South Carolina Legislative Document Classification

Wyatt Shiltz and Kevius Tribble

Background and Objective

Although state legislative documents are accessible online, the lack of an easy-to-use tool to sort through them has made it difficult for South Carolinians to be active in their state government. The goal of our project is to build a tool that will make it easier to search through South Carolina legislative documents by classifying them based on their contents so that citizens can find bills with information relevant to them.

Method Selection

For our baseline model we have chosen to use a gradient boosting classification model. We selected this model since it is one of the most popular classification models outside of neural networks. We avoided using a neural network for this model as we wanted our baseline model to be robust yet also simple, using more traditional data science methods. Since our data was naturally unlabeled, we also needed a model to label our data. For this we chose k-means clustering as it is a classic clustering algorithm that would be able to find similarities between the unstructured documents and assign them corresponding labels. This creates a unique approach utilizing both supervised and unsupervised machine learning methods to classify data that is traditionally unstructured and unlabeled.

Method Implementation

*Text Preprocessing*

Our dataset for this project consists of all of the South Carolina legislative documents from the 118th through the 125th general assemblies, which are publicly available on Legiscan, an online legislative database. These documents are in the form of unstructured text, so they needed to be cleaned, vectorized, and labeled before we could use them to train and test our gradient boosting classification model. The basic ways in which we cleaned the text include converting it all to lowercase, removing punctuation, converting numbers written in digits to their word representations, and removing spaces. We also removed common stop words, which are words that lack important meaning such as “the” or “a”, along with reducing words to their simplest forms through lemmatization. The goal of stop word removal and lemmatization was to get better vectorization results that captured the important points of a given document.

*Vectorization*

For vectorization we used Term Frequency-Inverse Document Frequency (TF-IDF) vectorization. We chose this approach over a simpler one such as Bag of Words since it has the advantage of considering the frequency a word appears in the document and the number of documents that contain it, allowing the later classification model to focus on words that have more relevance to the data. More advanced neural network-based approaches such as Word2Vec would also likely be effective in vectorizing our data, however these weren’t within the scope of our baseline model.

*Data Splitting*

Since our dataset is so large, we took a random sample of 1,500 documents to label with k-means clustering. We then split that data into testing and training sets, with 20% of the data being set aside for testing.

*Hyperparameter Selection*

The main hyperparameter that was relevant to this implementation is the value of k for the k-means clustering algorithm used to label the 1,500 sample documents. We determined the optimal value of k using the elbow method, which involves performing clustering with different values of k and finding the value that minimizes inertia without experiencing diminishing returns. On a graph, this is represented by the point that has the smallest inertia and a relatively flat line afterwards. Using the figure below, we chose 13 as our ideal number of clusters, although 5 is a really good contender since less labels can will result in a more explainable model.

A graph of a number of clusters

AI-generated content may be incorrect.

Figure 1: Elbow method graph showing the inertia of k-means clustering for various cluster sizes

*Evaluation Metrics*

The evaluation metrics that we used for the classification model included precision, recall, and F1 score, and all three of these metrics were collected for each class. These metrics were used to measure the performance of the trained gradient boosting classification model on the test data that was set aside previously.

Results

*K-Means Clustering*

*Below we visualized the clustering for 13 clusters and 5 clusters.*

A colorful dots in a line

AI-generated content may be incorrect.

Figure a: K-Means clustering(13 clusters) visualization using PCA for dimensionality reduction

A colorful dots in a line

AI-generated content may be incorrect.

Figure 2b: K-Means clustering(5 clusters) visualization using PCA for dimensionality reduction

There are some notable issues with the clustering. For example we notice that some of the clusters seem to be in the same area for the 13 and 5 clusters we believe that these are documents with similar meanings such as a house resolution or congratulating a group or individual.

*Gradient Boosting Classification*

The classification model performed rather well, with the average precision, recall, and F1-Score across all classes being above 0.85.

Table a: Classification performance evaluation metrics n=13 classifications

|  |  |  |  |
| --- | --- | --- | --- |
| Class | Precision | Recall | F1-Score |
| 0 | 0.81 | 0.81 | 0.81 |
| 1 | 0.86 | 0.83 | 0.84 |
| 2 | 0.81 | 1.00 | 0.89 |
| 3 | 0.83 | 0.83 | 0.84 |
| 4 | 0.75 | 0.67 | 0.71 |
| 5 | 1.00 | 0.89 | 0.94 |
| 6 | 0.89 | 1.00 | 0.94 |
| 7 | 1.00 | 0.95 | 0.98 |
| 8 | 0.50 | 1.00 | 0.67 |
| 9 | 0.85 | 0.96 | 0.90 |
| 10 | 0.87 | 0.63 | 0.73 |
| 11 | 1.00 | 1.00 | 1.00 |
| 12 | 1.00 | 0.93 | 0.96 |
| Macro-Average | 0.86 | 0.88 | 0.86 |

Table 1b: Classification performance evaluation metrics n=5 classifications

|  |  |  |  |
| --- | --- | --- | --- |
| Class | Precision | Recall | F1-Score |
| 0 | .94 | .86 | .90 |
| 1 | .81 | .91 | .86 |
| 2 | .84 | .93 | .89 |
| 3 | 1.0 | 1.0 | 1.00 |
| 4 | .91 | .85 | .88 |
| Macro Avg | .90 | .91 | .90 |

*Label Frequencies*

A graph of a number of bill types

AI-generated content may be incorrect.

Figure 3a: Label Frequencies for n=13 classes

A graph of a bar graph

AI-generated content may be incorrect.

Figure 3b: Label Frequencies for n=5 classes

Discussion

Overall, the classification model was successful in predicting classes for various legislative text documents. However, there is still work to be done to create a tool that can be used to search through South Carolina legislative documents. Although this approach did have good performance, one issue is that it is not completely clear how the classes generated through k-means clustering relate to the information contained within a document. While it does seem that documents labeled with the same class have similar topics, such as documents in class one generally being about criminal justice and documents in class three generally being about education, these themes can currently not be checked for all documents in a class to validate our results. In our final clustering/classification model, we plan on fixing this by dropping documents that are related to house resolutions or anything that isn’t a relevant law; we need to do more research to understand what data we can start cutting. This will allow us to look at more relevant documents and potentially prevent the clustering algorithm from getting confused by a resolution for a school and actual education legislation. We can either achieve this by using having human look at sample clusters and dropping all documents with a non-law label, then retraining, the model. Or we could try utilizing a pre-trained model such as MPnet to identify and drop irrelevant columns.