**Final Project Report**

Group 3: Jehan Bugli

**Overview**

This project tags congressional bills with one of [32 policy area designations](https://www.congress.gov/help/field-values/policy-area), replacing an important yet tedious task that is currently completed manually by the Congressional Research Service (CRS). This uses classical methods, with a focus on interpretability, to support practical use.

The CRS assists legislators with [policy analysis](https://www.congress.gov/crs-products), [official bill summaries](https://www.congress.gov/help/bill-summaries), and more; however, their job is rapidly becoming more difficult due to [rapidly increasing bill introductions](https://rollcall.com/2025/03/05/publishing-pileup-congressional-bills-slow-to-reach-public/) causing significant backlogs for official text releases, which in turn affects third parties reliant upon their processing efforts.

This report will:

* Discuss the data collection and processing efforts involved
* Describe the chosen modeling approach and experiment setup
* Outline the experiment results and conclusions

**Data collection and processing**

Data was sourced from the [GovInfo bulk data repository](https://www.govinfo.gov/bulkdata) which stores XML-formatted bill data for recent congressional sessions; this dates back to the 113th session (2013-2015).

To start, I downloaded bill text for each bill in the available sessions. This includes every version; for instance, if a bill received amendments between introduction and enactment, both the amended and original versions would be included. These were saved as a collection of session-specific .zip files.

I also downloaded associated data for each bill, including policy area tags and other CRS-processed items, in the same format as noted earlier.

Before further processing, I opted to “flatten” both folders, unzipping contents into two main directories.

Then, I converted each flattened folder into a tabular dataset. The folder with bill text was processed to generate a parquet file storing each file’s name and bill text.

Using BeautifulSoup’s XML parser, I searched for valid legislative body tags (e.g. ‘legis-body’, ‘preamble’) and discarded components such as the title, publishing date, sponsors, sessions, and more. While some of these items certainly may have provided some predictive value, I was concerned that such modeling may step outside of the NLP realm and muddy the results’ interpretability.

The folder with additional bill data was processed to generate a parquet file storing policy areas alongside text version names.

These text version names were used to join the two files and construct my finalized input dataset.

**Modeling approaches**

At its core, my approach uses Term Frequency – Inverse Document Frequency (TF-IDF). TF-IDF emerged out of pioneering information retrieval research as a method to quantify word importance.

This combines two key metrics:

* **Term Frequency (TF):** How frequently a term appears within a single document
* **Inverse Document Frequency (IDF):** How rare a term is across the entire document collection

The TF-IDF score multiplies these metrics to weight terms that are frequent within a given document but relatively rare across the entire collection of documents. This weighting can prove useful for many tasks, including feature extraction for classification.

I trained three different models on these extracted features:

* Linear Support Vector Classification
* Logistic Regression
* Multinomial Naïve Bayes