The Mercatus Center at George Mason University is a university-based economic research center that aims to fill the gap between academic ideas and real-world problems. Mercatus employs over 150 researchers, outreach professional, and support staff, who work together to provide economic research to the federal government and many state governments.

Mercatus has produced hundreds of documents that include full studies, research briefs, editorials, congressional testimonies, and more. These publications are posted publicly on the Mercatus website ([www.Mercatus.org)](http://www.Mercatus.org)) where they are tagged to aid in organizing and presenting similar research. Until now, this tagging has been performed by junior employees who choose from existing tags or create their own, a system that has led to a proliferation of tags and a sometimes-inconsistent approach to document labeling.

The goal of this project is to resolve this problem by creating a predictive tagging system. Such a system should be able to analyze a submitted document and return a list of tags, sorted by predicted probability that the tag correctly describes the document.

This project has several stages:

* *Gathering the data* – The existing publications must be scraped from the Mercatus website, together with tags, authors, and other metadata.
* *Processing the data* – Tags that are obsolete or underutilized should be removed, and text should be cleaned up to aid in model training.
* *Training and testing models* – Machine learning algorithms from the scikit-learn Python library must be trained, tuned, and tested on the source documents.

The end goal of the project is to have a script that can be used to classify new research documents. A web interface is a potential add-on for future development; this project aims simply to produce a script that can classify a document in the working directory from a command-line interface.

The Data

Data for this project comes directly from the Mercatus website, collected with a webscraping script I wrote and ran from a Jupyter Notebook. This script takes a starting “tag page” on the Mercatus site and trawls through the hyperlinks within that page, adding any “tag page” and “publication page” it finds to associative arrays for both types. It then loops through these arrays indefinitely, changing the associative array value of each URL it looks at from 0 to 1.

For each tag page in the associative array, the script will gather all tag page and document page links on the page. For each publication page in the associative array, it will perform the same link-gathering function, as well as saving the publication’s text, together with metadata such as title, date, authors, and publication type, to a text file. The “data” directory in which these text files are stored is organized by subdirectories of tag names, so each document is saved in multiple locations, with one copy in each tag subdirectory that applies to that document.

The resulting data includes 142 tags, which serve as labels for our training documents. Many of these labels are redundant, so consolidation is necessary. Through manual processing, six labels are removed, as documented [here](https://github.com/DanielWFrancis/springboard/blob/master/Manually%20Cleaning%20Labels.ipynb).

Further reductions are made after documents and labels are imported to a pandas dataframe for actual model training. The first reduction is of documents themselves, namely documents whose character count falls under a threshold level. Such documents are often the result of PDF conversion errors and simply contain the URL of the document and no other trainable data. Then documents with NaN values are removed – these are exclusively documents whose dates did not save correctly. Further label-level reductions include the elimination of any label that has five or less documents, due to the difficulty of training on such labels, and the removal of labels that have not had a new document added in the past four years, to remove labels that are not currently relevant to Mercatus’ mission and research activities. What remains is 91 labels spread over 1,873 unique documents.

*Further exploration of this data is available* [*here*](https://github.com/DanielWFrancis/springboard/blob/master/Apply%20Data%20Storytelling.ipynb)*.*