Joseph Barr, Sunil Green, Ashley Konger

Team 38

CS373

16 April 2021

Preliminary Report

In the past several weeks, we have implemented the initial stages of our project, such as acquiring and preprocessing our data, the first of our two algorithms, and our k-fold cross validation. This has given us insight into the fundamental questions posed by our dataset; for example, we planned to use `scikit`, which does not support categorical features, to execute our algorithms—yet most of our features are categorical. Finding the best solution to these questions has allowed us to refine our approach, and we have made substantial progress since.

Data Preprocessing

As discussed in our project plan, we have chosen to investigate the problem of pork barrel legislation (i.e., wasteful legislation that misappropriates federal resources) in the United States Congress. In order to simplify our data, we are focusing our analysis on the House of Representatives alone. We hope that, by considering the features below, we can develop a model capable of predicting whether novel legislation is "pork barreled". Our data, in addition to the key column `bill_id` and the label column `is_pork`, includes:

- `cosponsors`: the number of representatives who cosponsored the bill
- `sponsor_party`: the party affiliation of the representative who introduced the bill
- `sponsor_state`: the home state of the representative who introduced the bill
- `committee_codes`: a list of the committees to which the bill was referred
- `subcommittee_codes`: a list of the subcommittees to which the bill was referred
- `primary_subject`: The main subject that the bill addresses (e.g., 'Health')

Acquisition

We aggregated our data from a few different sources. First, we used OpenSecrets.org to find information on which Representatives received the most contributions from lobbying, PACs, and individuals as a fraction of the amount of funding that went to the interests represented by the lobbyists, PACs, and individuals. Essentially, we considered the bills that produced the best "return on investment" for Representatives to be the most pork barreled, and thus labeled these bills as pork barreled. Next, we used the Citizens Against Government Waste Earmark Database to get more detailed data on specific pork barrel bills. Finally, we used the bill search available on congress.gov to find bills that are not considered pork barreled. Bills we identified as not being pork barreled did not contain any Congressional spending. Bills from the years 2008-2021 were used in our data set. We used the ProPublica API to gather data from all the bills identified (both as pork barreled and not pork barreled) to complete our dataset. A final note, much of the pork barreling that goes on in government occurs when Congress passes its yearly budget through a series of appropriations bills. As such, a large amount of appropriation bills constitute the identified pork barreled bills in our dataset.

Transformation of Non-Atomic Features

The final two features above—`committee_codes` and `subcommittee_codes` —contain lists of values, rather than atomic values. In order to resolve this, we transformed them by adding a binary column for each distinct value found in any list. For example, if a bill was submitted under the HSFA committee, then that bill would have a "1" value in the newly-created HSFA column, and a bill submitted under a different committee would have a "0". After this modification, we were able to remove the original columns.

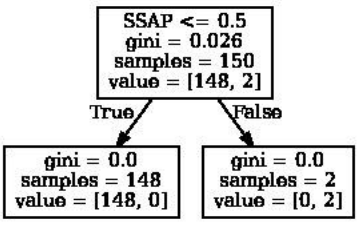
Encoding of Categorical Features

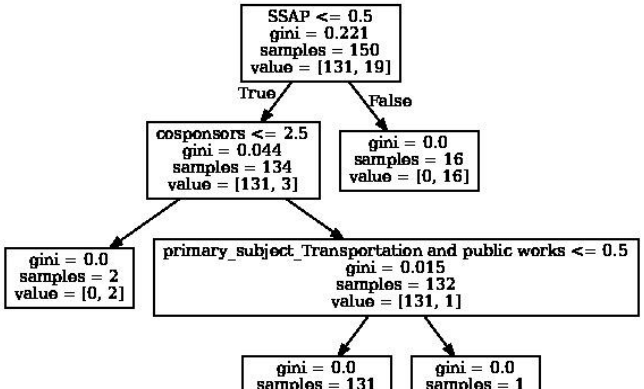
As previously mentioned, another issue with our data was the abundance of categorical features. A naïve solution to this would have been to encode them by incrementally assigning each value a corresponding integer key (akin to a C-style `enum`), but this has the potential issue of implying some quantitative significance. Instead, we used 'One-Hot Encoding'—which is, conceptually, an identical process to the transformation given above but does not assign a significance to the numerical values. Each possible value for a categorical variable is transformed into a column (feature) in our dataset and given a value of "0" or "1".

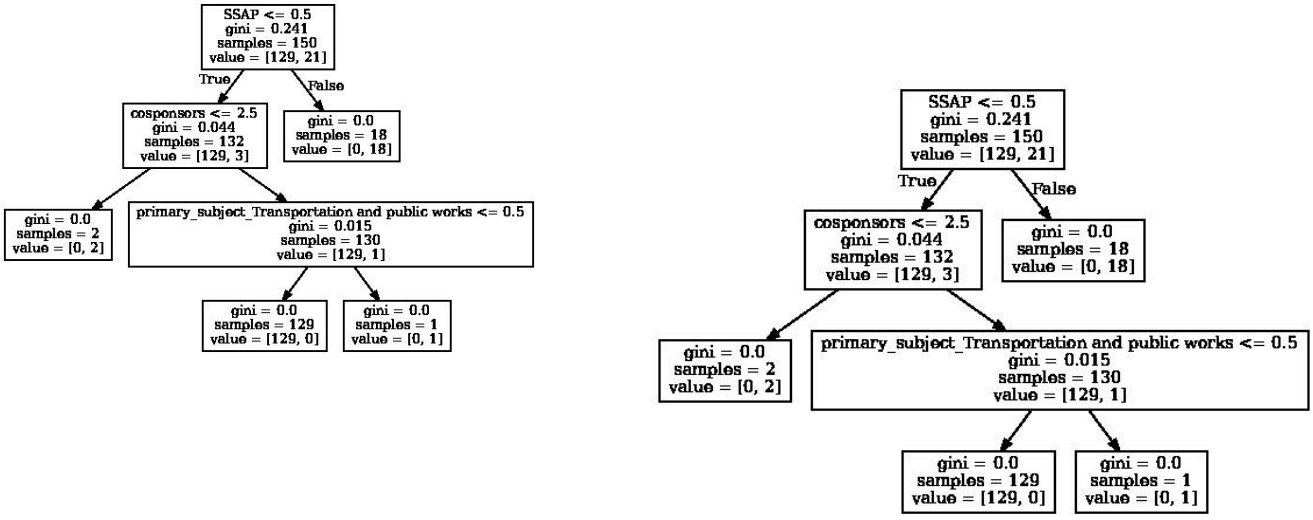
After performing both of these preprocessing steps, our data has a total of 168 features. These features are based on the 6 main features we identified outlined above, and the reason our number of features is so high is because of One-Hot Encoding's transformation of our categorical variables and the topic's reliance on categorical data.

CART and k-Fold Cross Validation

Thus far, we have implemented the CART classification tree algorithm, as well as cross validation for it. By using `graphviz`, we are able to visually display the resulting trees. Using our predetermined value of `k=4`, our model currently has an average classification error of only 6%. Here are each of the trees produced when using k-fold cross validation with `k=4`.







The 4 trees produced by cross validation.

Note: See comments in source code for details on how to run it.