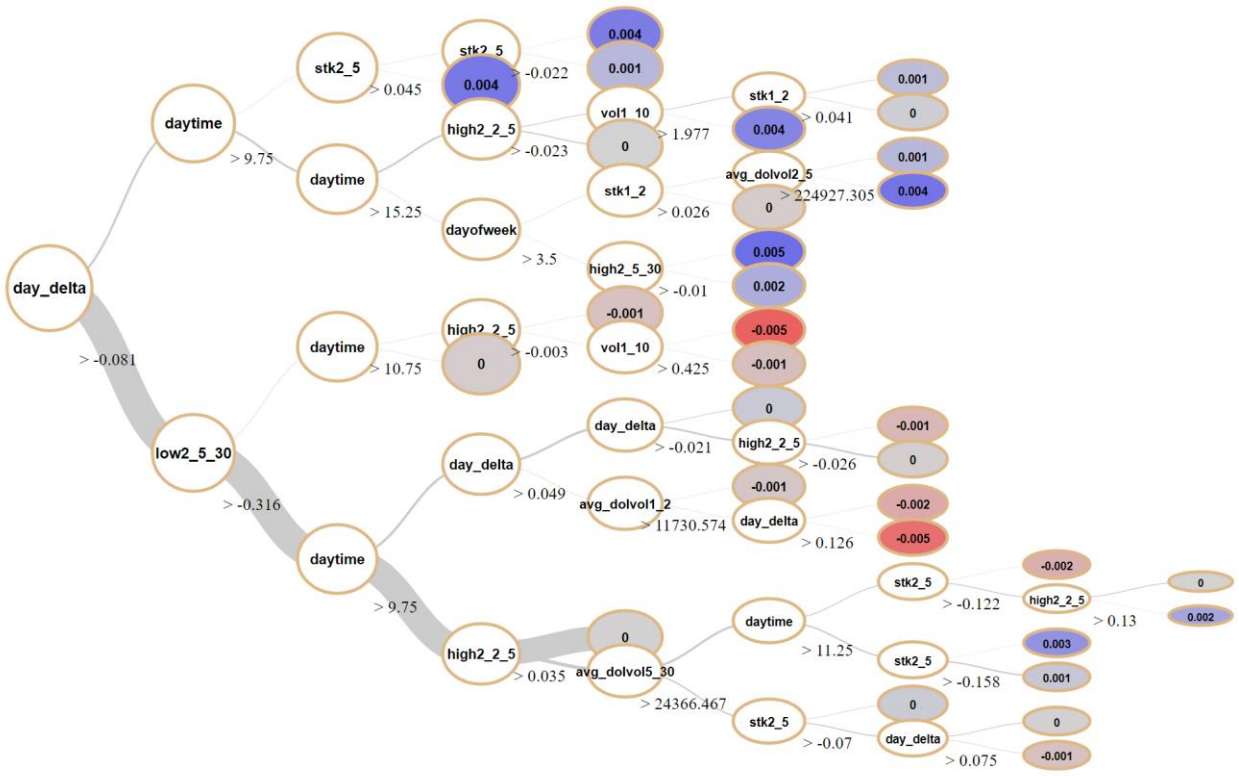
Jose Ortiz

Final Project, CSC 544

A Boosted Tree Visualization towards Financial Optimization



⓪ ① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨

Figure 1 - Fully-expanded view of a tree in the ensemble, trained to predict asset returns. Other trees can be selected using the navigation buttons 0-9. Branches are weighted according to the frequency that data travels through it, and the final leaves are colored and valued depending on their prediction. Intermediate nodes show the feature that splits a branch, and the lower branch shows the threshold value for that branch. Nodes can be contracted and expanded by clicking. Users can obtain detailed information about a node by a hover tooltip.

Gradient Boosting Tree (GBT) algorithms are a popular machine learning method for many reasons, the least of which is they are highly interpretable. Their building blocks are decision trees, making yes-or-no decisions step-by-step, each attempting to correct the flaws of those that came before it. It's not only poetic, it's intuitive. Potentially one could quickly gain useful insight into the decisions a GBT algorithm makes. For instance, often financial traders wonder about the interdependence of certain circumstances, and ask themselves questions such as "If the S&P500 has fallen by more than 1% and it's 10:00AM on a Wednesday, is that good or bad for my Google shares?" This is a very human way to ask a question, with each condition in the question slicing the feature-space exactly as a decision tree would. This correspondence between algorithmic decision trees and human decision processes drives the interpretability of tree methods, and motivates the development of effective visualizations.

While GBT algorithms are used ubiquitously, this visualization was in part developed to assist the predictions of financial returns. Hence, the input features here are data about the past of an asset, and the output is the predicted future fractional return of that asset, according to the selected tree. It is also inspired by one of the views used by Shixia Liu, et al (2017) in their toolkit, presented in *Visual Diagnosis of Tree Boosting Methods*. New features include explicit branch conditions, leaf coloring, and hover-over detail menus on nodes, leaves, and tree buttons.

Implementation Notes

LightGBM is used as the GBT algorithm, in a default mode of ten learners, with default parameters. Any GBT implementation that outputs standard DOT graph language, or a plain JSON tree, can take immediate advantage of the developed software. The developed visualization is written primarily in *d3*.

Use

The initial view when a user displays a new tree is a fully-expanded tree as seen in Figure 1. Users click the numbered buttons to navigate between trees, and are first struck by the difference between the full structures of the trees, as seen below.

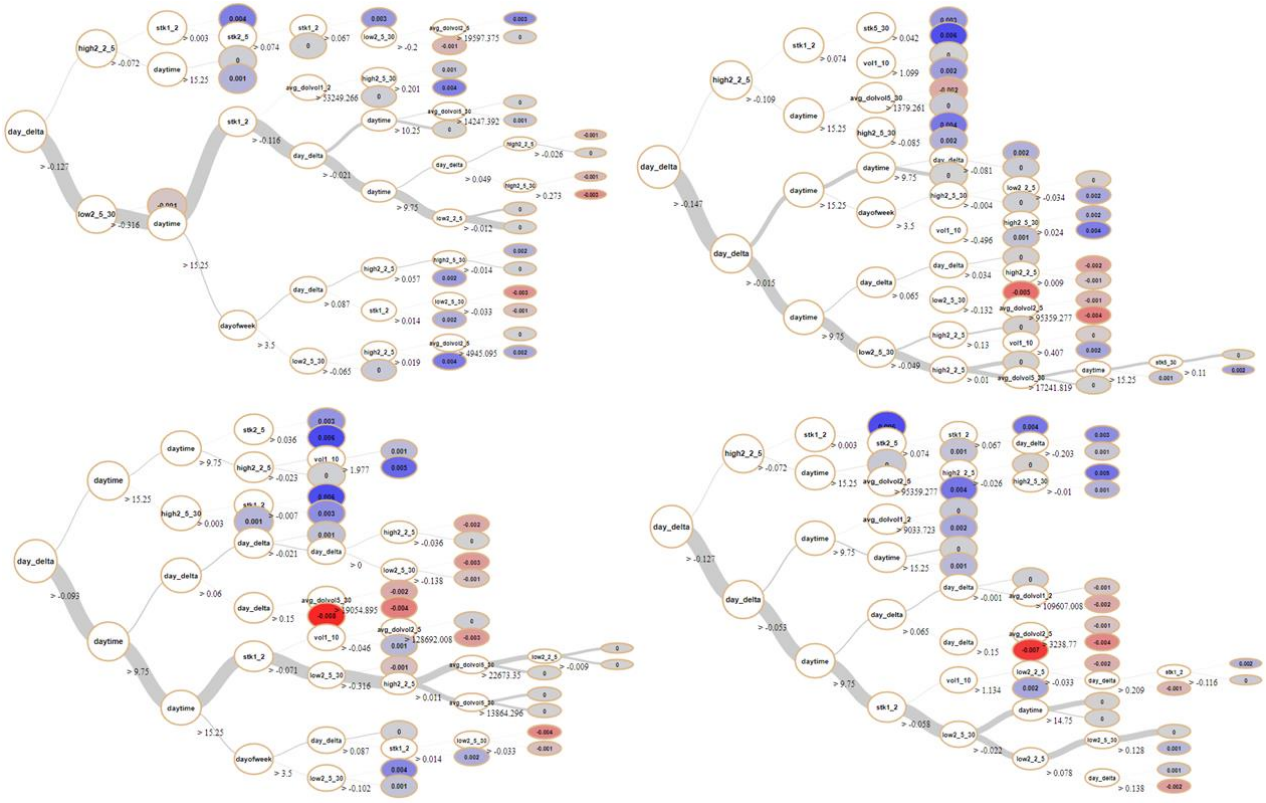


Figure 2 - Four different trees viewed in full expansion. Structural differences are clearly shown. By toggling through the trees, it becomes evident that most trees start with the same question: whether the stock in question has fallen dramatically over the day. Two trees on the right then differentiate by how bad, while the bottom-left one is clearly concerned about what time it is. These and other details readily pop out from comparing views, forming an effective narrative of the tree's character.

As the user quickly and efficiently compares the various strategies used by the trees, it becomes apparent that certain features are apparently extremely important, as all of the trees share the same root node, although diverging upon different criterion quickly from there.

Users can also collapse and expand nodes, as seen below. Collapsed nodes appear distinctly yellow, although it may be useful in the future to color collapsed nodes instead by a weighted average of the opinions of their leaves. Such color choices should be available as an option for the user.

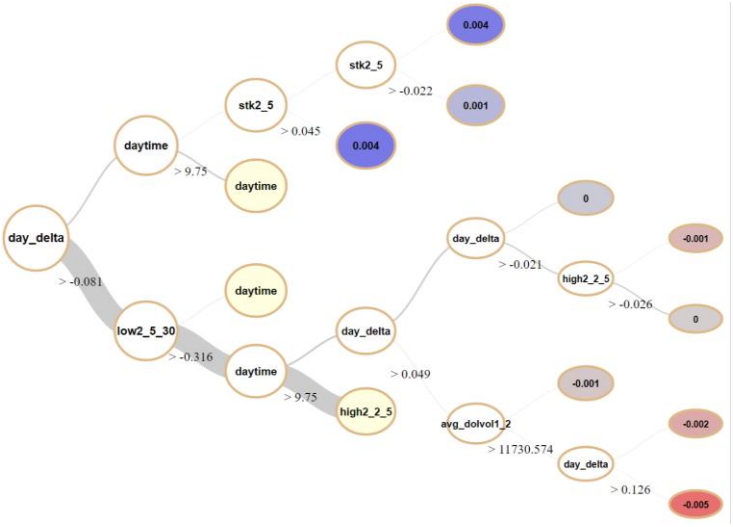


Figure 3 - Nodes collapse and expand in a smooth transition, and are highlighted when collapse

Users can also hover over elements to get a description of their parameters. Parameters of interest can be arbitrarily defined by the user and displayed in this toolbox.

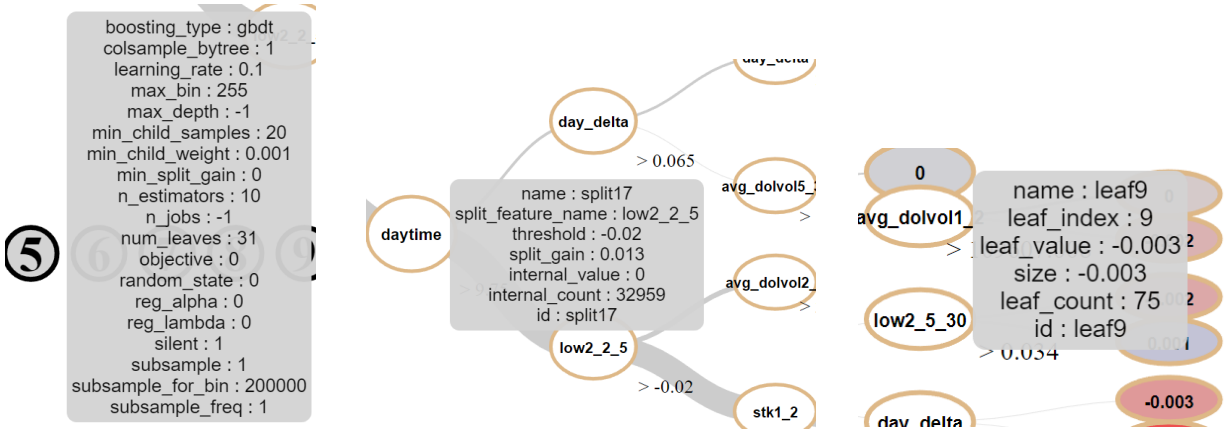


Figure 4 - A hover-over toolbox shows detailed parameters of trees (left) and nodes (middle) and leaves (right). Such parameters are useful, but sufficiently distracting to be placed in a separate view.

Finally, transitions and delays have been employed that greatly enhance the aesthetics of changing views or toggling nodes.

To come are major future enhancement, the first of which will be the inclusion of training and testing performance. Currently, only the frequency of events and the corresponding predictions are shown, but over the testing time period (or other untested parameter) predictive performance will likely vary in an interesting way.

Of great interest would also be to indicate the frequency of edge events or parameters of nodes as a function of real time, as if pulling of a graph of a street's traffic. This would provide great insight into how market conditions change, as well illuminating events that dictated the learned structure. Furthermore, other parameters of the appearance have yet to be used, leaving for now a clean appearance.

Github Project Repository:

https://github.com/joseortiz3/GradientBoostedTreeVis

Citied works:

"Visual Diagnosis of Tree Boosting Methods". 2017. Shixia Liu et al.
http://shixialiu.com/publications/boostvis/paper.pdf