The Dissertation Committee for Guy Freedman certifies that this is the approved version of the following dissertation:

Machine Learning Algorithms in Political Research

Committee:

Sean M. Theriault, Supervisor

Bryan D. Jones

Alison W. Craig

John D. Wilkerson

Machine Learning Algorithms in Political Research

by

Guy Freedman

Dissertation

Presented to the Faculty of the Graduate School

of the University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August 2022

To my parents, Sue & Stan,

– because of how worried you were when all I read as a child were comic books.

Acknowledgements

Writing a doctoral dissertation is a journey. Like in most journeys in life, I thought I knew where I was starting (I was wrong) and I knew I had no clue where I would end up (I was right). This section is that one that most readers skip, aside of course from those few who expect to be mentioned. I really hope I remembered to thank all of those who deserve it. If I left you out, please know it was an innocent mistake due to too many sleepless nights and I really do appreciate your support along this journey.

I have been fortunate in life to allow myself to dream. I have been especially fortunate in making several dreams come true. Prior to this dissertation, I made the dream of doing work that served a greater purpose than myself, come true. I made the dream of travelling the world come true, visiting places from the Galapagos islands to the Great Barrier Reef. I met—and married—the love of my life. Coupled with the journey that is this dissertation I have now made so many more dreams come true. I have now lived abroad (and come home). I have learned from the best of the best (only to find out they were more interested in what I had to say). I have added the title father to my collection. Twice. With this document, I am priveleged to earn a new title. For all of this, I give thanks.

To my adviser Sean Theriault. I wish upon every student to find an adviser who cares for them personally as much as Sean has cared for me. His sharp mind, expert knowledge and pragmatic approach enabled me to freely and efficiently pursue the discoveries that were most interesting to me. Sean's succinct comments on my work have always allowed me to clearly see the next steps and the places I can both learn from the most and contribute to the most. In the guidance he has provided me, my professional development was always key, but my personal well-being was of the utmost importance.

To the members of my committee, Bryan Jones, Alison Craig and John Wilkerson (and, again, Sean). My expectation heading into this journey was about the surreal opportunity to learn first-hand from the scholars who have been at the forefront of the discipline for years.

I did not expect to earn their respect and be treated by these giants as a fellow expert. I thank you for never laughing to my face about the crazy ideas I raised and for your helpful suggestions on improving my research. The lessons you have taught me are invaluable to me and many of them go well beyond a scholarly exercise.

Chris Wlezien was instrumental in bringing me to study at the University of Texas at Austin. Chris offered guidance—professional and personal—and served as a mentor to me for the better part of two difficult years. He welcomed me, and my family, into his home and supported me every step of the way. I am forever grateful.

One of the most influential professors I had the pleasure of engaging with at UT was Robert Luskin. Robert, although I ended up pursuing a very different research interest, you have no idea of the intellectual impact you had on me. Your teachings were for me the rarest moments in which I understood what it truly means to thoerize (a weakness that is solely my own) and every lesson, essay or meeting with you over coffee left me thoughtful (and frankly a bit concerned for the future of democracy).

To the Policy Agendas Project (PAP) at UT. Bryan Jones has created one of the greatest research bubbles in political science and this community gave me a framework to share ideas and be part of a group in an otherwise foreign land. The faculty and fellow students that take part in this group are an incredible force. I thank you for always giving me a place to present my work. I also thank Derek Epp, Zeynep Somer-Topcu, Dan Brinks, Annette Park, Chaz Naylor and the entire Government faculty and staff for all you've done for me. To my closest friends from the department—David Futscher, Sarah Heiss and Christine Bird—I'm so glad to have found you.

My academic achievements would not be possible without the mentorship, friendship and encouragement of Amnon Cavari. In a book we coauthored, we mentioned that Amnon and I met when I, an undergraduate student, had several comments on a course he taught. We have since become research partners, coauthors on several works, and close friends. We have grown apart and found our way back again. Quoting someone else (it may have been his

brother in-law), Amnon once told me that most students are squares—you have to push and work very hard to roll them across the finish line. I, he claimed, am a circle, who just needed a small nudge and went barreling full-speed ahead, dragging him with me. Amnon, thanks for the nudge!

I also want to thank Riskified. For the past two years I have been working as a data scientist at Riskified. This company has created a culture of learning that is unlike any other place. I was lucky to receive their encouragement to complete my Ph.D. and was given all of the tools necessary. So much of what I do at Riskified has influenced my dissertation, and Riskified has welcomed the insights, skills and ideas I bring with me from my research in political science (a seemingly unrelated discipline).

Finally, to my family. For teaching me to dream big and for standing beside me unquestionably no matter the sacrifice. Hagar, my wife, has no doubt sacrificed the most to allow me this achievement. You have also given me the strength I need to proceed and I am thankful to include this journey as part of our journey together. To my parents and brother and family. As a parent myself, I finally know how much we give our children to make the biggest dreams possible for them. I am eternally grateful for the opportunities you have afforded me. To my two wonderful, clever, funny and magnificent daughters, thank you for teaching me on a daily basis, what it means to learn.

Abstract

Machine Learning Algorithms in Political Research

Guy Freedman, Ph. D.

The University of Texas at Austin, 2022

Supervisor: Sean M. Theriault

In recent years, political science has witnessed an explosion of data. Political scientists have begun turning to machine learning methods to provide reliable and scalable measurements of such large datasets. Building on the emerging literature on the use of machine learning in political science, I contribute four major lessons to the students and scholars who wish to make the most of these methods. These lessons include the advantage of treating machine learning as a process, combining text as data with standard data practices, the strength of pooling together supervised and unsupervised learning and the importance of understanding a model's strengths and limits. Through two rigorous empirical chapters, I trace the process of machine learning in two case studies, with actual outcomes for two widely-used datasets in the discipline. The first centers on a model for identifying agency-creation in historical data of congressional hearings. In the second case study, I tackle a multi-classification problem of predicting one of 20 major policy topics (and over 220 minor topics) in congressional bills. I conclude with a look to the future of machine learning in the discipline as we shift from a first wave of the literature that served as an introduction to machine learning, to a second wave of utilizing machine learning in actual research on political data and the challenges that these data present.

Contents

List of Tables

List of Figures

1 The Data (R)evolution in Political Science

1.1 The Blessing and Curse of Big Data

The evolution of political science is closely tied to the evolution of data in the discipline. Early research relied heavily on theoretic approaches to understanding social problems and interactions, with minimal access to data in their modern application (e.g. Downs, 1957; Hardin, 1968; Schattschneider, 1960). Such studies lay the foundations that resonate in scholars' theoretical reasoning of politics to this very day. What they share in common is that the role of data was secondary to the theory. Some studies from this period of time relied on no data at all; others rested only on observational input from their own environment (e.g. Dahl, 1961).

Nearly 60 years after Dahl's (1961) publication, a new global dataset on members of cabinets emerged, bearing Dahl's original title "Who Governs" (Nyrup & Bramwell, 2020). In modern political science, data play a crucial role. Much of our time as scholars is dedicated to the collection, preparation and measurement of qualitative and quantitative measurements of data. Our use of data is no longer 'merely' complementary to theory, but has become equally important. Rigorous empirical analyses have become standard procedure and the major publications all host online repositories for replicating authors' data and analysis. In some cases, the data we collect have become a goal in and of itself—students are routinely encouraged to collect and thus 'own' an original dataset as both a demonstration of skill and as a pathway to publication.

The broad trend that characterizes political science as a whole is true to congressional research specifically. One would be hard-pressed to find current publications that do not cite some of the earliest foundational work of authors such as Mayhew (1974) or (Fenno, 1966; 1977). These studies are unique not only in how strongly they set the research agenda for years to come, but also in the fact that they relied on an exploratory approach to what the authors were confronted with in front of their very own eyes. Less than two decades later,

large-scale data started playing a major role in congressional research, for instance in our understanding of the federal budget process (Wildavsky, 1986) or in the governing patterns of unified and divided government through the analysis of legislation (Mayhew, 1991).

Research in this area continued to evolve with the advent of data resources. B. D. Jones et al. (1993a) revolutionized the study of policy-making in Congress, by providing a theoretical paradigm that continues to define policy research. Their theoretical achievement was also an empirical one; offering a coding system for measuring the policy agenda and introducing a unique dataset—and one of the largest of its time—of congressional hearings spanning the post-WWII era. Tens of thousands of hearings were put together in a single tabular dataset providing never-before-seen measurements of what it is that Congress does on a daily basis.

The dataverse of this sub-field alone has since exploded to include numerous datasets of policy-making both within the United States and across over twenty different countries (Baumgartner et al., 2019).[1] Authors such as (Adler & Wilkerson, 2008, 2013) further pushed the boundaries of congressional data collection, introducing the congressional bills dataset. This dataset has by now surpassed half a million observations of all bills introduced in Congress since WWII and has fueled several avenues of research. Similarly, advances in understanding the role of the media in covering or influencing public policy continue to provide scholars with large scale datasets (Boydstun, 2013; Dun et al., 2021; Soroka & Wlezien, 2022). And, as technological innovation introduces new tools for governing, scholars have doubled-down on the effort to collect and trace this activity, providing, for example, insights into senatorial representation in the age of social media through the analysis of over 180,000 tweets, in a span of just two years (A. Russell, 2021).

These large-data publications are not anecdotal outliers; they represent the trend of increasing datasets in political science as a whole. To illustrate, I plot the increase in file sizes of replication data in three of the highest ranking journals in political science: American Political Science Review (APSR), American Journal of Political Science (AJPS) and the

[1] See https://www.comparativeagendas.net/.

Journal of Politics (JOP). All three journals host replication data on the Harvard Dataverse (https://dataverse.harvard.edu/) and I queried the dataverse's API to extract metadata on each journal's publications. Data for APSR and AJPS are available consistently since 2012 (N=368 and 586, respectively). The JOP provides the journal's data since 2015 (N=706). This is an imperfect method by any means, but as Figure 1.1 suggests, we are witnessing over time larger files in publications in all three journals.

Figure 1.1: Increasing File Size in Top Three Political Science Journals

Big data is here. It holds promise and opportunity for all avenues of research that social scientists care about: measurement, description, formal theory and causal inference (Brady, 2019; Grimmer, 2015; Grimmer et al., 2022; Grossman & Pedahzur, 2020; Monroe, 2013; Monroe et al., 2015; Salganik, 2019). Researching policy-making in Congress has, in this respect, come to parallel policy-making in Congress itself. Just as policy-makers are often over-burdened with a fire-hose of information (Baumgartner & Jones, 2015), congressional

scholars now face the challenge of having too much data.

Congress supplies and documents so much policy content that the challenge is no longer about collecting sufficient data for hypothesis testing (a burden many political scientists faced in the past) and is instead about sifting through an abundance of data, sorting it and measuring qualities of interest. Hundreds of thousands of observations are available to us and it won't be long before we're faced with millions of data points to analyze. The human hours and skill it takes to collect and categorize these volumes of data in a reliable fashion can be tremendous. Human coders need to be expertly trained, their work must be reconciled to solve any disagreements and some measurements may be too costly for humans. With every observation that needs to be reviewed, and every additional variable we wish to measure, the duration and complexity of the task is multiplied.

This explosion of data presents new challenges for congressional scholars. How do we balance speed and scale with reliability to provide full and consistently measured data?

1.2 Shifting the Burden: From Human to Machine Learning

Machine learning (ML) methods are at the forefront of dealing with the challenges of big data. Machine learning is a sub-field of artificial intelligence that applies algorithms to make sense of data. Algorithms are able to identify and learn from patterns in data to create knowledge (Raschka, 2015). We can then apply that knowledge to provide measurements, insights and even decisions relating to unseen data. Their application is wide and varied, from targeted online advertisements or online fraud detection to the identification of cancerous growths (Amethiya et al., 2021; Perlich et al., 2014; Yee et al., 2018).[2]

In my dissertation, I build on the work of several scholars, who have taken up the task of illustrating how political scientists can benefit from machine learning (e.g. Grimmer et al., 2021; Wilkerson & Casas, 2017), transforming the challenge of big data into an opportunity. Some of the same statistical models political scientists use for inference can be used in a

[2]There's even an expert machine that takes out all of the fun of asking "Where's Waldo?"; see https://www.businessinsider.com/wheres-waldo-robot-ai-machine-learning-2019-2.

machine learning setting (for example linear or logistic regressions, although they tend to under-perform compared to more advanced developments in the field). If we shift our focus from understanding the (causal) relationship between X and y, to using a statistical model for making accurate predictions of y, we have shifted into the realm of machine learning (Cranmer & Desmarais, 2017; Grimmer et al., 2021; Molina & Garip, 2019; Mullainathan & Spiess, 2017; see also N.-C. Chen et al., 2018; Rudin, 2015; Wallach, 2016 on the differences between the social sciences and the approach that practitioners of machine learning usually adopt, including causal relationship vs. prediction and the deductive vs. inductive approach to theory-building and data collection).[3]

Prediction is a far more scarce exercise than causal analysis in the social science, primarily because its contribution to theory-building is thought to be limited (Shmueli, 2010). However, this trend is changing as scholars have recently begun to identify ways in which machine learning-based predictions can be used to inform thoeretical interests, including causal relationships (Grimmer et al., 2022). For instance, highlighting the relationship between campaign contributions and roll-call voting patterns (Bonica, 2018), using machine learning to empirically estimate delegation and constraint in EU legislation (Anastasopoulos & Bertelli, 2020), or using the accuracy itself of a trained model to gain insights on temporal political changes, such as the extent of polarization in the UK House of Commons (Peterson & Spirling, 2018) or the meaning of human rights standards (Greene et al., 2019).

The use cases of prediction that I examine in this dissertation are closest to the tradition of using machine learning to predict missing values in variables that are meant to be included as covariates in some regression model (Anastasopoulos et al., 2016; Fong & Tyler, 2021; Grimmer et al., 2012; Imai & Khanna, 2016; G. King et al., 2013; Stewart & Zhukov, 2009; Theocharis et al., 2016). Otherwise labeled as *scientific prediction*, this practice follows the logic that if some condition in X is true, we can accurately estimate the value of $\hat{y}$ even if we don't know the true value of y. I am less interested, here, in *pragmatic prediction*, which

[3]In inferential settings, authors often use the change in the predicted value of $\hat{y}$ for a given value of X to demonstrate the impact of X on $\hat{y}$, but rarely is the concern to provide an accurate prediction of $\hat{y}$.

relates to a more *prophetic* notion of prediction, predicting the likelihood of a particular event to occur in the future (Dowding & Miller, 2019). This type of practice is even more scarce in political science (with the exception perhaps of election outcomes, e.g. Y. Chen et al., 2022).

Just as in the inferential tradition, the choice of algorithm may depend on several aspects of our data, first and foremost our outcome variable of interest y. Unlike the inferential setting, the scale on which y is measured, and its distribution, are not the first questions we might ask. The first question we should ask is whether we know the true values of y for a given sample of data. If, in the data we plan to learn from, we have values or labels for our outcome variable, we may rely on supervised learning algorithms.

1.2.1 Supervised Learning

In supervised learning, we provide an algorithm with some collection of features X (the term for predictors or independent variables in machine learning lingo) and a corresponding list of y values. We rely on the algorithm to identify meaningful patterns in the relationships between X and y to allow us to provide accurate predictions of $\hat{y}$ in unseen or unlabeled data (i.e. data in which we do not know the true value of y and are relying on the machine to provide us with a good estimate). Below, I provide a descriptive take on several models; see an excellent review of these methods in Wilkerson & Casas (2017), Montgomery & Olivella (2018), Molina & Garip (2019) and Grimmer et al. (2021), including a greater emphasis on the mathematical foundations and assumptions of each model, as well as illustrations of their potential use in political science.

Some of the best performing supervised learning algorithms rely on ensemble methods of tree-based weak learners (Hastie et al., 2009). Imagine building a tree. At each step, our algorithm creates a split on some value of one of the features we provided the model, e.g. $X_1 > 5$ would create a split with two branches—if the value is greater than 5 it goes left, else it goes right. Next, it might split on another feature and so on. At the end of all these splits are "leaves" or nodes, which are essentially predictions of $\hat{y}$ values. At each step, the algorithm

splits on the feature (and its value) that would maximize the explainable variance of $\hat{y}$ *at that step*. Maximizing explained variance at each split is often referred to as the greediness of tree-based algorithms.[4]

The problem with a single tree is that it may be very sensitive to the structure of the data and is too weak to identify enough patterns to yield accurate predictions. But, if you were to build n trees (a parameter which you can control and optimize for the algorithm, e.g. 1,000 trees), each receiving a random subset of the features provided (and sometimes random subsets of the data as well), you may average across all of them to provide a much better performing model. You now have a forest of trees, which is the source for one of the most popular tree-based algorithms: Random Forest.

Tree-based ensemble methods usually perform much better than regression models. Muchlinski et al. (2016) for example, illustrate the strength of Random Forest algorithms, compared to logistic regression, in predicting the onset of civil war. Tree-based methods are appealing for three main reasons. First, they are far lass sensitive than traditional statistical models to co-linearity among predictors. Second, they are far more adept at dealing with rare-event data (Muchlinski et al., 2016). Finally, given the structure of the tree and its splits, handling multiple and complicated interactions is inherent to these methods (Montgomery & Olivella, 2018). Kastellec (2010) illustrates this last benefit in understanding legal doctrine through the analysis of search and seizure cases decided by the U.S. Supreme Court and confession cases decided by the courts of appeals. Green & Kern (2012) make a similar point, using Bayesian Additive Regression Trees (BART) in the analysis of a well-known experiment on public support for welfare spending.[5]

In Chapter 3, I compare the performance of the Random Forest algorithm to a gradient boosting model, which differ in several meaningful ways. The key difference I highlight is

[4]One downside of the greediness of trees is that it's possible that we could provide a better prediction by splitting first on a and then on b (their combination being the key), but if at the first step b explains more variance, it will first split on that, creating a sub-optimal prediction.

[5]My emphasis is on tree-based models because I chose to work with such algorithms in this dissertation. Of course, machine learning encompasses a wide array of algorithms. See for example D'Orazio et al. (2014) and Seb & Kacsuk (2021) on the use of support vector machines for document classification.

that while the Random Forest begins each tree with a random subset of features without prior knowledge of the previous tree's errors, a boosting model uses the knowledge of the previous tree's errors and tries to correct them. In my limited experience, the former may be more useful in small datasets with a small number of trees; the latter supersedes the former if the data are sufficiently large and we can increase the number of trees. Kaufman et al. (2019) demonstrate the superiority of a boosting algorithm over other predictive methods, including Random Forest, in predicting U.S. Supreme Court rulings.

In Chapter 4, I use one of the leading boosting algorithms today: Catboost. The main advantage of this algorithm that interests me (again, several differences exist compared to other models) is its unique method of allowing the use of categorical features in the model (other models require one-hot-encoding, i.e. a $k-1$ series of binary predictors, each representing a different category and leaving one out as a reference category).[6]

Unlike regression models, which require us to choose an appropriate regression based on the scale on which the outcome variable is measured and its distribution, the same tree-based models can be used for both prediction of numeric values and classification of categorical labels—themselves based on numeric probabilities. Categorical data can refer to either dichotomous or multi-categorical data (mutually exclusive or multi-labeled, see Erlich et al., 2021; Verberne et al., 2014).

1.2.2 Unsupervised Learning

Unsupervised methods are useful when we have some dataset of X features, but no measurement of y. Thus, we're relying on the algorithm to reveal patterns that separate our data into meaningful values, e.g. clusters, we may conceive of as $\hat{y}$. The two main appeals of using unsupervised methods are (a) that they don't require labeled data, reducing the cost of building such a model; and (b) that they require fewer a-priori assumptions about the data

[6]Another advantage that is worth mentioning is that it is both highly customizable and very easy to apply in *R* or *Python*. Assuming small differences in performance, I value the accessibility and ease-of-use in research when working with these methods.

and they let the data "speak for themselves." Quinn et al. (2010) demonstrate this property in classifying senate speeches into topics to learn about the congressional agenda. But, the onus is then on the researcher to evaluate the algorithm's output and identify if said values represent anything that is theoretically meaningful, and if so, what.

Unsupervised methods can be used as standalone models to yield data-driven theoretical insights. One of the simplest, yet very powerful, examples of unsupervised models is K-means clustering. For a given dataset of columns (numeric variables) and rows (observations), this algorithm outputs clusters. Each observation in the dataset is assigned to a single cluster based on its similarity to other observations. The driving mechanism of the algorithm is the attempt to minimize the variance within each cluster (grouping together most-similar observations) and maximize the variance between each cluster (separating least-similar observations). The only input that the researcher must provide is the value of K, i.e. the number of clusters. Several data-driven methods exist for determining a correct value of K.

For example, Cavari & Freedman (2021) use K-means clustering to identify four tiers of affect in public opinion data, providing an empirically-based description of Americans' views toward the world, without making any prior assumptions about the data. As such, they are most useful for discovery (Grimmer et al., 2022), best summed up in the work of Wilkerson & Casas (2017, p. 533):

> "Grimmer & King (2011) demonstrate how unsupervised methods can lead to new discoveries. They find that congressional press releases cluster in ways that match Mayhew's (1974) typology of constituent advertising, position taking, and credit claiming, but they also observe an additional cluster they label 'partisan taunting' "

In Chapter 4, I illustrate a novel method of combining supervised and unsupervised methods within a single process, in this case aimed at measuring the policy topics of congressional bills. I use two unsupervised methods—K-means clustering and word vector representation—in the pre-processing of data to create useful features for the model. The

combination of supervised and unsupervised methods can be very powerful, as illustrated by G. King et al. (2017), who use trained models to classify documents into groups and then extract the keywords that best represent each group.

1.3 Other Types of Learning

The world of machine learning is as large as the datasets such methods are applied to. I do not intend for this review to cover all types of learning and I barely scratched the surface of supervised and unsupervised learning, mentioning only some of the more widely-known algorithms and emphasizing those I intend on using in my research. It is, however, important to consider other types of learning as well, including semi-supervised learning (combining labeled and unlabeled data, Zhu & Goldberg, 2009), reinforcement learning (in which a "learning system's actions influence its later inputs," Sutton & Barto, 2018, p. 2) or deep learning (in which models are composed of multiple hidden layers of abstraction of the data, LeCun et al., 2015).[7] Learning to navigate this world of models based on one's data, problem and desired solution, can be very conducive to optimizing results.

[7]Two particularly interesting aspects of deep learning are that (a) they have the ability to learn as they progress, making them very useful for online production systems, adapting to the population of data as they evolve; and (b) they can work with unstructured texts, rather than tokenized tabular data, making use of additional information stored in texts, such as the order in which terms appear, their grammatical role in a sentence, etc.

2 Learning to Teach (the Machine)

In the previous chapter I surveyed the literature to serve three purposes: (a) to illustrate that the volumes of data that political scientists are facing in their research are increasingly growing; (b) to review how, as in other fields, political scientists have adopted machine learning as a solution to provide fast, scalable and reliable measurements of data, on the path to scientific discovery; and (c) to provide a simple foundation of machine learning techniques, as an introduction to the chapters to come.

In this chapter, I outline the most important lessons I wish to add to this exciting body of literature. I make two types of contributions to this growing body of research. The first comprises of methodological lessons. Reviewing the literature on machine learning in political science and the vast world of online resources, first attempts at applying machine learning algorithms can be as disheartening as they are exciting. The ease with which we can use R or Python packages to make use of machine learning on our personal computer makes it all the more disappointing when results are not up to par. The four lessons I list below are hard-earned lessons I learned myself throughout this dissertation that the literature does not always prepare you for. For example, how do you train a model to identify an incredibly rare event? How do you balance the bias-variance trade-off in a multi-classification problem?[8] How do you test and/or prevent overfitting when your training data and unseen data are from two completely different periods?

This first type of contribution fits well under the title of "How do we get there?" The second type of contribution I make is what *there* actually is. In this case, the product is two new updated datasets to be used in congressional research. Following this chapter, I present two empirical chapters. In Chapter 3, I deal with a particularly difficult problem. I use the congressional hearings dataset—a dataset comprised of all hearings held in Congress in the post-WWII period—to identify hearings that relate to the creation of federal government agencies in unseen hearings data. The unseen data are in fact very old data—a collection of

[8] Some might wonder what does the bias-variance trade-off even mean! (stay tuned)

congressional hearings held in the period after the American Civil War and prior to WWII. What makes this problem challenging is that hearings on agency creation are very rare (only 1.5% of hearings in the post-WWII era discuss agency creation) and the fact that my training data and the unseen data are from two distinct periods, which vary in several ways, first and foremost, politically.

In Chapter 4, I take on the giant that is the congressional bills project. The dataset comprises of all bills introduced in Congress in the post-WWII era, leading up to the 114th Congress (ended January 3, 2017). The challenge here is to find a way to reliably code each bill into one of 20 major topics of the Policy Agendas Codebook, and one of over 220 minor topics. Several authors have published on the use of machine learning in the congressional bills project (Collingwood & Wilkerson, 2012; Hillard et al., 2008; Purpura & Hillard, 2006). It is extremely difficult to provide reliable predictions to this multi-classification problem and even the best efforts so far have converged at about 89% for the major topic level, and 81% for the minor topic level. Combining supervised and unsupervised methods, as well as extensive human validation, I was able to correct many misclassified bills in the current dataset and provide an updated version of it, including the 115th Congress (and am currently preparing the 116th Congress data).

I present both chapters as an empirical endeavour, highlighting the challenges each of the problems presents and the machine learning process I chose to overcome them. This process is a collection of methodological choices that bear consequences for both the product itself and its indented role in research. The chapters also serve as a hands-on introduction to several of the components and concepts in machine learning. Each of them ends with the product of a measurement, or dataset, as described above. Together, they also illustrate the four takeways that I believe will contribute to the good use of this method in political science. I outline them below.

2.1 Machine Learning as a Process

Machine learning—in congressional research or otherwise—is not *just* a method, it is a process of sequential, sometimes iterable, steps. Each step represents a methodological progression within an entire workflow. Progressing from one step to the next requires several decisions relating to how the algorithm of choice (itself a decision) is ultimately implemented. Describing it as a process opens the door for inductive discovery (Grimmer et al., 2021, 2022) and emphasizes that it isn't enough to think only about the desired product. When we use machine learning methods for creating a reliable measurement, we need to consider the data generation process, i.e. how we got there: "formal theory is central to modern data analysis [. . .] formal theory, specifically social choice theory, speaks to how the data with which the test will be carried out was created" (Patty & Penn, 2015, p. 95).

I illustrate a typical flow of the process of supervised learning (see Figure 2.1). We begin with defining a population of interest, from which we will sample a test set and one or more training sets. The test set is meant to serve as a set for assessing model performance. It, therefore, should meet three important conditions.

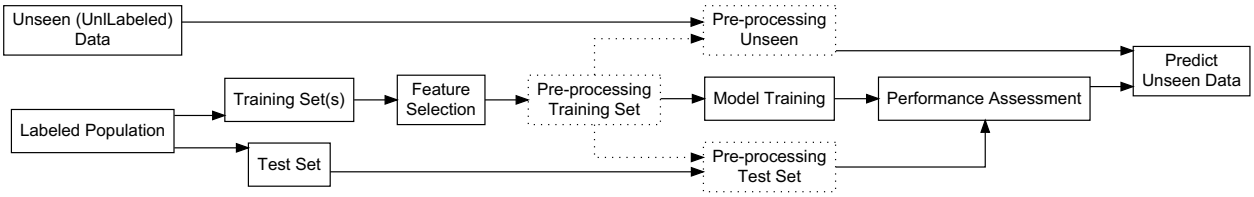
First, the algorithm should never learn from any information included in the test set because that would overestimate model performance. At its simplest, this means there should be no overlap of observations between a test set and the training data. More complicated overlaps may refer for example to time periods—if the data bear some chronological meaning, perhaps our training data and test set should be sampled from two distinct periods altogether. Additionally, feature values in the training sets should not be influenced by information stored in the test set.

Second, the test set must be labeled. Whatever quality we're interested in measuring, to properly asses how well our model is performing, we need to test its predictions against known data.

Finally, it should be a good representation of the data we're ultimately interested in predicting. The more similar our test set is to our unseen (unlabeled) data, the more valid

our test of the model's performance. If the test set is too different from the unseen data, it will serve a poor estimate of model performance and we may end up introducing mistaken predictions into our new data.

Figure 2.1: Typical Model Training Process



Once we've sampled our test set, we may sample one or more training sets. Textbooks often list 80/20 as the desired ratio between training sets and test sets (Raschka, 2015). While it is a good rule of thumb, understanding why it is a good rule of thumb makes using it, and in fact, deviating from it, much more efficient. A test set that is roughly 20% of the population is meant to convey that it should be sufficiently large to represent the population when testing model performance, but not too large so that we leave enough data to train on. The key is to have *as much* relevant information to train on while having *enough* relevant information to test on. In some cases, this might be achievable with a 90/10 ratio; others might require a 50/50 ratio. The point is not the actual ratio, but rather what goes into each set in a way that maximizes learning *and* allows a good assessment of model performance.

The remaining data after creating a test set may not represent the actual training set, but in fact the *training population*, from which we draw samples to construct meaningful, well-balanced and manageable training sets. Why not use the entire left-over population as a training set? Building a training set isn't about representing the population. It's about providing the machine the best cases to learn from. This simple understanding provides several motivations to use samples rather than the entire set, despite the obvious implication: A lot of data may be ultimately unused, essentially wasted.

First, some observations may be better suited for learning, providing a clearer distinction between classes, or at least the type of distinction the researcher is interested in making.

Time may even be a factor; e.g. it may be more useful to train a model predicting the policy topics of the 115th on only a handful of the congresses that preceded it, rather than going all the way back to the 80th Congress.

Second, sometimes, being true to the ratio of classes in the population may make it more difficult for the machine to find useful patterns. Balancing the ratio between classes may assist the machine in identifying useful patterns for making accurate predictions. We may opt, for example to have 50% of our data from one class and 50% from the other class (a 1:1 ratio, in which for every observation from one class we provide one observation from the other class), even though in reality, one class dominates 95% of the observations (19:1). Even if we reduce this extreme ratio to 4:1, we've controlled the ratio between classes in such a way that makes it easier for the machine to learn. Sometimes we might attempt to construct more than one training set, in an effort to compare the use of different observations, class ratios, sampling methods etc.

Finally, we may be limited in resources. A model training on 1 million observations will take much longer and require far more resources than a model trained on "only" 200,000 observations. If, using a fifth of the data can substantially reduce run-time, while maintaining the quality of the model, it may be a good enough trade-off.

Once we've selected our observations that compile each of our sets, we may select the features that go into our model. If observations represent a unit of information for the machine to learn from, features offer the machine a useful measurement (numeric, categorical, Boolean, etc.) to learn from; they are therefore a meaningful representation of a single dimension of the information stored in each observation.[9] Many machine learning algorithms, especially tree-based ones, offer advantages over standard statistical procedures, e.g. the ability to handle hundreds, even thousands of features, with varying degrees of correlation between them. While modelling such data isn't necessarily best practice, it is possible.

[9]One of the advantages of deep learning methods compared to supervised machine learning is that the latter requires the researcher to define and measure the features, while deep learning may rely on a more basic definition of features to reveal in hidden layers the features that comprise of the information stored in the data.

Several methods may reduce the dimensions (the number of features) that try to eliminate the problems such approaches introduce into the model.

Feature selection is about deciding what pieces of information to use as features, how to measure them and what to exclude from the model. The two cases I present in this dissertation illustrate some of the considerations of selecting features. Throughout the dissertation, I engineer features in my models from various data sources, combining text as data with non-textual data. In the first case, I use single terms from congressional hearings as features and combine them with several non-textual features relating to the Congress in which each hearing was held. I maximize the amount of features passed to the model but avoid categorical features (using one-hot-encoding for categorical features) due to the nature of the algorithm. In the second case, I use clusters of terms to form features to both reduce the number of features in the model and the weight of the single term. I also use a single categorical feature—easily handled by my algorithm of choice—but minimize my non-textual features in the model, to avoid creating endogeneity problems later down the line.

Every feature, be it based on text sources or otherwise, requires pre-processing. Several guides describe the pre-processing involved in text-based features, e.g. stemming words, removing stopwords, digits, punctuation etc. But even non-textual features require pre-processing. For example, how do we handle missing values? In categorical features we can pool them altogether into an "Other" category (although if in fact they represent several unknown categories, this may be a dubious decision). For numeric features do we impute some value such as the mean/min/max/median? Or use some more advanced method to impute the missing values? Alternatively, do we drop all observations with a missing value? Some algorithms have their own method of treating missing data to avoid losing information stored in such observations.

What students often don't take into account is that how we decide to pre-process our training sets has to inform both our test set and our unseen data. The algorithm simply fails to predict data that is not in the same structure of the set on which it was trained. It

won't provide incorrect predictions or even missing predictions—it simply returns an error. This prerequisite is important for two reasons. First, reproducibility is an issue even within one's own flow. If you don't keep track of how you pre-processed your training set, it is very difficult to usefully test your model's performance or apply it to unseen data. Second, with every feature you use, consider whether that information is available to you in your unseen data and whether it means the same thing. If you can't usefully measure it in your unseen data, your algorithm might prove useless in predicting your unseen data.

Finally, we've reached the model training stage. Choices here vary from the particular algorithm we choose (which has upstream effects such as minimal training set sizes, types of features it accepts, etc.), to how we apply our algorithm. For example, in a tree-based model how many trees should we use? 1,000? 4,000? 14,000? 13,912? How much weight should we allocate to the machine's learning from one tree to the next (more commonly known as learning rate)? These hyper-paremeters, indicate how the machine learns, and they are parameters that the machine itself can't learn on its own. Several methods exist for hyper-paramater tuning, ranging form manual experiments when data are insufficient to random or bayesian grids of parameter combinations. Model training also often involves trying to prevent overfitting to our training set, e.g. using cross-validation or a separate evaluation set (I illustrate each of these methods in the coming chapters).

As good as a model may seem on our training set, we don't have a good indication of its performance until we use it to predict known data in our test set. Several metrics exist for assessing model performance but even then, deciding what would be considered a good outcome in each metric is a matter of perspective—dependent on the research itself and the researcher's priorities. For instance, imagine a machine that has to sort ripe tomatoes from ones that aren't ripe yet, sending the ripe ones left and the not-so-ripe ones right. Is it more important to me to find *all* of the ripe tomatoes, even if it means also sending some not-so-ripe tomatoes to the left bin (false-positives)? Or is it more important to me to send *only* ripe ones to the left bin, even if it means sending some ripe ones to the right bin

(false-negatives)? In machine learning terminology, the question is one of *recall* (the former) vs. *precision* (the latter), which I cover more extensively in Chapter 3. Many of the metrics I dive into in the empirical chapters rest on the foundation of these two metrics—recall and precision—which together, represent the model's accuracy. We often aim to find a balance that maximizes the two, while meeting some minimal threshold for whichever of the two we prioritize.

Assessing model performance also introduces the question of the bias-variance trade-off. Suppose I examine the overall accuracy of my model's predictions on my test set and discover it's providing correct predictions for 75% of the data. Is this high or low? Satisfactory or won't do? As I mentioned before, these questions do no have one true answer and it often depends on the research itself and the researcher's priorities. For example, for a machine meant to identify breast cancer, 75% is very problematic. If the machine's job is to provide Netflix users with recommendations for series they might like, perhaps 75% is good enough. Setting aside this perspective, how would you asses 75% accuracy on the test set, if you knew your model was correctly predicting 98% of the data in your training set? And what if it were correctly predicting only 78% of your training data? In the former scenario, we might be very disappointed in our results because our model suffers from overfitting to our training data and high variance (much lower performance on our test set compared to our training set). In the latter scenario, we may still be disappointed because we're not doing we well as we hoped, but it appears our model has low variance (small difference in performance between our test set and training set) while suffering from high bias in both (high error rates).

Yu et al. (2008) provide an excellent empirical example of the bias-variance trade-off. The authors make use of party classifiers for congressional speech data. They find that "party classifiers trained on 2005 House speeches can be generalized to the Senate speeches of the same year, but not vice versa" (p. 33). In other words, training on house data has low variance when applied to Senate data because the bias (error rate) is similar in both; but suffers from high variance when the direction is reversed, i.e. training on Senate data, resulting in higher

error rates in the House data. They also find that their classifiers did better on data from recent years (compared to the training data) than on older ones, illustrating high variance rooted in time.

To avoid complication, the process I outline depicts a somewhat linear flow that moves only in one direction (recall Figure 2.1). The truth is, often we have to backtrack (e.g. realizing we need to pre-process our data differently to work with our algorithm of choice). Much more than that though, when assessing model performance, we may decide to return all the way to building new training sets, after understanding where our model is failing. Such backtracking is the iterable side of the process. The risk here is that if we keep using the same test set we may be informing our model based on insights from our test set, causing overfit to our test set. With sufficient data, it's often useful to create more than one test set, keeping one set separate from all the rest, which we use only at the very end, when we're confident enough in our model.

Only once we've gone through this entire flow and we've reached a model we approve of, can we use it to predict new data. This step should also include some assessment of model performance, e.g. by examining samples of its predictions. Note how the most important part of this exercise where we predict labels in our unseen data—the goal itself of the entire endeavor—consists of the least effort. In fact, our entire effort in building a good model occurs in the several steps that precede it and they represent a myriad of decisions, each one affecting the ultimate outcome.

Thinking of this process as a black box that receives some minimal-effort input and magically produces outstanding output is misleading and often results in sub-par results. Instead, understanding and conveying to the reader all of the choices that went into this process usually provides a much better outcome and helps avoiding pitfalls in research. Consider the excellent guide put forth by Barberá et al. (2021) on the choices scholars are required to make when using text as data in a machine learning context. Examining different sampling methods, the use of keywords vs. a-priori category association and defining

sentences vs. article segments as units of analysis affect which data the model is trained on, how it trains and what its outcomes eventually are.

In this context, I highlight the centrality of questions such as what performance metrics the author prioritized; how research goals influence these priorities, and in turn, the outcome; what features are included in the model and how they might relate to the research at hand (do they increase the risk of endogeneity?); or what is the inherent risk of overfitting our model to the training data. Thinking about the choices we face when approaching such problems and the consequences they have for our theoretical research (Denny & Spirling, 2018) is an important lesson I discovered through this dissertation. Be it theoretical considerations or methodological ones, these choices can have substantial consequences for subsequent research that rely on these data.

2.2 Text as Data

My second contribution is an important clarification about the meaning of "text as data." Natural language processing is a field in and of itself with amazing developments. Most recently, the GPT3 algorithm is able to write artificial documents based on a training set, or provide accurate summaries of existing ones. Similarly, Github Co-pilot is an advanced natural language algorithm that completes programming code for the user as they type.

One of the most applied methods of machine learning in political science is the use of text as data (Cardie & Wilkerson, 2008; Slapin & Proksch, 2014) and almost all of the papers cited in this dissertation make use of text as data. Examples include measurements of the congressional agenda through legislation (Collingwood & Wilkerson, 2012; Hillard et al., 2008; Purpura & Hillard, 2006; Quinn et al., 2010), of ideological positions or party membership through legislative speeches (Diermeier et al., 2012; Yu et al., 2008) of topical themes through the analysis of election manifestos (Verberne et al., 2014) or the analysis of agenda and tone in news articles (Barberá et al., 2021; Boydstun, 2013).

Much of the work that has made use of textual-data in machine learning often rely *solely*

on textual data. They can easily be misinterpreted to suggest that text-based models can *only* rely on text. Instead, I encourage researchers to think of text as simply another source for measurement. Through empirical analysis I illustrate that a single model consists of features. Each feature may be constructed using different sources. Features are meant to capture useful information for making accurate predictions. Some features may rely exclusively on text (e.g. the number of times term a appears in a congressional hearing title), others may rely on information from non-textual sources (e.g. the party that controled Congress when a particular hearing was held) and some may combine both pieces of information (e.g. measuring separately the number of times term a was mentioned by members of each party in a particular hearing). The advent of big data and the methods that go with it, introduce a variety of data types (Monroe, 2013) that can be used together, rather than separately, for empirical inquiry.

2.3 Know Your Machine's Strengths & Limits

While I am obviously advocating for the strengths of machine learning models, I also place an important emphasis on model error. On the one hand, ignoring model error may result in systematic mistakes in measurement, and subsequently in theory-testing. On the other hand, students are often disheartened when several first attempts at using such promising methods provide sub-par results. Understanding where errors are coming from can be useful for a well-executed machine learning process. Doing so may allow improving model performance to maximize the metrics we prioritize through data correction, model parameters and if necessary, increasing algorithm complexity. It may require us to redefine the very metrics we prioritize, and it can help in identifying the limits of our model, maximizing the benefit of the model while minimizing the need for human intervention/review.

The question at heart is one of validity: How do we know a machine's prediction is a good measure of what we're interested in (Monroe, 2013)? I adopt the agnostic approach of Grimmer et al. (2021) in that I am not aspiring to find a model that represents some

unquestionable truth, but rather I am looking for a model that can yield good predictions.

Even when we overcome several root causes of model errors, and train an excellent model, every model has its limits. The best model, trained on the most reliable of data and using the most sophisticated method still makes *some* incorrect predictions when applied to new data. The difference between the data we use to teach our model and the unseen data to which we apply our model often reveals overfit. Hajare et al. (2021) provide an excellent example. In their study, they make use of a pre-labeled dataset of political speeches made in Congress to identify political bias in a separate, unrelated dataset, of social media posts from Twitter and Gab. Their method illustrates maximizing a machine's ability to provide accurate predictions of 70% of the unseen data, thus using the machine to provide measurements for a large majority of the data, while being of aware of its limits to avoid carrying mistakes forward.

Such limits may be rooted in different causes. The case study of the congressional bills project (Chapter 4) illustrates two particular types of sources. First, When our training data are inconsistent and may themselves include a large share of misclassified observations, our model does poorly—it either replicates mistakes going forward, or finds it difficult to learn meaningful patterns because of contradictory relationships. Second, the theoretical definitions of the policy agendas codebook often result in overlapping terms used in different combinations and contexts. Most of these, even 80-90%, can be overcome with a good model, but it undoubtedly yields at least 10% of observations that confuse the machine. The bills project also presents a rather extreme case of this type of problem because of my use of a machine learning algorithm for a multi-classification problem. Models are complicated enough when they are required to make only binary distinctions. Here, we are asking the machine to learn how to distinguish between 20 different major categories and the 11 (on average) minor categories within each major category.

The agenda-creation case study illustrates two more extreme problems we need to address. First, an extremely unbalanced class distribution, in which the event we're interested in identifying is extremely rare—less than 2% of our training population and evidently, even

rarer in the unseen data. Second, the data used to train the model (congressional hearings in the post-WWII era) are markedly different from the unseen data (congressional hearings pre-WWII) both politically and linguistically.

Even so, one of the most interesting advantages compared to the use of human coders that I discovered, was the ability of machine learning to provide reliable predictions across time. While scholars are usually extremely prudent about horizontal intercoder reliability (i.e. better than random agreement between two or more coders in a single batch of documents), we do not always guarantee intercoder reliability across time, for example when relying on a changing cohort of research assistants for coding data every year. Even with the best training, this can result in inconsistency in coding patterns from one year to the next.

A well-trained model, in some respects, can be more reliabile than human coding. While researchers in the field, and PAP in particular, go to great lengths to train human coders and to ensure inter-coder reliability, two targets remain difficult to achieve. Human coders are susceptible to heuristics. In this case, the most recently available information they were exposed to may influence human coders (Tversky & Kahneman, 1973), especially with so many categories to choose from. With so many related categories in a single codebook that aims to be comprehensive and mutually exclusive (B. D. Jones, 2016), coders may often code based on the categories most readily-accessible to them.

Moreover, ensuring backward compatibility is a very difficult task. Whether the same coders code the data over time, or coders change with the passage of time, both encompass a risk of inconsistency. In fact, training a model for this study revealed this very problem in the data and the topics of over 10,000 bills had to be manually corrected before a model was able to accurately learn useful patterns from the data.

Finally, models offer additional information that may be close to impossible to ask of human coders. For instance, when assigning one of twenty major policy topics to a particular observation, the machine produces a probability for each topic. I use this probability to identify a threshold above which the machine's classifications appear reliable without requiring

human intervention. No doubt such probability distributions may have additional uses. For a human coder, providing reliable measures of certainty of their decision would be incredibly difficult and time-consuming. Machine learning decisions can also be reverse-engineered to understand how a model reached its classification and yielded these probabilities. Reverse-engineering can be useful in understanding mistakes, justifying decisions and improving performance in the future.

2.4 Maximize Performance by Combining Methods

Finally, I provide a useful demonstration that combines supervised and unsupervised learning for dimension reduction and maximal model performance. Big data and machine learning models often suffer from having too many features and researchers often use various methods of reducing the dimensionality of the data (e.g. principal component analysis or feature selection) to lower the amount of features in a single model (Grimmer et al., 2021; Patty & Penn, 2015).

With "text as data," dimensionality becomes even more complicated as each word in each document can become a feature of its own creating a model with thousands of features, and standard methods for dimension reduction do not apply here. Moreover, relying on a single word can limit the effectiveness of a single feature. Consider the terms 'gas,' 'fuel' and 'petrol.' All three terms may be synonymous in certain contexts, yet predicting the topic of an observation based only on the occurrance of the word 'gas' may result in missing some relevant observations that used the words 'fuel' or 'petrol' instead.

The method I adopted in Chapter 4 used an unsupervised method to group related/similar terms together to create features based on groups of terms, thus reducing the dimensionality of the data and yielding better features. This method also has the advantage of succeeding even when proponents change the terms they are using in an attempt to reframe the issue (Baumgartner et al., 2008; Gamson & Modigliani, 1989; Jacoby, 2000; Nelson, 2011). Then, I used a supervised method to learn patterns in the data for predicting policy topics of

congressional bills.

The next two chapters serve as empirical use-cases of machine learning and illustrate the four lessons outlined above.

3 Congressional Hearings on Agency-Creation

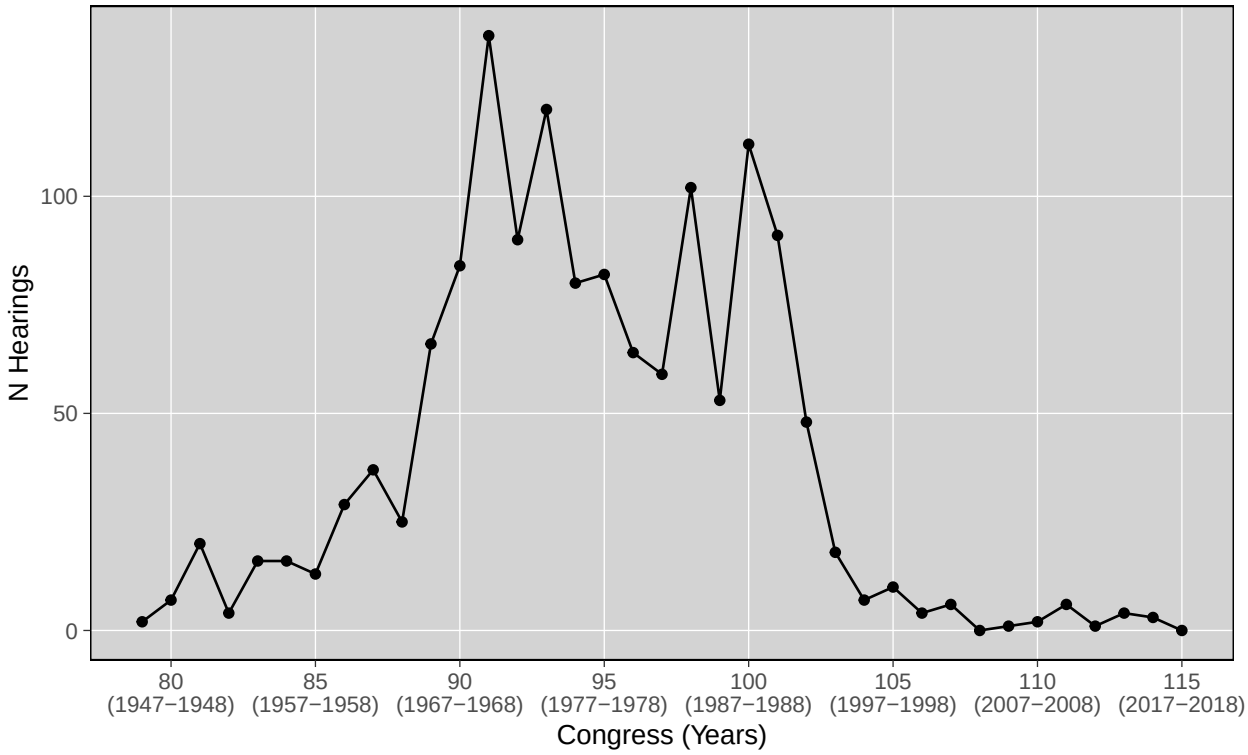
3.1 The Problem

In the middle of the 20th Century, the American political system underwent a huge transformation, expanding its authority, involvement and policy scope (Pierson & Skocpol, 2007). In a recent study, B. D. Jones et al. (2019) reveal the causes and consequences of this great broadening of government. In their research, the authors offer a myriad of empirical evidence to demonstrate how this expansion of government manifested itself in the congressional agenda. Among other datasets, they rely on the dataset on congressional hearings (hereafter: modern hearings dataset) at the Policy Agendas Project (PAP) that collects all hearings held in Congress since the 80th Congress (1947, N = 100,942).[10] Among the many indicators in the set, is a binary indicator for hearings that discuss the creation of a new federal government agency. I plot the number of hearings in each Congress that discusses agency-creation according to the modern dataset in Figure 3.1. According to B. D. Jones et al. (2019), the major increase starting in the 1960s and ending in the 1990s, lines up with our understanding of the expansion of the federal government during this period.

The modern hearings dataset has afforded scholars an immense contribution to congressional and policy studies, providing empirical evidence for the development of policy-making in Congress. For instance, several studies highlight the increasing fragmentation—and erosion—of committee jurisdictions in Congress (Adler & Wilkerson, 2013; Baumgartner et al., 2000a; Baumgartner & Jones, 2015; B. D. Jones et al., 1993b). Others describe how committees collect and process information and their effect on policy (Fagan & Shannon, 2020; Lewallen et al., 2016) or the relationship between its agenda and economic inequality (Epp, 2018). This dataset has also provided insights into Congress's transition from a law-making body to one that is more concerned with oversight of its bureaucratic agents (e.g. Lewallen, 2020; McGrath, 2013), as well as an indicator of the degree to which congressional attention is

[10] At the time this study was carried out, the modern hearings dataset spanned 1946-2017 (79th- 115th Congress). It has since been extended to 2020, including the 116th Congress.

Figure 3.1: Number of Hearings Discussing agency-creation Post-WWII



Excluding 8,345 hearings with missing values.

representative of mass preferences (B. D. Jones & Baumgartner, 2004).

The newest addition to the PAP collection of datasets covers congressional hearings spanning the 40th Congress (1848, following the end of the American Civil War) all the way through to the 80th Congress (hereafter: the old hearings dataset, N = 30,338), when the modern dataset begins. Human coders coded each observation for the relevant major and minor policy topic.

The old hearings dataset may provide some insight into the question of when Congress began holding hearings that included agency-creation, the extent to which Congress held such hearings and whether evidence exists of similar broadenings in America's history. To do so requires an indicator for whether each hearing discussed the creation of a government agency or not, much like in the modern hearings dataset.

The congressional hearings datasets offer the potential for both insight into the above political question and an opportunity for methodological innovation. The modern hearings

dataset serves as a population of nearly 100,000 observations that have already been labeled for the creation of government agency. In a supervised learning context, we may therefore sample from these data to train a model (or several) for classifying agency-creation. Learning based on which hearings discuss agency-creation, we may train a model to classify hearings in the old hearings dataset. If successful, this method may save hours of human labor and provide us with a reliable and consistent measure of agency-creation in congressional hearings across time.

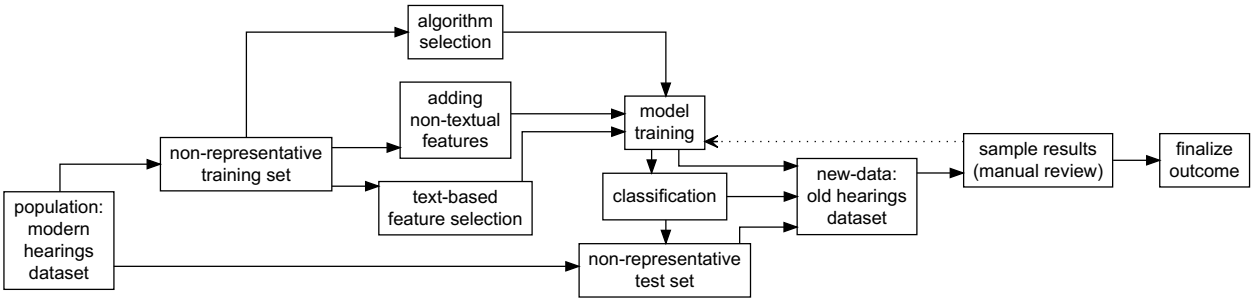
3.2 Model Training Strategy: An Overview

The desired outcome of the method implemented here is a single, reliable column to be added to the old hearings dataset. This column should indicate whether each hearing in the old hearings dataset references the creation of federal government agencies. Ultimately, researchers may use this column as a dependent or independent variable in an inferential setting. Here, it is the outcome of a structured process, using machine learning algorithms to correctly identify the values of this column (Figure 3.2).

After identifying the modern hearings dataset as the population of data I wish to train and test my model on, I sample the population to create two sets — a training set and a test set. Training sets are often structured to represent a good sample to learn from (e.g. by balancing the classes of interest), instead of representing the population from which it is sampled. Usually, we prefer our test sets to serve as a good representation of the population. Here, I opt for a non-representative test set of the population because the two classes are severely imbalanced—only a rare minority of the observations reference agency-creation (I discuss the extent and implications of this imbalance later in the chapter).

Training the model includes several steps. First, pre-processing the data, which in this case means creating a document-term matrix (DTM) such that each row (observation) is a hearing and each column (feature) is a term from the hearing's description (see an illustration below). Values indicate the number of times each term appeared in each hearing. I avoid

Figure 3.2: Model Training Strategy



weighting the values using TFIDF or any other weights because of how few terms each document is comprised of (in this case, a document is a hearing's description).[11] Because the descriptions are short, dimension reduction to reduce the number of features in the model is also not possible (each document is comprised of a small list of terms that appear together with only a small number of other terms in other observations). I also find no justification for scaling or standardizing the features necessary because they're all on an almost identical scale.[12]

Second, I stem terms (pooling together variations of the same term into a single feature), remove stop words and remove sparse features to avoid overfitting (by excluding rare terms) and to reduce the number of features in the model. Third, I compare the performance of two appropriate algorithms (random forest and gradient boosting). Fourth, I manually tune some of the best model's parameters. I avoid hyper-parameter tuning because the sets that I can use for training, validating and testing my model are very small. In addition, the problem is relatively simple and I manually review samples of the classification results as a feedback mechanism. Finally, I examine the improvement in accuracy when including several congress-related features that are not drawn from the hearings' descriptions, and are therefore

[11] TFIDF or Term-Frequency Inverse Document-Frequency is a method of weighting the frequency of which a term appears relative to the number of documents it appears in.

[12] All text-based features count the number of times a term appears in each hearing's description. Descriptions are usually quite short and it is rare for the same term to appear in the same description more than once. Thus, most features are binary variables and only rarely have a count greater than one. Most of the non-textual features I add to the model are binary and only a handful are continuous and/or scaled. Nonetheless, I empirically test applying TFIDF weights and standardizing the features using z-scores and neither improved model performance.

not text-based.

Text-based features are obviously important but they are not generated based on theoretical expectations. Instead, they rely on some corellational pattern, representing either a fake or real relationship with the class of interest and that are able to predict that class with some degree of accuracy. Including theoretically-meaningful features may improve performance. Together with the text-based features, they may improve prediction because, due to the interactions embedded into the splits of tree-based algorithms, terms are now analyzed in a particular context (e.g. some terms may be more meaningful when both chambers of Congress are controlled by the same party).

I classify the test-set using several training sets. As I describe below, the data are too limited to create different samples and so the same observations are used in all training sets. The variation between them rests in the algorithm chosen, the model training parameters (specifically, number of trees, depth and learning rate) and the addition of non-text based features.

Classifying any set also requires pre-processing in the same manner as the training set. First, it must be converted into a DTM and terms must be stemmed. Second, features that appear in the training set but not in the test set, need to be added as columns to the DTM, imputing some relevant value (in this case, 0, representing the term does not appear in any of the observations).

Once I identify a model with sufficiently good performance on the test-set, and the best of all those examined, I use it to classify the unseen data—the old hearings dataset. Since unseen data are in fact unlabeled data, I have no way of assessing the performance of classification. If anything, I am using the model to provide me with an indicator I do not currently have in the unlabeled data. Therefore, I sample the unseen data based on the probabilities that the model produced. My main concern is recall or true-positives (correctly identifying as many agency-creation hearings as possible) and removing false-positives, while simultaneously keeping them at a minimum. Rather than manually coding all 30,000 observations in the set,

the model's performance was good enough, that I required to sample and review only about 11% of the data (a little over 3,000 observations).[13]

Based on the samples, it is possible a retrain would be required (note the backward dashed line in Figure 3.2), especially if recall is low. In other words, if the model incorrectly classified certain hearings as non-agency-creation. The advantage of a retrain at this stage, is that I could train based on the samples of the unseen data. The samples, previously unlabeled, are now labeled, having been classified by the model and manually validated. These data represent the most relevant data for both classes from the old hearings dataset, because they include many of the strongest examples of agency-creation observations and they have been subject to human review.

Unfortunately, I could not start with a random sample of the old hearings dataset to manually code and train on it, because a random sample will likely have too few examples of the class I am most interested in—agency-creation. Thankfully, results of the first model were good enough and did not require a retrain. Recall was high, false-positives were easily removed and the few false-negatives that remained were easily corrected in a manual search. Ultimately, I am left with the final outcome: A binary agency-creation column, to match that which exists in the modern hearings dataset.

3.2.1 Document-Term-Matrix: Illustration

In Table 3.1 I sample two hearings from each class of agency-creation (1 representing agency-creation). The table also lists the meeting of Congress and year in which each hearing was held, as well as the description of the hearings. The first column is simply a unique internal id associated with each hearing.

[13]The sample was double-coded by me and a second coder. A special thank you to Iynkary Vigneswaran Warr for her assistance in reviewing the sample.

Table 3.1: Sample Hearing Descriptions

ID	Congress	Year	Agency Creation	Description
35859	93	1973	0	Safe transportation of Hazardous materials by air.
37155	92	1971	0	Federal government's role in the achievement of equal opportunity in housing.
84213	106	1999	1	To consider bill to establish a Bureau of Immigration Services and a Bureau of Immigration Enforcement within Dept. of Justice.
99426	112	2011	1	To establish within the Department of Interior a new department to consolidate responsibilities.

In Table 3.2 I provide an illustration of the same four hearings in the form of a document-term-matrix. Note, for presentation purposes only, I split the table into three. Terms have been stemmed and stopwords have been removed. For the sake of illustration I have not removed sparse terms. Each row still represents a single hearing and each column represents a term from all terms in the descriptions of all hearings—these are potential features to be used in a model. The values represent the number of times each term appears in each hearing.

Table 3.2: Document-Term-Matrix

ID	achiev	air	bill	bureau	consolid	depart	dept	enforc
35859	0	1	0	0	0	0	0	0
37155	1	0	0	0	0	0	0	0
84213	0	0	1	2	0	0	1	1
99426	0	0	0	0	1	2	0	0

(Table Continued)

ID	equal	establish	feder	govern	hazard	hous	immigr	interior
35859	0	0	0	0	1	0	0	0
37155	1	0	1	1	0	1	0	0
84213	0	1	0	0	0	0	2	0
99426	0	1	0	0	0	0	0	1

(Table Continued)

ID	justic	materi	opportun	respons	role	safe	servic	transport
35859	0	1	0	0	0	1	0	1
37155	0	0	1	0	1	0	0	0
84213	1	0	0	0	0	0	1	0
99426	0	0	0	1	0	0	0	0

3.3 Challenges for Supervised Learning

The problem at hand, and the algorithm I employ, are relatively simple. I could settle on a model that uses information stored only in the hearings' descriptions, i.e. terms. But, as I show later in the chapter, such a model can be substantially improved by accounting for additional information we know about each hearing and the political environment in which it was held. Ignoring this information might lead to sub-optimal performance because I am not accounting for theoretical (and empirical) differences that change how each term from the hearings' descriptions is related to agency-creation. That said, these theoretical differences also pose severe challenges and not all of them can be fully addressed. In fact, attempting to address some of them might worsen model performance and lead to misclassification due to overfitting my model. In other words, I risk training a model on patterns that are true only for the modern hearings dataset and mistakenly applying them to the old hearings dataset. I outline these challenges and my attempt to address them below.

3.3.1 Old vs. New Linguistic Features

The model I trained relies heavily on the language associated with the hearings. Specifically, terms used in the hearing's brief summary become features in my model.

Language is an incredibly dynamic form of documentation and communication. It is constantly changing (Aitchison, 2001; Buerki, 2019). New terms continuously enter a language as old ones slowly become extinct. Grammar too tends to change over time. With the advent of social media, new methods of short-hand typing and acronyms have entered our linguistic arsenal of tools. History tells us that even how we say things has changed (Wolfe, 1972). Even in short periods when language itself may be relatively static, the advent of technology has greatly increased the scope and volume of words that citizens may be exposed to, e.g. through the media (Pool, 1983).

Linguistic evolution thus presents a difficult challenge to the method proposed here. I am attempting to train a model on hearings' summaries from the last 75 years (1947 onward),

to classify newly collected data from the 80 or so years prior. Not only that, but it is likely that within each of these two datasets, language has evolved, introducing a great deal of variance into the training set, test set and the unseen data. The main question is how quickly congressional language has evolved and has it evolved too quickly for a model of this type to work.

The question is more complex than the challenge that language evolution presents on its own. As I describe in the next section, Congress as an institution has developed. The size and allocation of staff has changed, with meaningful effects on policy (DeGregorio, 1994; Ornstein et al., 2009; Salisbury & Shepsle, 1981; Schiff & Smith, 1983). So have congressional practices, e.g. in documenting hearings material. This is especially true with the advent of technology, which has made it much easier to document such material. Moreover, these datasets present a data-generation question. The hearings' descriptions used in the modern hearings dataset were largely written by students, trained by PAP to review a congressional summary of the hearing and write out a brief description. However, the old hearings dataset is comprised of descriptions provided by Proquest Congressional, which might follow different guidelines when writing out such summaries.

Despite these potential problems, linguistic differences between the two datasets are quite small. Of the 1,085 terms used in the final model, only 64 did not appear in the unseen data. However, the unseen data include an additional 488 terms that are not part of the model (after removing sparse terms, with sparsity set to 0.9995). The information stored in such terms is essentially ignored. For comparison, only 1 of the model's terms did not appear in the test set. These numbers suggest changes in language over time make it more challenging for a model like this to perform, but it is not too severe. Nearly all of the model's text-based features do in fact exist in the new data and can therefore be used. If the periods examined were significantly longer, this may have become a bigger problem.

3.3.2 Old vs. New Political Environment

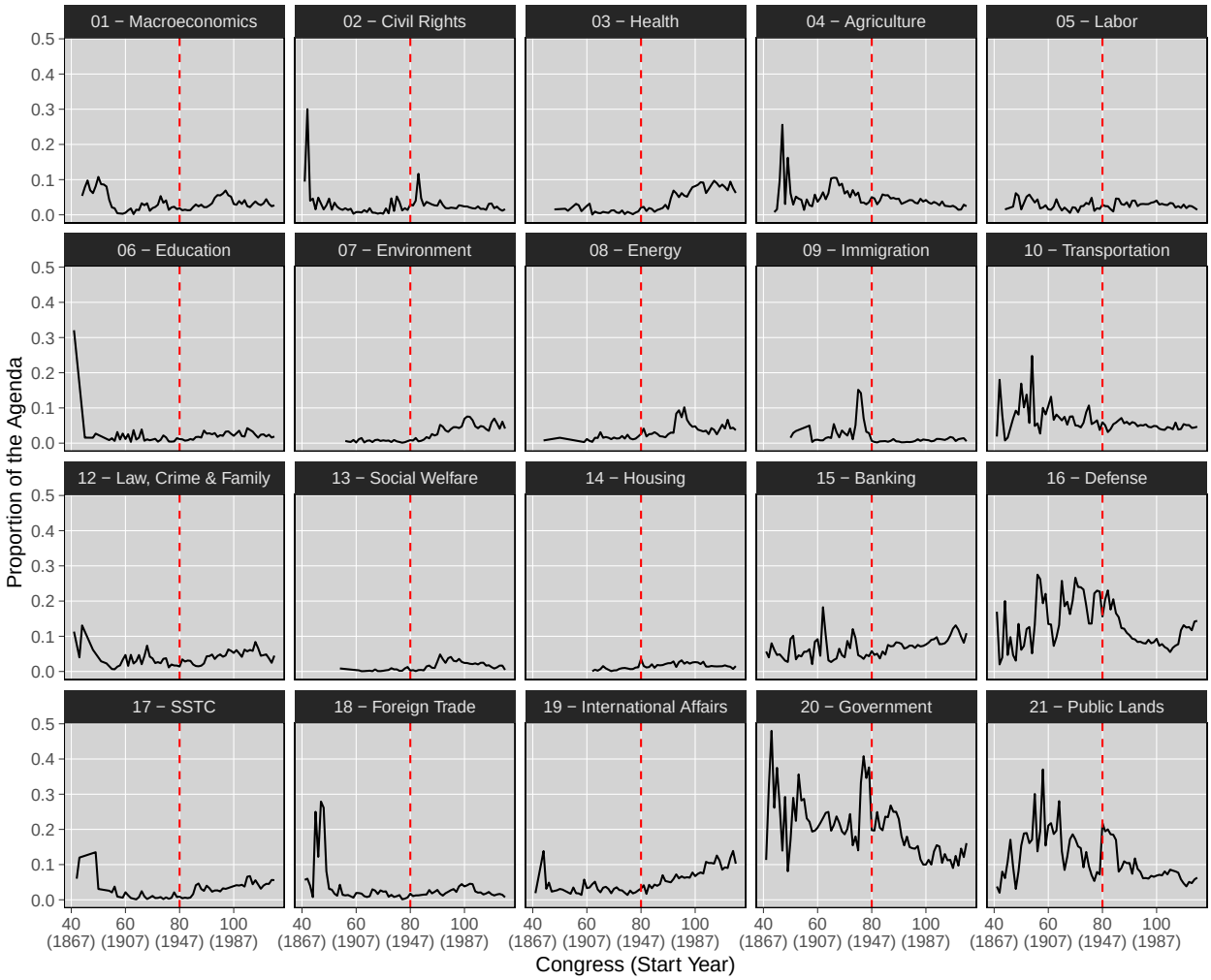
It is very difficult to compare the period of 1868-1946 to that of 1947-2021. In many respects, the entire political system has changed. The issues that were of concern in Congress in the earlier period have mostly disappeared from the agenda, replaced by new issues. The congressional structure, namely its committee system, has been reformed several times throughout these periods. Committee jurisdictions and power over determining the agenda are incredibly dynamic. The parties operating within Congress represented entirely different agendas, positions and voters and partisan polarization has changed dramatically over time. Each of these contribute to the challenges of relying on information in modern hearings to identify similar information in older hearings.

3.3.2.1 Agenda Figure 3.3 plots the congressional hearings' agenda throughout the entire period examined. For each topic, I plot the proportion of the agenda it occupies in each meeting of Congress. The red dashed line marks the transition from the old dataset to the new one and offers a crude distinction between two very different policy agendas.[14] Very few of the issues maintain a consistent proportion of the agenda throughout the entire period. Most that do—e.g. labor, education (ignoring the unusual spike in the 40th Congress, largely driven by a small number of hearings), social welfare, housing and macroeconomics—usually amount to a small proportion of the agenda.

Other topics are usually more dominant only in a single period. Government operations, perhaps the most dominant topic of all, is far more dominant prior to the 80th Congress, than after it. Many of these hearings in the first period related to elections, investigations into corruption charges made against Representatives and Senators, federal buildings, etc. Public lands and defense follow a similar pattern. Public lands mostly relating to Alaska, sale and

[14]The collection of hearings in each dataset is not perfect; some hearings from the 79th hearings appear in the modern dataset and some hearings from the 80th hearings appear in the old dataset. This is largely an error in data collection that was outside of my control and I therefore treat the 80th Congress as the start of the modern dataset, especially given the changes in Congress that began in the 80th Congress, namely the restructuring of the committee system through the Legislative Reorganization Act of 1946 (Evans, 1999).

Figure 3.3: The Congressional Hearings' Agenda



Not displayed: 23 (Arts, 1 hearing) and 99 (Other, 11 hearings).

acquisition of land, and Native American territories ("e.g. Dividing Portion of Reservation of Sioux Nation of Indians, in Dakota, into Separate Reservations, and Securing Relinquishment of Indian Title to Remainder"). Defense was heavily influenced by wars the U.S. was involved in at the time, the two world wars and the development of the military and navy.

Other issues were less dominant overall, but follow a similar pattern: health was hardly a congressional concern in the first period and has consistently become about 10% of the agenda in recent decades; environment and energy have both become nearly 10% of the agenda in the second period after being non-existent in the first period; agriculture was far more prominent in the first period, as was transportation (the railroad system underwent massive development in this period); and several other issues, e.g. immigration and foreign trade spiked for a short while in only one of the periods.

The implication of a changing agenda is that the *types* of agencies under consideration may change. The more they change, the more difficult it may be for a model to correctly identify agencies in new data. Consider for example, a model trained on agencies relating to civil rights, social welfare or foreign trade and its performance on a set comprising mostly of agencies relating to land regulation or military reorganization. Fortunately, as B. D. Jones et al. (2019) demonstrate, the number of subtopics on the agenda has only increased with time. In other words, the training data from which the model learns, includes examples of new agencies from *more* topics than previously were on the agenda. The model is therefore more likely to encounter an agency in the unseen data from a topic it has already learned from, rather than one it has not.

To strengthen the model's predictive ability, in the final model, which includes non-textual features as well as text-based features, I add two indicators relating to the composition of the agenda. Accounting for the specific topic of each hearing might be too complicated for a model like this, and may even lead to overfitting. Instead, the indicators I include represent how dominant the policy topic of a given hearing is on the entire agenda of a given meeting of Congress:

1. A scaled version of the number of hearings in the entire meeting of Congress that share the same subtopic as the hearing in question. Labeled *subtopic_count_scaled*.
2. The proportion of hearings in the entire meeting of Congress that share the same subtopic as the hearing in question. *prop_subtopic*.

3.3.2.2 Congressional Development The agenda, and the part of the agenda that is dedicated to agency-creation, are a product of the institution itself—it's *modus operandi*. Congress has changed in several important ways that may affect the success of a model predicting agency-creation: The development of the committee system; differences between the two chambers; the role of party leaders; who the parties are and what they represent; and party polarization in Congress. I address the latter two of these in the next sections.

Schickler & Bloch Rubin (2018) offer an excellent review of these developments and how they coincide with congressional research. Here, I am interested specifically in theorizing about the effect such developments might have on model performance and providing possible solutions.

Both datasets record hearings after the creation of a system composed of specialized standing committees, replacing the previous system of temporary select committees (Cooper, 1988). Unfortunately, that is about all the two datasets share considering committee structure and each dataset contains meaningful changes within the period they cover. For example, prior to the 1910-11 revolt against Speaker Cannon, committee assignments on standing committees pointed to high turnover. The seniority system adopted in the early 20th Century resulted in a more stable record of committee assignments (Abram & Cooper, 1968; Canon & Stewart, 2001; Polsby et al., 1969).

The seniority system affected the appointment of committee chairs, who for a brief period controlled the committee's agenda (e.g. Fenno, 1966), but it only lasted till the reforms of the 1970s. The restrengthening of party leaders in Congress (Rohde, 1991; Zelizer, 2006), especially after the reforms of the 1970s, came at the expense of the chairs' power in

determining the agenda (Cohen, 1999; Mann & Ornstein, 2006). More generally, the dynamic nature of the Speaker of the House's power in itself points to substantial changes in Congress as an institution (see Bloch Rubin, 2013; Cooper & Brady, 1981; Schickler, 2001; Schickler & Bloch Rubin, 2018 for a review of the changes bolstering the Speaker's power leading up to the 1910-11 Cannon revolt, and the reemergence of the Speaker's power following the 1970s reforms).

Beyond the question of who wields greater power in determining the congressional and committee agenda, perhaps the most important distinction between the two datasets is the Legislative Reorganization Act of 1946—right at the end of the old hearings dataset and the beginning of the new hearings dataset. The act completely redefined committee jurisdictions through official rules (Evans, 1999), affecting the topics each committee had power over, and further bolstered the control of committee chairs (Davidson, 1990; Deering & Smith, 1997), at least until the reemergence of the parties. The official rules changed again in the 93rd Congress (1973-1974), with the Bolling committee (see Adler & Wilkerson, 2008, 2013; S. S. Smith, 1986; Strahan, 1988 on the changes these reforms introduced and their effect on policy-making in Congress). Furthermore, several studies have highlighted members' attempt to expand their committee jurisdictions by holding hearings on topics that may be considered outside their scope, thereby setting an important new precedent for jurisdiction (Baumgartner et al., 2000b; B. D. Jones et al., 1993b; D. C. King, 1997; Lawrence, 2013; Sheingate, 2006). Thus, the relationship between the agenda and the committee structure changed between the datasets and within them.

To summarize thus far, the first half of the old hearings dataset is characterized by strong speakers and party leaders and it more closely resembles the second half of the modern hearings dataset. In the period in between, covering the second half of the old hearings dataset and the first half of the modern hearings dataset, parties and their leaders were weaker while committee chairs wielded greater power. Moreover, committee jurisdictions are dynamic, changing both through official rules and through legal precedents determined

by the hearings themselves. These changes separate the two datasets from one another and occur within each one.

The implication of the way in which Congress has developed is that who determines the agenda may be markedly different between my training population (modern hearings dataset) and my unseen data (old hearings dataset). Thus, the types of agencies they are interested in creating, as well as their general tendency to discuss agency-creation, may be different. The extent to which the committee system has changed, which committees exist and what falls under their jurisdiction, makes controlling for them in a statistical model irrelevant. The same committees do not operate in the two datasets and what they are responsible for changes from Congress to Congress. In fact, assuming a relationship between the committee holding the hearing and the topic of a hearing will likely lead to misclassifications and may prevent us from revealing fragmentation in committee jurisdiction. As for the implications of the changes that the two parties within Congress experienced, I address these in the next sections.

The only valid feature I could add to the model at this point is the extent of power the Speaker of the House has but this suffers from two main limitations. First, I am unaware of a good quantitative or ordinal measure of Speaker power. Thus, I would resort to a binary measure of high/low power (1/0 respectively), separating into broad periods (e.g. 1890-1910 would be considered high). Second, there's no theoretical justification for a relationship between Speaker power and agency-creation. Power may affect the topics themselves on the agenda (which I already account for as described in the previous section), but the effect is less likely with agency-creation. I therefore refrain from including such a feature in my model.

Before turning to the role of parties and the challenges they create for model performance, I must address one other aspect of Congress: Bicameralism. Fundamental to its nature as an institution, is the fact that it is composed of two separate chambers, with different compositions, responsibilities and modes of representation. Two chambers often represent different interests (Llanos & Nolte, 2003; Patterson & Mughan, 2001; Riker, 1992; M. Russell,

2001; Tsebelis & Money, 1997), and therefore the House & Senate might diverge in their approach to agency creation and the interests that serves. Bicameralism in Congress represents a relatively stable set of differences rooted in the constitution (Lee, 2018) and relevant in both periods. The chamber in which the hearing was held, could therefore be meaningful for the likelihood of agency-creation. Thus, using one-hot-encoding, I add to the model two indicators for whether the hearing in question was held in the House or whether it was a joint hearing (0 in both indicators represents a hearing held in the Senate).

3.3.2.3 Political Parties Although the parties' names have remained the same since the mid 19th Century, the two parties themselves have been anything but stable (Silbey, 2010). They have realigned around both issues and voters (Key, 1959), creating several party systems in American history (Brewer & Stonecash, 2009; Burnham, 1965; Sundquist, 1983). For example, in many aspects, Lincoln's Republican party more closely resembles today's Democrats on issues such as the role of government and racial relations, than modern-day Republicans (Foner, 1988; Richardson, 2009). The realignment of Southern Democrats with the Republican party (Black & Black, 2009; Jacobson, 2000; Roberts & Smith, 2003; Rohde, 1991) and the rise of Republican conservatism (Pierson & Skocpol, 2007) have resulted in parties that are drastically different from their identically-named predecessors.

Together, the old and modern datasets span at least four different party systems. The old dataset incorporates the 3rd (1850s-1890s), 4th (1890s-1930s) and half of the 5th party system (1930s-1960s, the New Deal). The modern dataset includes the second half of the 5th party system, and possibly a 6th and even 7th system (Aldrich, 1999; Aldrich & Niemi, 2018; Karol & Hershey, 2014).

In practical terms, the challenge this presents is that the parties that are active in Congress in the modern hearings dataset and infleunce the hearings' agenda, are incredibly different compared to those in the old dataset. A model that learns from data generated by one set of party systems might fail at accurately predicting data generated by a separate set of party

systems. Consider what it would mean to include various party features in such a model, for instance which party controls each chamber (Republican/Democrat), which party heads the committee holding each hearing, the number of members from each party on the committee holding each hearing, etc. A Republican in the modern dataset is entirely different from a Republican in the old dataset (as are Democrats) and such features might worsen model performance, or at least, won't improve it.

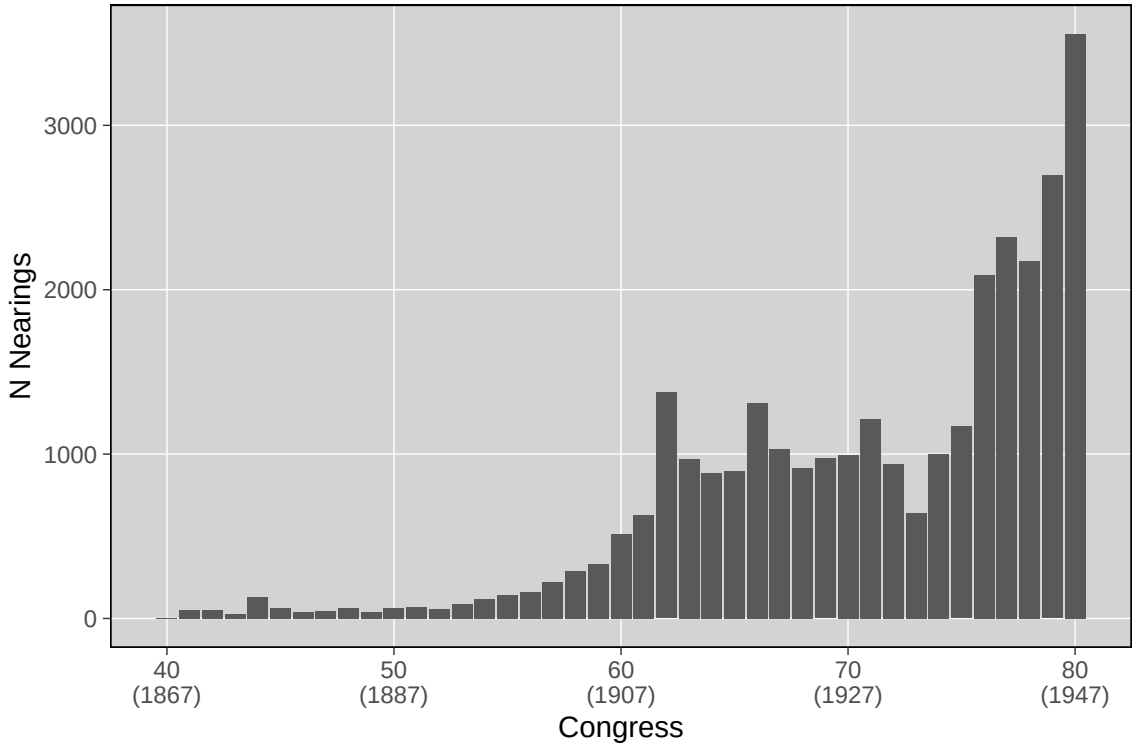
Figure 3.4 plots the number of hearings per Congress in the old dataset. The data are clearly skewed toward the second half of the dataset—54.7% of the hearings were held in the 72nd Congress (1931-1932) onward.[15] This pattern actually minimizes the party-system differences between the two dataset. The majority of the hearings in the old dataset are from the 5th party system, which began in the 1930s with the New Deal. This represented a major change in Democrats' approach to the role and size of government (Hawley, 2015; Leuchtenburg, 1963; Romasco, 1983), which was decisively different compared to the Democratic stance in previous systems (Argersinger, 1992; Foner, 1988; Richardson, 2009; Welch, 1988) and that has lasted till today.

As I illustrated in the previous sections, I am able to address this challenge in two ways at the agenda level. First, the terms that make up the textual features of the model represent the content itself that is produced in the two periods. As I demonstrated earlier, this difference is less severe than expected. Second, I account for changes in the agenda itself and the relative size of the agenda that the topic of each hearing covers.

To include party-level features such as those described above, would likely reduce the model's accuracy, overfitting it to patterns that are true for the training and test data, but not for the old dataset. The solution I propose is to account for certain things that the parties represent, without capturing system-specific party characteristics. My goal is to maximize prediction accuracy using as much information as I can gain from the parties in Congress

[15]This pattern may suggest that the data are incomplete. Data documentation and collection become more challenging as we move back in time. While it hardly matters for the purposes of classifying agency-creation, it may serve a problem for the theoretical inference we wish to make about agency-creation and the expansion of government. Simply put, we may be seeing a non-random sample of the data.

Figure 3.4: Number of Hearings Per Congress (Old Dataset)



without overfitting my model, which I achieve by including two sets of non-textual features in my final model that relate to party control of American government and party polarization.[16]

The first set of features includes indicators of party control, but instead of indicating which party controls each chamber, I indicate whether the two chambers are controlled by the same party. This way, I am not including information about *who* the parties are but I am assuming that when the same party controls both chambers, it is more likely to hold hearings concerning agency-creation. The logic behind my hypothesis is that federal agency-creation is a difficult task, one that is likely to fail if the two parties disagree on the role of the federal

[16]Readers might wonder if adding a feature representing the party-system may be useful, if for example, agency-creation is more likely in a given period (we could also include a congress or year feature). While it is possible, it is unlikely to be very useful. The main problem is that, in this case, the model will have been trained on one set of values (for these features), but it will encounter, for the first time, a different set of values in the unseen data. For example, a Congress feature will include values between 80 and 115 and might suggest that values between 90 and 100 (see Figure 3.1) are most likely to point to agency-creation. What is a model to assume when the unseen data include values between 41 and 80? Standardizing such features will not solve the problem either. Even if some models are able to technically overcome this (e.g. by converting missing/new values to the modal value in the training set), it effectively negates the potential information to be gained from these features, making them useless and possibly harmful for predicting unlabeled data.

government *and* if they each control a different chamber. It may be more worthwhile to invest precious time on agency-creation when there is a greater likelihood of achieving it. Following the same line of thought, I account for whether the same party also controls the presidency. Below, I list a series of binary features added to the final model:

- House, Senate & President controlled by the same party. Labeled *SamePartyAll*.
- House & Senate only controlled by the same party. *SamePartyHouseSenate*.
- House & President only controlled by the same party. *SamePartyHousePresident*.
- Senate & President only controlled by the same party. *SamePartySenatePresident*.

I describe the second set of relevant features in the next section, which capture elements of party polarization.

3.3.2.4 Party Polarization As I described in the previous section, accounting for the specific party controlling Congress or the committee holding a given hearing, might prove fruitless when the sets we are working with cover distinctly different party systems. I therefore find alternative means of accounting for the parties in Congress and their potential effect on agency-creation. One important and dynamic characteristic of the two parties is their degree of polarization over time.

Conventional wisdom suggests that American history has been rife with political polarization. Parties have traditionally been polarized on the topic of the day: "the Democrats and Republicans were polarized on slavery in the 1850s, agrarian and currency issues in the 1890s, the social welfare issues surrounding the New Deal in the 1930s, and civil rights in the 1960s (Stimson & Carmines, 1989; Sundquist, 1983)" (Layman et al., 2006, p. 85). In the mid 19th century it became so extreme, it culminated in a civil war.

The middle of the 20th century is quite unusual in this regard. Polarization existed, but it wasn't sorted along partisan lines (Hetherington, 2009). Thus, this period, in which the

parties appeared indistinguishable on most issues, is likely the exception to the rule and is in keeping with the Schattschneider (1960) tradition on party conflict, which posits that partisan polarization on the dominant issue of the day is to be expected in an effort to gain power.

Still, party polarization in recent years stands out compared to previous eras of polarization. Several studies have traced the development, causes and consequences of polarization in the US Congress (e.g. Black & Black, 2009; Jacobson, 2000; Lee, 2008; Roberts & Smith, 2003; Rohde, 1991; Schaffner, 2018; Theriault, 2008). Modern times appear unique because perspectives on polarization in American history "point to party polarization on a single dominant policy dimension, we argue that the current parties have grown increasingly divided on all the major policy dimensions in American politics—a process that we term conflict extension" (Layman et al., 2006, p. 84).

Changes in partisan polarization matter for agency-creation because—under the assumption that parties (and their leaders) matter for legislative outcomes (Aldrich & Rohde, 2001; Cox & McCubbins, 2005)—parties and their members in Congress are more likely to agree to create a new agency within the American government when differences between them are smaller. Higher polarization suggests greater differences and may be tied to gridlock (Binder, 1999; D. R. Jones, 2001; Thurber & Yoshinaka, 2015), which could result in denying agenda-space (Bachrach & Baratz, 1962; Sinclair, 1986) to agency-creation in congressional hearings.[17]

For model training, measures of how liberal or conservative the two parties are (and thereby, how polarized they are) may be useful. Such measures are able to capture how different the two parties are, without making any assumptions about who or what the parties are, or what agenda they promote. Instead, they pick up on an important way in which the parties have changed. They are therefore more appropriate for addressing the temporal aspects of the hearings datasets. They may be particularly useful in conjunction with features

[17]Although, some scholars offer competing evidence, suggesting Congress continues to pass meaningful legislation despite party polarization (e.g. Adler & Wilkerson, 2013; Mayhew, 1991).

relating to the agenda, party control and the given chamber.

DW-Nominate scores offer some insight to the changes in polarization over time. Using aggregated mean scores by party-chamber-Congress supplied by voteview.com (Poole, 2005; Poole & Rosenthal, 2017), Figure 3.5 illustrates how the parties' mean legislative behavior has changed within each chamber of Congress, and with respect to the two dimensions of political differences. On the first dimension, measuring a general liberal versus conservative voting tendency, the two parties have been polarized throughout most of the period examined, but the Cold War period does represent an unusual time for political polarization in which the distance between the two parties was much smaller than in other times. In fact, in the late 19th Century, polarization reached historically record-breaking heights (Mettler & Lieberman, 2020) and the period during which the debate over the creation of most agencies was at its highest, was when the parties were least polarized.

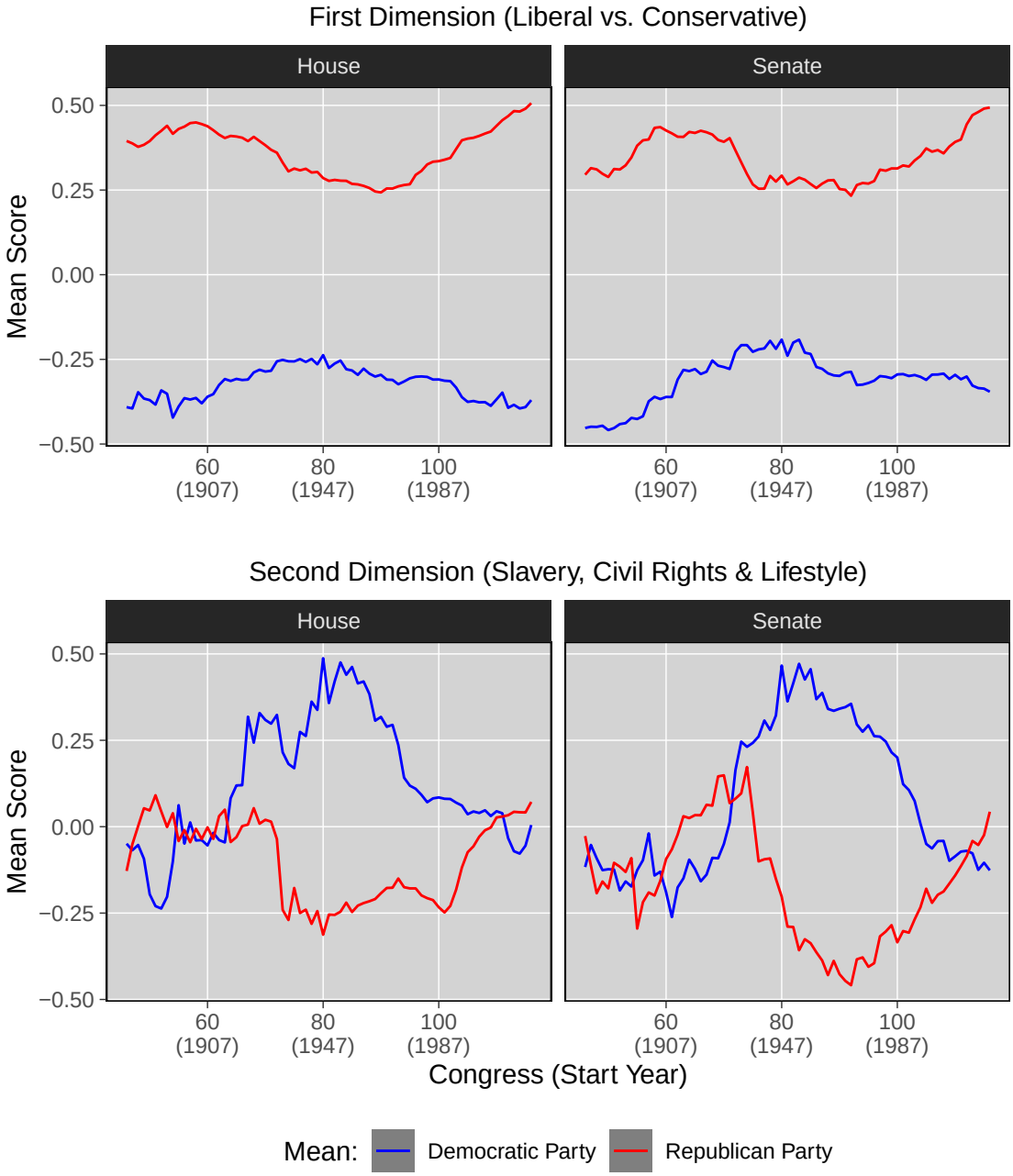
On the second dimension, measuring behavior with respect to slavery, currency, nativism, civil rights, and lifestyle issues, there appears to be a somewhat mirror-image. Up until about the 70th Congress (1927) most differences between parties were small, if they existed at all. The two parties then polarized and the gap between the two lasted for almost the entire period, only to be closed again in recent years.

Changes in ideological voting behavior could be meaningful to the likelihood of discussing agency-creation. Specifically, as the average DW nominate scores of the two parties appear closer to one another, the possibility of discussing agency-creation appears higher. I therefore include several relevant measures in my final model:

1. Average DW nominate score in each Congress (both chambers). *Labeled avg_dw.*
2. Average DW nominate score in each Congress, of the relevant chamber in which the hearing was held. *avg_dw_chamber.*
3. Average DW nominate score in each Congress, of the two parties. *avg_dw_dem (Democrats)* and *avg_dw_rep (Republicans).*
4. Average DW nominate score in each Congress, of the two parties in the relevant

chamber in which the hearing was held. *avg_dw_dem_chamber* (Democrats) and *avg_dw_rep_chamber* (Republicans).

Figure 3.5: DW Nominate Scores



3.3.3 Severely Imbalanced Training Data

The challenges thus far relate mostly to theoretical differences between the two periods and their empirical implications for the datasets I am working with. These challenges may be more or less influential for model training and for the most part, their severity becomes evident post-hoc. To some degree, the political environment challenges can also be accounted for in the model by including several of the features described in previous sections.

The challenge I present here relates to the data themselves. In the modern hearings dataset, only 1,419 (1.53%) of 92,597 hearings discuss agency-creation. Such a small percentage presents a problem of imbalanced data between our classes (agency-creation or not), illustrated in Figure 3.6.

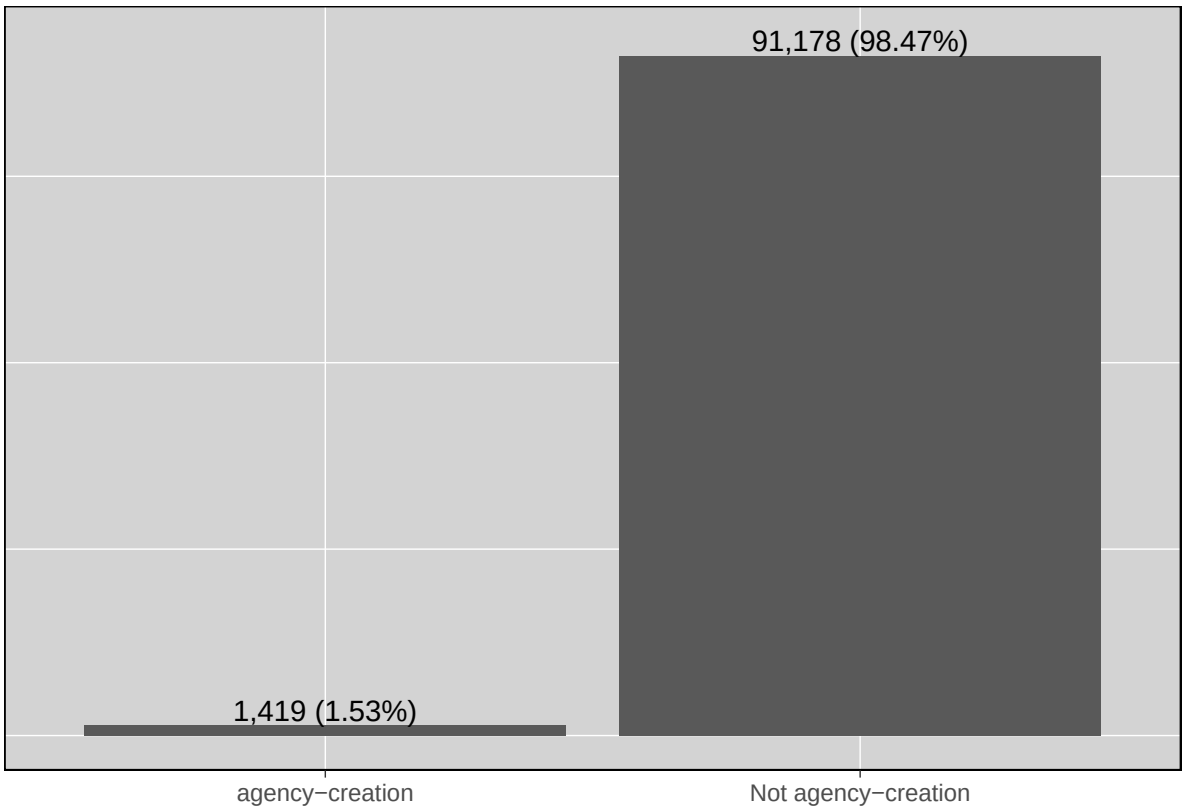
To understand why this imbalance is so severe, imagine you're at the roulette table in Las Vegas. You start to notice a pattern: 98 out of every 100 rolls turn up black and only 2 rolls turn up red. At this point, it's not so important to understand when the ball might land on red (and it might prove near impossible to try) because betting on black will win 98% of the time. With agency-creation, a model trained on a training set that represents the true imbalance of the population will simply always predict "not agency-creation" and it will yield 98% accuracy. Any model that can yield 98% accuracy will be nearly impossible to improve, even if its recall is in fact 0%. But here, almost all we care about is recall—the ability to identify those few hearings that do relate to agency-creation.

3.4 Model Training: The Modern Hearings Dataset

3.4.1 Data

I begin by splitting the modern hearings dataset into a test set and a training population. For my test set I randomly sampled about a third of the agency-creation hearings (N = 419) and complemented it with a random sample of the non-agency-creation hearings (N = 2,095), five times the size of the former. Normally, we would aim to use a representative

Figure 3.6: agency-creation in the Modern Hearings Dataset



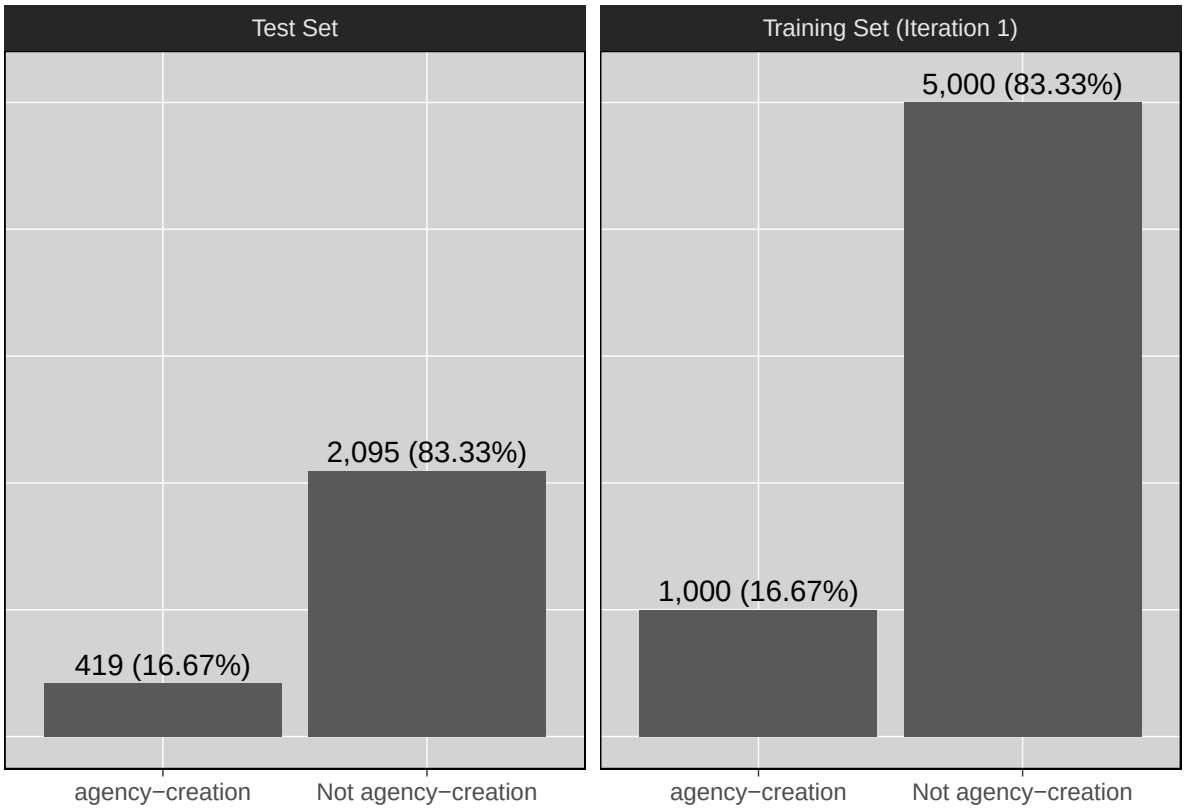
sample of the population as our test set. But, the imbalance between classes (agency-creation vs. non-agency-creation) is so severe, and the number of agency-creation hearings is so small, that only a handful of them would be included in such a sample. Instead, I opted for a stratified random sample that includes a sufficiently large share of the class of interest (agency-creation), maintains a substantial imbalance compared to the non-agency class (even if not as severe) and still leaves enough observations of the class of interest in the training population.

For the training set, I selected all remaining hearings that discuss agency-creation (N = 1,000) and add to them a random sample of non-agency hearings, five times its size. This ratio is not in itself meaningful in any way. I tried several ratios between the two classes, from 2:1 (non-agency:agency) to 9:1 and 5:1 appeared to provide the best results, which I present here. The key was to maintain the imbalance between the two classes, but make it less severe than in the true population so that the machine can identify patterns to predict

agency-creation, *and* to use as much information on the class of interest as possible.

Note, however, that I in fact am forced to ignore a large mass of information by sampling only several thousand non-agency hearings, of a potential 92,597 hearings. While this information is crucial in order to distinguish agency-creation from non-agency-creation, oversampling the non-agency class will diminish the model's recall. Therefore, a small sample that uses only some of the data is actually preferable in this case.

Figure 3.7: Sets for Iteration 1



3.4.2 Machine Learning Algorithms

I test the performance of two widely used algorithms: Random Forest and gradient boosting. The former builds several trees and uses a random subset of the data in each tree (aka bagging). Then, it averages the probability that each tree assigns each of the classes examined, to produce a single probability that a given observation belongs to a given class. Using random subsets of the data is a useful method of overcoming tthe greediness of trees, reducing overfit

and allowing the machine to learn weaker patterns. Gradient boosting also build a collection of trees, but instead of randomly generating each tree on a subset of the data, it fits each tree to predict the residuals of the previous tree (aka boosting). By doing so, it essentially attempts to correctly predict the errors made by the previous tree. The term gradient refers to the slope of the function, which is used to minimize the cost function of the model.

My expectation is that a gradient boosting model (GBM) will do better than a Random Forest (RF) model, especially with a sufficient number of trees. The number of trees, along with several other parameters (e.g. learning rate, how much weight to give the pattern learned from each tree, or depth, how many features to split before making a prediction) are often referred to as hyper-parameters. Hyper-parameters affect how the machine learns but unlike the patterns that help make predictions of y based on values of X, hyper-parameters cannot be learned by the machine itself and have to be provided externally.

Hyper-parameter tuning is a field in and of itself and practitioners use several approaches. For example, in a grid approach we may build a grid of parameters, with a range of values for each parameter. For each combination of values, we train a model and test its performance, proceeding with the combination that yields the best performance. Table 3.3 includes an illustration of a very small grid, tuning 4 values of trees (1000, 4000, 7000, 10000), two depth values (4, 5) and three learning rates (0.05, 0.075, 0.1).

Grids are usually much larger, including a much greater range for each parameter and can be very demanding. Random grids offer one solution, in which instead of trying all possible values and combinations, we randomly sample a number of combinations from the grid and test those. A Bayesian grid uses prior knowledge of the performance from each combination to limit the random sampling to areas in which performance seems to be best. Rather than making the entire process completely random, and "wasting" attempts on combinations that likely won't prove useful, it focuses on areas that seem to be most promising according to previous results and searches within them, to converge on an optimal combination of parameters.

Table 3.3: Hyper-Parameters Grid

trees	depth	learning_rate
1000	4	0.050
4000	4	0.050
7000	4	0.050
10000	4	0.050
1000	5	0.050
4000	5	0.050
7000	5	0.050
10000	5	0.050
1000	4	0.075
4000	4	0.075
7000	4	0.075
10000	4	0.075
1000	5	0.075
4000	5	0.075
7000	5	0.075
10000	5	0.075
1000	4	0.100
4000	4	0.100
7000	4	0.100
10000	4	0.100
1000	5	0.100
4000	5	0.100
7000	5	0.100
10000	5	0.100

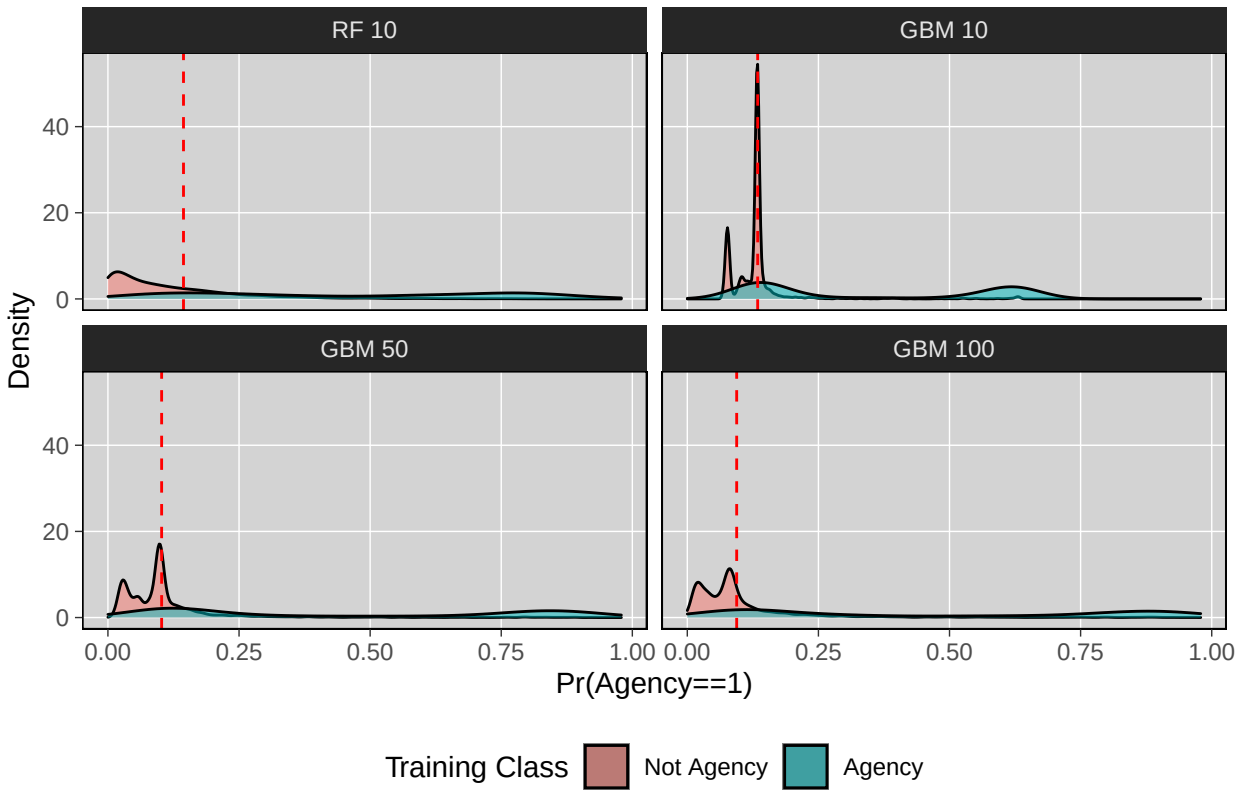
Here, my data are incredibly limited and hyper-parameter tuning using one of the methods above will likely be inefficient and perhaps a waste of time and resources. Instead, I manually tune the parameters—testing out several different values of the parameters myself and choosing those which appear to do best. Besides the comparison between RF and GBM, I illustrate the manual tuning of the number of trees in GBM. Given how small the training data are, I use an unusually small number of trees and demonstrate that increasing them to the typical range of thousands of trees does not, in this case, improve model performance.

In all training, I also use repeated cross-validation with 10 folds (subset), repeated 3 times. Cross-validation is a method in which a set is randomly split into v folds (here, $v = 10$). The model is then trained on $v - 1$ (9) folds separately and the best of those models is applied to the remaining (10th) fold (Geisser, 1975; Schaffer, 1993; Stone, 1974). In essence, it is as though we're creating an additional test set to test our model on, as it trains. In repeated cross-validation, we repeat the process k times. Cross-validation is a useful method for minimizing overfit, especially when our sets are two small to create additional, separate evaluation sets (see Chapter 4).

3.4.3 Results

3.4.3.1 Model Performance In Figure 3.8 I present density plots of the predicted probabilities of agency-creation **in the training set**. The higher the probability, the greater the proportion of trees that predicted agency-creation for a given observation, and therefore, the greater the likelihood of an observation to relate to agency-creation. The plot is split into four panels, one for each of the four models I examined prior to adding non-textual features. Density distributions are plotted by training class. That is, since these data are pre-labeled into their respective classes—agency-creation or not—can use the probabilities that the model attributes to them to learn about its ability to distinguish between classes. A reminder, features at this point include only term-based features that are included in the hearings' description.

Figure 3.8: Model Probabilities by Class



Vertical red dashed line marks 60th percentile per model.

Several conclusions are apparent. First, the probability distributions differ substantially by class in all four models, suggesting all models do a pretty good job distinguishing between the classes. Second, obviously, ten trees is not enough (note the density of the not-agency-creation class when the number of trees is low, especially for gradient boosting) and increasing at least to 50 improves the model's ability to distinguish between classes.[18] Although, increasing the number of trees in the GBM model is limited in its ability to improve model performance: Note the small change from 50 to 100 trees (and increasing further to 1,000 trees do not improve results). Third, the overlap between classes suggests that no matter which model we choose, we are likely to have errors. We will either have to sacrifice precision (accurately classifying observation i as agency-creation) for recall (identifying all agency-creation observations but including a portion of false-positives, i.e. incorrectly identifying some observations as agency-creation) or vice versa. In this case, recall was more important and false-negatives can be handled by complementing the machine's classifications with human review.

The figure also illustrates another problem we face with classification: Identifying an appropriate threshold. By default, many applications of ML algorithms will choose 0.5 as a cut-off, suggesting that anything above or equal to a probability of 0.5 should be classified as the class of interest and anything below it should be classified as the remaining class. A threshold of 0.5 makes intuitive sense but is often disconnected from the actual data. Here, it is clear that such a threshold would result in very high precision (100% in all the GBM models and close to it in the RF) but very poor recall—ignoring at least 50% of the agency-creation observations. Instead, we may choose a lower threshold that best balances the trade-off between precision and recall in a given case, for a given purpose.

The vertical dashed line in red illustrates such a possible threshold. Note, the threshold doesn't have to be a fixed probability. Instead, it may be a fixed percentile, thus expressing a cut-off that is more closely related to the output distribution of the model. In the figure, the

[18] I present here only the results of a Random Forest with 10 trees, for the sake of comparison. Increasing the number of trees up to 1,000 (which may be more appropriate with weak learners) did not improve the results from the Random Forest model and it is clear that in this case, the GBM is a better alternative.

Table 3.4: Accuracy Measures

Model	Precision	Recall	False.Positive.Rate
RF 10	0.706	0.818	0.659
GBM 10	0.686	0.759	0.684
GBM 50	0.708	0.824	0.657
GBM 100	0.709	0.826	0.656

[a] Threshold set to 60th percentile.

red line marks the 60th percentile, which may be a good threshold as it would provide a high level of recall with a relatively low false-positive rate.[19]

In Table 3.4 I compare accuracy measures, treating the 60th percentile as a threshold. The table illustrates that random forests does appear to do poorly compared to its gradient boosting counterpart, but only when the number of trees is increased. Since gradient boosting attempts to correct the errors of the previous tree, this makes sense. With fewer trees, averaging across trees that were generated randomly is an advantage. Although improvements are small, it is clear that increasing the number of trees to 50-100 maximizes recall and slightly reduces the false positive rate.

The values in the table are calculated using a confusion matrix, which compares observed and predicted values. While confusion matrices are very convenient for interpretability, they can only be calculated for one given threshold at a time. They are therefore quite limiting and may be prone to error.

Fortunately, ROC curves plot these very measures for every single threshold possible in the data, essentially aggregating the results of all possible confusion matrices. As such, they are particularly useful for comparing model performance. I plot the ROC curves for all

[19]Of course, choosing a correct threshold is an enterprise in and of itself. For identifying things like this, it is often a good idea to create an additional set (data-permitting), that is separate from both the training and test set. This set offers the opportunity to apply the trained model on pre-labeled data and learn from it to inform certain decisions, prior to applying them to the test set. In this case, there is too little data to create an additional set so I rely on the training set for this information. The risk here is the potential to overfit my model to my training set—if not in the actual training, then in my decision on a good cut-off threshold. I overcome this in two ways. First, by sampling and manually reviewing some of the machine's classifications. Second, by using repeated cross-validation in the model training itself. Although this doesn't apply to the choice of threshold, it does apply to the probabilities and their distribution that each model yields.

four models in Figure 3.9. The rule of thumb is that models that trend toward the upper left corner show the best performance because they maximize the true-positive-rate while minimizing the false-positive-rate. However, the trend can be noisy and often we are more interested in the area under the curve (AUC). The larger the area, the better the performance. And yet, even here we may decide to prioritize a model with lower AUC, if the ROC curve is higher in a particular area. For example, if we are willing to accept a certain rate of false-positives, we may focus on a particular range of the ROC curve and examine which line is highest in that range.

The figure illustrates the advantage that the GBM models have over the RF model. The AUC is similar for all four models but for most of the values, the RF curve is lower than all of the other three. The figure also illustrates that the differences between the three GBM models are very small, but that more trees increase the true-positive-rate when we wish to minimize the false-positive-rate (left side of the panel) but fewer trees improve our true-positive-rate when we are willing to accept more false-positives.

It is for this reason that I decide to ultimately proceed with 50 trees. Results so far suggest that increasing to 100 or more trees yield very small improvement in precision, possibly at the expense of recall.

3.4.3.2 Adding Non-Textual Features Next, I retrain the GBM 50 model with the addition of several non-textual features outlined earlier. Figure 3.10 compares the ROC curve for the GBM 50 model, with (marked by a + sign) and without non-textual features. The curve and the AUC clearly illustrate the advantage of adding non-textual features. At every point along the curve, the model that includes non-textual features out-performs the model that excludes them. I choose this model to proceed and use it to classify my test set.

3.4.3.3 Feature Importance The decision to proceed with a particular model is based not only on performance and accuracy measures, but also on feature importance. One of the challenges of using ensemble methods such as a collection of decision-trees is that we

Figure 3.9: ROC Curves for Basic Models

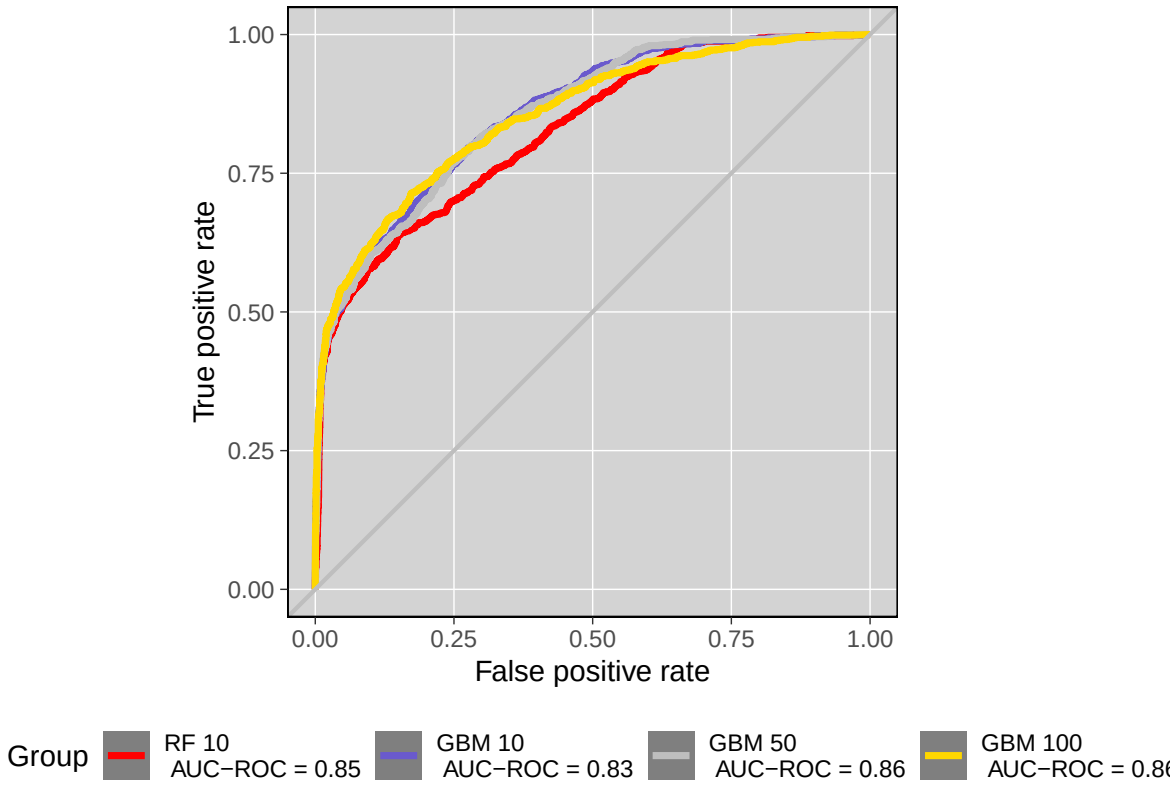
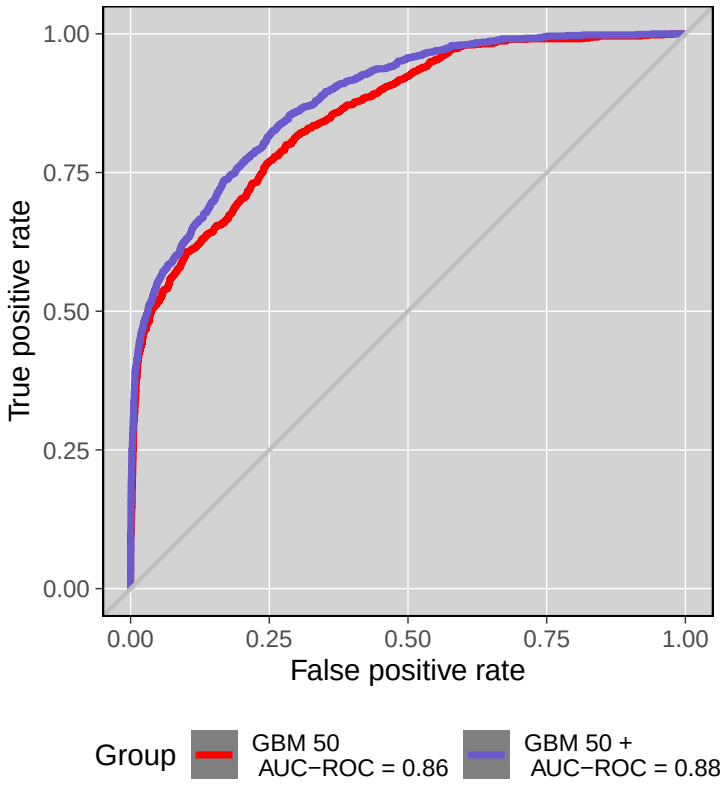


Figure 3.10: ROC Curves after Adding Non-Textual Features



sacrifice interperability for performance. That is, it becomes very difficult to understand which features the model chose to split on, when, on what values and how they interact with one another. For a single decision-tree, with a limited number of features, we may plot this quite easily. How do we plot a model that sums over several (sometimes thousands) of trees and uses hundreds of features?

With this in mind, understanding feature importance has become an important part of assessing model performance. Several methods are available for assessing feature importance, for example information gain or SHAP values at both the aggregate and the individual (observation) level (Antwarg et al., 2021; Giudici & Raffinetti, 2021; Heuillet et al., 2021; Lundberg et al., 2019; Lundberg & Lee, 2017; Marcilio & Eler, 2020; Marcílio & Eler, 2021; Raschka, 2015; Shapley, 1953; M. Smith & Alvarez, 2021). Reviewing feature importance is not simply meant to satisfy our curiosity. It is an important post-hoc method of assuring that the model "makes sense." When reviewing it, we should often look for red-flags such as features we expected to be important but are ranked low; features that appear unusually high, etc. More often than not, exploring such anomalies reveal problems in our data and/or point to observations/features that may contribute to overfitting. Correcting for such problems and retraining our model may increase its performance and more importantly, the ability to generalize it to unseen data.

In Table 3.5 I list the top 30 features of the two models using a relative importance (RI) metric (such that all features are scaled relative to the most important one, which is set to 100). As before, I compare the GBM model with 50 trees to it's counterpart that includes non-textual features.

The first thing to note is that, overall, the textual features in both models make sense. Terms such as 'establish,' 'creat(e)'/'creation,' 'reorgan(ize),' 'commiss(ion)' and 'center' are all likely to be associated with the creation of new government agencies. Some words may indeed be associated with agency-creation, but in a very specific context. For instance 'indian' and 'park' relate to the creation of agencies dealing with public lands and Native American

lands. Such terms may be problematic when generalizing to data from a period where these types of agencies were not as prominent. The final results suggest that, here, these may be more useful because some of the old hearings that deal with agency creation, actually do relate to lands in Alaska and Native American lands.

Table 3.5: Feature Importance

GBM 50 Feature	GBM 50 RI	GBM 50 (+ Non-Textual) Feature	GBM 50 (+ Non-Textual) RI
establish	100	establish	100
examin	11.26	**avg_dw**	16.55
creation	9.217	**avg_dw_dem**	11.87
creat	7.181	creation	9.998
reorgan	6.237	**avg_dw_rep**	8.495
commiss	6.088	commiss	7.181
review	5.791	creat	7.004
h.r	2.322	reorgan	6.825
mainten	2.269	**prop_subtopic**	5.271
coordin	2.187	**avg_dw_rep_chamber**	4.08
indian	2.121	improv	3.618
park	2.047	**SamePartyAll**	1.819
ethic	1.861	indian	1.689
program	1.86	mainten	1.515
hear	1.854	research	1.505
center	1.845	**subtopic_count_scaled**	1.462
preserv	1.763	advisori	1.158
improv	1.636	ethic	1.109
nation	1.625	park	0.9615
nomin	1.556	addit	0.9584
advisori	1.521	act	0.9165
uranium	1.423	nation	0.8764
control	1.371	resourc	0.8463
conserv	1.357	monitor	0.8372
act	1.331	coordin	0.8196
cleanup	1.256	**avg_dw_chamber**	0.8006
addit	1.221	preserv	0.7837
fund	1.147	tribal	0.7818
limit	1.103	center	0.7616
regul	1.098	control	0.7372

Perhaps the most interesting insight gleaned from this table is the importance of the

non-textual features (in bold). Note how several of the top-most features relate to the average DW nominate scores in Congress, the prominence of the hearing's policy subtopic and the combined control of the two chambers of Congress and the presidency. Not only did including these features improve model performance, now it becomes clear they are some of the most important features on which the model chooses to split first—before most of the textual features come into play. These importance rankings may suggest that many of the terms relating to agency-creation are more indicative when the topic receives more attention, the chambers and presidency are all controlled by the same party and as a function of voting patterns within Congress. The latter is difficult to interpret. For instance, clearly, the average DW nominate score (avg_dw) in the entire Congress is important. But it is unclear what about this feature is important—are high values associated with agency-creation? Low values? What of the interaction with other features? A model aimed at statistical inference, rather than classification, would increase interpretability of these features (though it might introduce questions of endogeneity, which I will address later).

Some weaknesses stand out as well. First, the stemming algorithm I used may not be strong enough, illustrated by the distinction between 'creation' and 'creat'—two features that can likely be combined. Second, the terms 'h.r.' and 'act' are a little concerning. That a hearing refers to a bill should not be so important in identifying agency-creation and it may result in false-positives. The word 'uranium' is also very likely context-specific and not relevant to a pre-WWII dataset. Note how these terms fall in ranking when non-textual features are added to the model. Thus, even though I did not correct for these weaknesses (and in general, I would recommend correcting for them), the improved model appears to correct some of the weaknesses on its own. In this respect, the feature importance of the improved model makes more sense.

It is also interesting that none of the top 30 features include terms such as 'agency,' 'department,' 'bureau,' 'organization,' etc. These may still be important, but simply ranked below 30. Such terms might be likely candidates to include in a dictionary if we were

compiling one a-priori, yet the model indicates they are not as important for identifying agency-creation on their own. Perhaps this is because the type of body in question may change from observation to observation, whereas the verbs relating to their establishment may be more consistent. It may also have to do with their correlation with other top-ranking features in the model.

3.5 Classification of the Test Set: Pre-Labeled Data

When using machine learning for this type of problem, we spend most of our time on training the model. Until this point, I only worked on our test set once: When I created it and separated the training population from it. Only now that I have completed training and found a satisfactory (even if not perfect) model, do I return to the test set.

As its name suggests, its sole purpose is to *test* how well our model performs on known data. It is an important test because it illustrates the extent that our model can be generalized from the data it was trained on, to other data. As such, it also rests on two particularly important assumptions. First, that no data leakage occurred between the test set and our trained model(s). In practice, this means that the two sets do not share any of the same observations and nothing about the data in the test set was used to inform the trained model.

Second, that the test set is a good representation of the population we're interested in, and specifically, of the unseen data we plan on classifying using our model. This assumption can be difficult to confirm and often to meet. As a reminder, our test set was drawn from the modern hearings dataset, whereas our new data are from a much older dataset of congressional hearings.

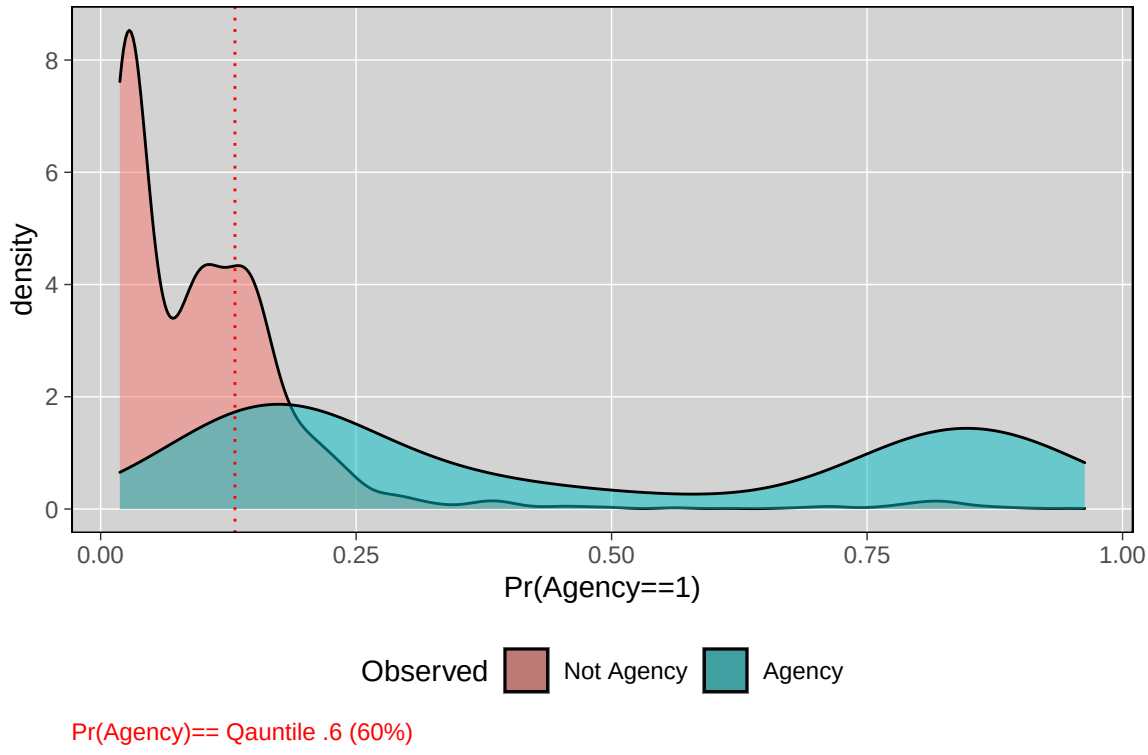
Applying our trained model to other data—any data, test set or otherwise—requires pre-processing the other data in the same way. In this case, all it requires is to ensure that all features used in the model appear in the test set as well.[20] If a single feature—in this

[20]In other cases, it may require different types of pre-processing that might be more complicated. Including, for example, imputation for missing values, handling of categorical features, scaling and/or centering features and principal component analysis to reduce the dimensionality of the data when features may be correlated

case term—is included in the model but not in the test set, it will fail to classify the test set. Because most of our features are in fact terms, we may identify which terms are missing from the test set and add them, filling all rows with 0 (because they do not appear in any observation).[21]

In Figure 3.11 I present density plots of the predicted probabilities in the test set. The probabilities are obtained by applying the trained model to the observations in the test set. The advantage of the test set is that, much like the training set, the true class of every observations is known, allowing me to assess the model's accuracy on the test data. I plot the probabilities by class.

Figure 3.11: Predicted Probabilities in the Test Set



or numerous. Whatever method is chosen, the same exact procedure must be applied to the training set, as well as any other sets the model is to be applied to (e.g. a test set or unseen data). In this case, the large number of features (terms) would make principal component analysis appealing but because so few words appear together, most are not likely correlated with one another and dimension reduction in such sets usually proves useless.

[21]This approach will vary based on the model being trained. Here, features represent terms so the decision to fill in the value 0 is appropriate because those terms truly do not appear at all in any of the observations. The choice on if and how to fill missing values may largely depend on what the feature itself measures and on the chosen algorithm.

It is striking how similar the distribution of probabilities in the test set is compared to those of the training set (see Figure 3.8). The Agency class exhibits the same two local maxima at the start and end of the distribution, though the dip in between the two is not as pronounced. The not-agency-creation class starts with two maxima and quickly dissipates. Almost all of the not-agency-creation received a probability lower than 0.25. The dotted red line illustrates the 60th percentile of the entire distribution, using the same potential threshold as before. If I treat the 60th percentile as the threshold, recall will be very high, correctly identifying 87.35% of the agency-creation observations in the test set. I will of course have a relatively large amount of false-positives, resulting in a lower precision value (0.72) but the majority of not-agency-creation observations will fall below the threshold. Since I plan to manually review samples of the unseen data post-classification, in the trade-off between recall and precision, I prefer recall.

3.6 Classification of Unseen Data: Unlabeled Data

Much like with the test set, unseen data require pre-processing and then simply classifying them with our model. An important difference is that in unseen data, labels are unknown—I am using the model to estimate the correct label. In some scenarios we might have a feedback mechanism that at some point confirms classification or marks them as an error. For example, in commerce, a transaction can either be approved or declined. When someone commits fraud and the transaction is approved, at some point in time we will have an indication that the decision to approve the transaction was wrong, because it was reported as fraud (e.g. a customer notifies the merchant that their credit card was stolen and they did not commit the relevant purchase). If, after a certain period an approved order is not reported as fraud, we can assume the decision to approve it was correct.

Feedback is an important mechanism of evaluating the performance of a model on unseen data, monitoring it over time and retraining if necessary. Feedback is especially important when using models to create measurements like I am here. Validating measurements has

been underused in political research (Ying et al., 2021) and can lead to misclassifications. It is therefore crucial to make sure that we are indeed measuring what we indented to, at the accuracy rate we expected.

Feedback can also be very important for retraining a model at a later stage because (a) we may wish to correct for the errors the previous model made; and (b) somewhat ironically, errors represent the observations for which we have the highest confidence in their labels (as opposed to an observation that received no feedback and we can only assume its classification was correct). Thus, they represent the best samples for a new model to learn from.

Figure 3.12 plots the density distribution of probabilities in the new data. Almost all observations received a very low probability of including agency-creation. On the one hand, this distribution is unsurprising—fewer agencies were in fact created in this period and so we might expect fewer hearings to discuss agency-creation. On the other hand, the distribution is concerning because as good as the model appears (at least on recall), it seems to be identifying very few observations as agency-creation. The concern is that these data are so different from the data the model was trained and tested on, that its generalizability is in fact very poor.

This concern is made worse when reviewing the actual probabilities. In Table 3.6 I group these probabilities together, illustrating how little variance they exhibit. Eighty-one percent of the observations received the probability 0.068. Less than 5% of the data received a probability lower than that, and only about 14% of the data received a higher probability. Using the 60th percentile as a cut-off would be meaningless here. Doing so, would result in almost all observations in the set to be classified as potential agency-creation hearings.

Instead, I divide the data into three groups, assuming that a probability of 0.137 or higher indicates a strong likelihood of agency-creation; 0.083-0.137 indicates a moderate likelihood; and below 0.083 indicates a low probability of agency-creation. I manually review all observations in the strong-likelihood category and a random sample of 1,000 observations from each of the remaining categories, totaling 3,405 observations (about 11% of the original

Figure 3.12: Predicted Probabilities in the Old Hearings Dataset

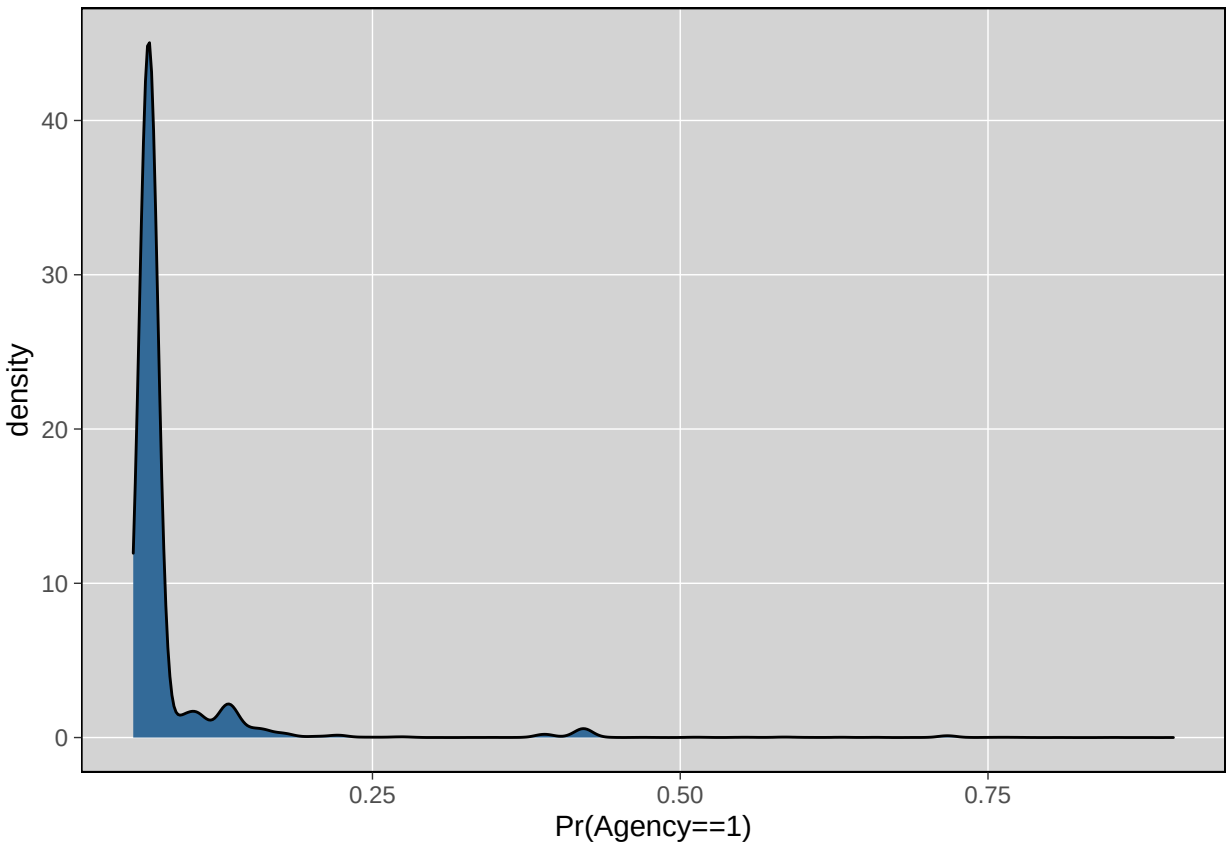


Table 3.6: Percentile Groups

Percentile Group	N	Percent
(0,0.0651]	1,403	4.59%
(0.0684,0.0719]	24,769	81.09%
(0.0719,0.0834]	152	0.50%
(0.0834,0.137]	2,817	9.22%
(0.137,1]	1,405	4.60%

Table 3.7: Recall

Likelihood	N	Sample	True_Agency
Strong	1,405	1,405	223
Moderate	2,817	1,000	17
Low	26,324	1,000	11

dataset).

In Table 3.7 I list the three groups, together with their sample sizes for manual review and the outcome. Precision is obviously very low, because of a large number of false-positives. This was the price I was willing to pay to maximize recall. Results suggest the model did an excellent job in terms of recall, identifying 223 agency-creation (of 251) in the strong-likelihood category. An additional 17 and 11 hearings were identified in the moderate and low categories, respectively, out of a sample of 1,000 observations. Most of these agencies related to governmental functions in territories such as Alaska and Hawaii (at the time, they were the responsibility of the federal Government in Washington). A few observations related to military agencies. The latter were not comprehensively coded for in the modern dataset, and therefore the model was unable to properly learn about such observations.

Subsequently, I searched both datasets for military-related terms to identify any additional agency-creation hearings. In the old dataset I also searched for various American territories. I identified an additional 15 hearings using this method, reaching a total of 266 agency-creation hearings. Identifying the weakness of the model, rooted in a type of agency the model was not trained on, allowed me to complement the model's performance with a simple dictionary, ultimately increasing recall and my confidence in the outcome measurement.

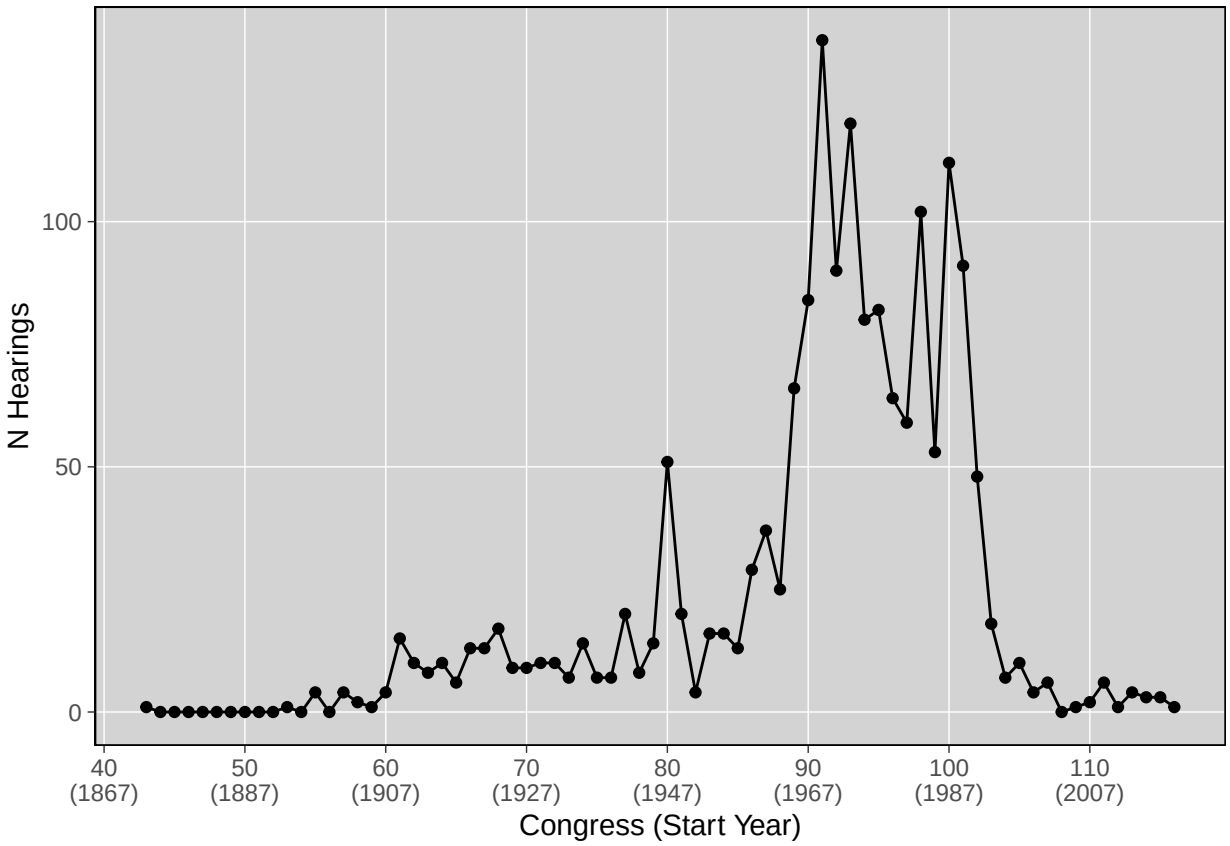
Models are never perfect and errors are to be assumed. These results actually boost my confidence in the model's performance, despite the various challenges it faced. I found it very difficult to address both recall and precision and chose to prioritize the former over the latter. Prioritizing recall allowed me to identify the most relevant observations using the model's predictions, manually removing any false-positives and searching for any false-negatives, i.e. agency-creation hearings misclassified as non-agency. Of course, there may be some additional false-negatives but based on the results from sampling the two lower likelihood groups in Table 3.7 and the follow-up searches, I am confident very few relevant observations remain unidentified. Thus, the trained-model allowed a manual review of only 11% of the data, instead of sifting through the entire dataset, and yielded very reliable results.

Finally, In Figure 3.13 I plot the number of hearings concerning agency-creation, within each Congress, over the entire period, combining the two datasets. Clearly, the great broadening that occurred in the second half of the 20th century was unique in terms of agency-creation at the federal level. Aside from a spike in the 80th Congress, immediately after WWII, no other period is as dramatic as the period starting in the late 1960s. That said, the first half of the 20th century does indicate a change with respect to previous years. Previously, most congresses did not hold a single hearing that included agency-creation, and only a handful held at most 4 hearings. Between the 60th Congress and the 79th Congress (1907-1946), every single congress held a minimum of 4 relevant hearings, with a median of 10 and an average of 10.55 hearings per Congress. It is possible that this represents a slow start of the great broadening that exploded later in the 20th Century, and has since died down, almost returning to the non-existent level of the 19th century.

3.7 Summary

My very first attempt at this project was a failure. I did not, at first, consider how severely imbalanced the classes were in the modern hearings dataset and the model I trained predicted that all observations do not relate to agency creation. Its predictions were correct for 98.5%

Figure 3.13: agency-creation in Congressional Hearings



of the observations in the modern hearings dataset and it would've been correct for 99.1% of the observations in the old hearings dataset—a precision that no model can compete with. But, its recall was a perfect 0. It failed to positively identify a single observation in the class of interest: Agency-creation.

Addressing this imbalance in both my training set and my test set was the first successful choice toward a useful model. Balancing the training set—controlling the ratio between classes and choosing the best examples for the machine to learn from—is a widely used technique. Balancing the test set is far more unusual. Had I not done so, relying instead on a random sample of data, my test set would have had only a handful of observations relating to agency-creation. It would've been impossible to assess model performance based on such a model.

A second choice I made was about which metric to prioritize. Given how severe the imbalance between classes is, it quickly became clear to me that I must prioritize recall. The most important aspect of evaluating my model was its ability to correctly identify all of the agency-creation observations. Of course, the best model would be one that is able to maximize recall first, while still minimizing false-positives (i.e. maximizing precision second). I was willing to manually review a large portion of the data but still required keeping this to a minimum to avoid reviewing the entire dataset (and thus, negating the entire point of using a machine learning algorithm here).

Finally, I chose to include non-textual features in the model. At first glance, this appears an easy decision. It increased the model's predictive power, substantially and wilted out many of the false-positives. Deciding on which features to include and how to measure them required addressing the theoretical differences between the two datasets, that stem from two politically distinct periods.

I proceeded with including non-textual features precisely because of how much they improved the model's predictions. Yet, this choice comes with great risk for anyone making use of this measurement. Consider a researcher who is interested in knowing if and how

Table 3.8: Negative Binomial Coefficients

dataset	term	estimate	std.error	statistic	p.value
modern hearings	avg_dw_dem	6.383	3.176	2.010	0.044
old hearings	avg_dw_dem	14.337	1.500	9.561	0.000

the distribution of DW nominate scores, as a proxy for ideology, affect Congress' tendency to discuss agency-creation. In Table 3.8 I list the results of a negative binomial regression. The target variable is a count of the number of hearings in each Congress that relate to agency-creation. The single predictor is the average DW score of Democrats in each Congress (this feature was ranked 3rd in feature importance for the trained model, see Table 3.5). Of course, such a model should include other predictors and the outcome variable might be measured as a proportion to account for the changing number of hearings held in each Congress overall, but I use this model for the sake of simplicity. I also remove the constant from the table.

Distinguishing between the two datasets, the models suggest that increases in the DW nominate scores of Democrats are associated with a greater likelihood to discuss agency creation. The effect is significant in both sets, though much stronger in the old hearings dataset. What are we to make of the effect we find in the old hearings dataset? Given that this feature was used in training the model that classifies agency-creation, and that the feature was one of the top-most ranking features, does this relationship reflect the true relationship between DW scores and agency-creation? Or, does it reflect the predictions that we made in the old hearings based on the relationship in the modern hearings dataset?

The problem is one of endogeneity that illustrates how the decisions I made during the process of training the model can have downstream effects on any research carried out with this measure. To some degree, I have mitigated the risk as much as possible by manually reviewing a large enough sample of the data and validating the machine's true-positive predictions (as well as identifying the false-positives). Additionally, assuming the effect above holds in a more suitable model, the researcher might seek a method to explain why this effect

is so much stronger in the old hearings dataset—both theoretically and considering the data themselves.

I presented in this chapter a well-detailed process of training a model for a relatively simple purpose. The process is infused with a theoretical understanding of the topic and a careful consideration of the data. My goal was to illustrate to the user the myriad of choices we make when training such a model. Big or small, we encounter such choices in every step of the way and they have consequences that affect both the outcome itself and how it might be used later on in research. Even if we don't justify every single choice, understanding that each choice can have meaningful consequences for model performance and its ultimate use in research provides a healthy perspective for using machine learning in practice.

4 Policy Topics in Congressional Bills

4.1 The Problem

The congressional bills dataset includes all bills introduced in Congress since the 80th Congress (1947). Currently, it ends with the 114th Congress, in 2016 and is therefore quite outdated. The data consist of several indicators about each bill submitted in Congress including information about the member who submitted the bill, the chamber it was submitted in, the committees it was referred to, the legislative status of the bill, etc. My goal in this study was to train a model for qualitative coding (N.-C. Chen et al., 2018). The model should accurately classify bills, based on the title of introduction, into the PAP major & minor policy topics. Assuming some random error, some level of error that a supervised model cannot address (because of a few PAP coding rules that are not necessarily expressed in the bill title), and changes from one Congress to the next (expressed in unseen data to be classified), I plan to maximize the model's performance such that it may reliably classify as many observations on its own. By doing so, only the smallest amount of bills possible will require human review. The threshold for reliable classifications of the model is the maximum recall possible that will maintain a minimal precision of 0.95.

The project is incredibly complicated for several reasons. First, the outcome is a multiclass variable. Excluding the rare occasion of no policy topic at all and the topic of arts (23), the PAP major topics consist of 20 different topics. Multiclass problems with 4 or 5 categories are substantially more complex than a dichotomous outcome, let alone 20.

Second, the outcome in fact includes several multiclass variables. After successfully classifying the major topics, I am left with classifying minor topics. Thus, each major topic can be broken down into a multiclass problem of its own. I am therefore faced with a total of 20 multiclass problems, each requiring their own model (one for classifying 20 major topics and then 19 different models for classifying the minor topics within each major topic; this excludes Immigration, which consists of only one minor topic). The number of minor topics

per major topic is listed in Table 4.1. The average number of topics in each major topic is 11.

Table 4.1: Number of Minor Topics Per Major Topic (PAP Codebook)

Major Topic	N Subtopics
1. Macroeconomics	9
2. Civil Rights, Minority Issues, and Civil Liberties	10
3. Health	18
4. Agriculture	9
5. Labor and Employment	10
6. Education	10
7. Environment	12
8. Energy	9
9. Immigration	1
10. Transportation	10
12. Law, Crime, and Family Issues	13
13. Social Welfare	7
14. Community Development and Housing Issues	11
15. Banking, Finance, and Domestic Commerce	14
16. Defense	19
17. Space, Science and Communications	10
18. Foreign Trade	8
19. International Affairs and Foreign Aid	13
20. Government Operations	18
21. Public Lands and Water Management	7

Third, it is likely that even 20 models may not be sufficient. Changes in the terms used, and their association with policy topics, from one Congress to the next are sufficiently frequent to make it difficult to classify data from one Congress based on the data from the previous meeting of Congress. Therefore, I can expect a good model, trained on the last few meetings of Congress in the data, which end with the 114th Congress, to classify a majority of the 115th Congress, but not all of it. The performance of such a model would likely be reduced further if applied to the subsequent 116th Congress and even more so, to the first year of the 117th Congress. Thus, with the addition of every additional Congress classified, I may need to retrain the series of models to rely on the most recent data and improve performance with respect to the subsequent model. Potentially, this may require an iterative process of

training 60 models.

Fourth, I am resigned to relying almost exclusively on features based in bill titles. To phrase it more accurately, my plan is to train a model without having to rely on features that may likely be used in research. For instance, assessing model performance on a test set, I can expect model performance to improve if I include features about who introduced the bill (e.g. seniority, state, party, the committee they serve on), the chamber the bill was introduced in, the committee the bill was referred to, etc. This approach raises two potential issues. First, it assumes the relationship between these features, the terms they use and their relationship to policy topics is fixed. Thus, if for example in the new data, Republicans begin to use terms previously associated with Democrats, the model will misclassify observations because of this. Second, these data are routinely used in political science for examining differences in the agenda across the very same variables—party, chamber, committee and so on. To use them as predictors that distinguish topics from one another and then have them used in research to also examine differences across these variables creates a major problem of endogeneity for an entire sub-discipline in the field (recall the example in the summary of the previous chapter). My model is therefore limited to the information that can be gleaned almost exclusively from bill titles.

Finally, in the very first attempts at training the very first model—for classifying major topics—it became clear that two dimensions of the problem need to be adequately addressed. A significant portion of the data used for training and assessing model performance appeared to have been misclassified. A large portion of the labeled data I was relying on was simply coded wrong. Any model trained on these data was unable to surpass the 80% level of precision in the *training data*, let alone the test data (no higher than 75%). When too many of the observations are coded wrong, the model struggles to find meaningful patterns, replicates errors and it becomes very difficult to assess its performance because the test data itself may be incorrect.

The second dimension relates to complexity. With a problem this complex, a Random

Forest algorithm using a simple document-term-matrix—the applied method in past attempts (Collingwood & Wilkerson, 2012; Hillard et al., 2008; Purpura & Hillard, 2006)—was simply not strong enough to address the complexity of the problem.

4.2 Model Training Strategy

As I have already alluded, the model training strategy relies on a supervised machine learning framework in which words are translated into features and we rely on pre-labeled data to identify patterns for predicting those labels. In the previous chapter, I made use of a document-term-matrix in which each row is an observation (in this case, a bill) and each column is a term. In each cell I record the number of times each term appeared in each observation and compared the performance of two supervised training algorithms—Random Forest and GBM. For the purposes of the previous study that approach was sufficient. Here, it proved inadequate.

The training data used here were much larger. After removing stop words, stemming terms and excluding terms that appeared less than 4 times in the entire training data, I am still left with 8,199 terms. To use these as features would create a very large, memory-demanding model. Moreover, it places a great deal of emphasis on the role of a single word, making it less likely to perform well on unseen data and likely resulting in overfitting. To improve performance on the bills dataset (and reduce dimensionality), I use unsupervised methods to create clusters or groups of terms as features based on how often they appear together (or with other terms). Each feature thus represents a count of a group of terms rather than a single term, making the feature itself more powerful, reducing the weight of a single term and reducing the number of features used in the model (an important consideration for both highly correlated features and for how memory-demanding running the model might prove).

4.2.1 Feature Engineering through Unsupervised Learning

To reduce the dimensionality of the data I pre-process it using two unsupervised learning algorithms that allow me to combine terms into meaningful groups. The first algorithm is the Word2vec algorithm, which, based on a corpus of data, creates numeric vectors of d dimensions as word representations (Mikolov, Sutskever, et al., 2013; Mikolov, Yih, et al., 2013; Mikolov, Chen, et al., 2013). These vectors measure the distance between terms based on shared and/or similar appearances in the data:

> "Extensive prior work in natural language processing has focused on automatically identifying semantically similar words. In general, this research relies on the distributional hypothesis, and the idea that words used in similar contexts have similar meanings. Building from this central insight, researchers have recently sought to identify methods for understanding a word's embedding in a vector space; that is, these approaches seek to capture meaning that is lost in sparse, discrete representations of terms. Consider, for instance, the terms 'king' and 'queen.' Standard approaches take the terms as discrete (i.e., 0 or 1). Instead, vector space models represent terms as distributions over word dimensions. Though none of the dimensions of the estimated vector are named, the 'loading' of each term on the dimensions often captures substantively important relationships. For instance, 'king' and 'queen' might have a similar concentration on a dimension that seems to relate to the concept of royalty but deviate on a dimension that seems to relate to man. The resulting word vectors provide a wealth of linguistic information" (Rice & Zorn, 2021, p. 3).

I process the terms in my data to yield vectors of 300 dimensions per term (a standard number of dimensions often employed with this algorithm). Prior to the use of this algorithm, I remove stopwords, stem all terms and use only terms that appear 4 times or more in the data.

Next, based on these numeric representations of the terms in my corpus, I use K-means clustering to group together terms. As detailed in Chapter 1, K-means clustering is an unsupervised algorithm that groups together columns of data in a way that minimizes within-group variance and maximizes between-group variance (Hartigan, 1975; Hartigan & Wong, 1979). The result is a series of K groups. The members of each group most resemble each other according to the measurement used and are most unlike the members of all other groups.

I use the the gap statistic to determine the number of clusters (at the major topic level, K=344) in my corpus of data (Tibshirani et al., 2001).[22] Combined, these methods allowed me to identify groups of terms that are related in some way, often appearing together and/or in similar contexts. I use each group as a feature in my model and therefore reduce the number of features (from over 8,000 to ~350) and reduce the risk of overfitting to patterns that emerge from single words.

Table 4.2 includes an example of two such clusters. The first combines several terms relating to vehicles and fuel consumption; the second relates to immigration and related procedures. Note, the latter cluster also includes a term that could lead to overfit, specifically listing the term 'haitan.' All terms are stemmed.

After a first few training iterations I refined these clusters by completely excluding terms that appeared to be causing misclassifications[23] and separated a handful of terms, which increased model performance when used as stand-alone features (and in various combinations with other features given the nature of the supervised algorithm).[24]

[22] I try several variations, using as few as 150 clusters and as many as 570 clusters; results are largely the same and do not improve on the 344 clusters determined using the gap statistic.

[23] Terms excluded completely: 'amend,' 'act,' 'bill,' 'oper,' 'implement,' 'program,' 'titl,' 'administration,' 'american,' 'institut,' 'department,' 'secretari,' 'offic.'

[24] Terms used as stand-alone features: 'safeti,' 'transit,' 'secur,' 'effici,' 'job,' 'vehicl,' 'reimburs,' 'construct,' 'research,' 'school,' 'compet,' 'youth,' 'young,' 'clean,' 'production,' 'power.'

Table 4.2: Example Clusters

cluster	terms
5	fuel; vehicl; electr; motor; emiss; automobil; coal; transmiss; phone; haul; speedway; vehicular; injector
276	immigr; alien; statu; citizen; legal; visa; waiver; nonimmigr; admiss; citizenship; admit; unaccompani; haitian; undocu; reunifi

One caveat to the clustering of these terms is that it lowers the interpretability of the model. Machine learning models are notorious for their black-box-like character, making it difficult to understand why a model makes the decisions it makes. Several methods exists for opening this black box, e.g. using SHAP values (Antwarg et al., 2021; Giudici & Raffinetti, 2021; Heuillet et al., 2021; Lundberg et al., 2019; Lundberg & Lee, 2017; Marcilio & Eler, 2020; Marcílio & Eler, 2021; Shapley, 1953; M. Smith & Alvarez, 2021) and they offer wonderful insights on the contribution of each feature to a model's decision on a specific observation or at the aggregate on all observations. With word-clusters, these features are simply named cluster 0 through n and as the researcher, it requires an additional step to reveal which terms are associated with each cluster and make interpretation of model decisions meaningful.

To be clear, these clusters of terms do not perfectly match up to the topics in the PAP codebook and cannot be used as a simple dictionary. It is the combination of these clusters used as features within the framework of an ensemble of decision trees that proves most effective.

To this list of features I add a single additional categorical feature. Every bill in the modern era has been classified by the Congressional Research Service (CRS) into subject

areas of their own. These subject areas do not correspond to the PAP policy topics in any straightforward manner. But they do represent an additional piece of information about the bills that can be used without risking endogeneity down the line. The list of subject areas includes 34 subjects.

Table 4.3 lists all 34 subject areas and their frequencies in the data used for training (80% of the bills in the 108th-114th Congresses; see next section). In Figure 4.1 I plot the relationship between the PAP major topic of bills and the corresponding CRS subject areas. The plot demonstrates two important trends. First, that the two coding schemes *do not* line up perfectly; every PAP topic is split between several CRS topics and vice versa. Second, despite that, some issues correspond quite well, for example the majority of PAP Health bills are coded as Health as well according to CRS. Similarly, most of the PAP Defense bills correspond to CRS Armed Forces and National Security. Together, these trends illustrate that they may include additional information on the likely PAP topic of each bill, if used correctly with useful patterns found in the bills' titles.[25]

4.2.2 Supervised Learning Algorithm: Catboost

Due to the categorical nature of subject-area feature and several different hyper-parameters that are easy to tune, I train the model using a Catboost algorithm (Dorogush et al., 2018). This algorithm is an ensemble of decision trees that employs boosting over gradient descent. In other words, each decision tree learns from the errors of the previous tree, rather than starting at a random split. It is unique compared to other boosting algorithms (e.g. GBM), in its method of handling categorical features, which does not require one-hot-encoding (converting it to m-1 binary columns). This algorithm is also unique in its ability to use symmetrical splitting. That is, within the decision tree, a feature can appear on both sides of a split made on a previous feature, increasing the ability to learn from feature values.

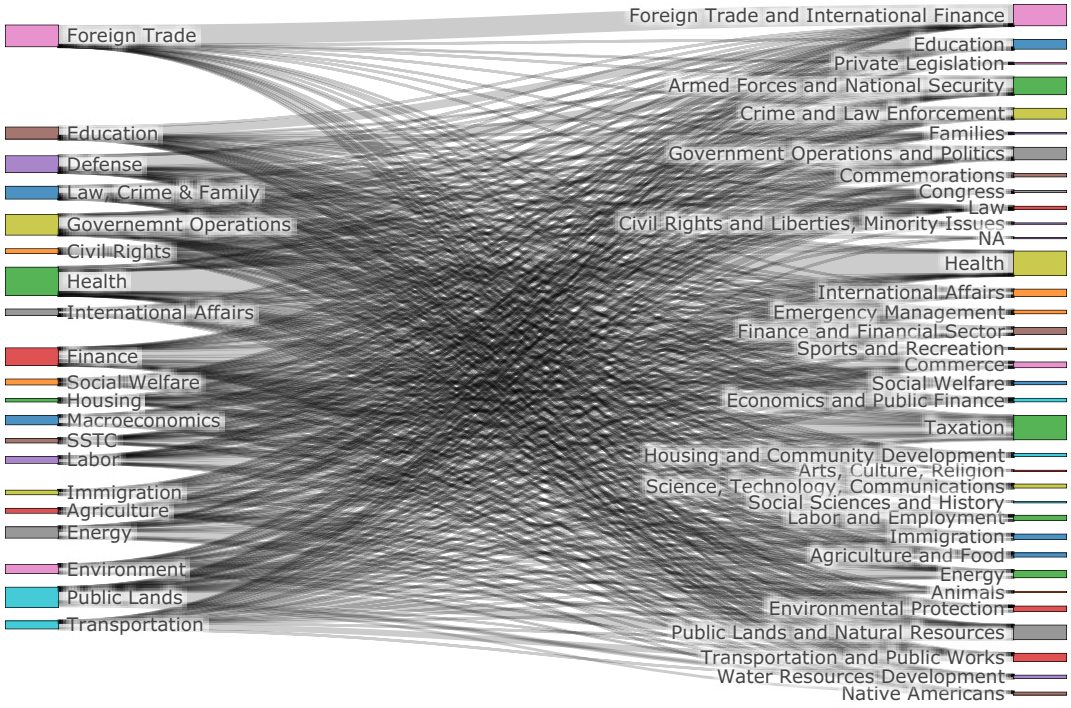
[25]See the online appendix for an enlarged version of this plot, along with separate plots for each PAP major topic and its relationship to CRS subject areas. Plots are presented as html widgets offering additional information (number of bills in each relationship) and additional functionality.

Table 4.3: Distribution of CRS Subject Area in Training Data

Subject Area	n
Agriculture and Food	1021
Animals	288
Armed Forces and National Security	4270
Arts, Culture, Religion	69
Civil Rights and Liberties, Minority Issues	254
Commemorations	861
Commerce	1391
Congress	526
Crime and Law Enforcement	2621
Economics and Public Finance	863
Education	2379
Emergency Management	879
Energy	1716
Environmental Protection	1313
Families	258
Finance and Financial Sector	1759
Foreign Trade and International Finance	5164
Government Operations and Politics	3030
Health	5916
Housing and Community Development	803
Immigration	1272
International Affairs	1807
Labor and Employment	1270
Law	742
Native Americans	731
Private Legislation	181
Public Lands and Natural Resources	3545
Science, Technology, Communications	965
Social Sciences and History	30
Social Welfare	839
Sports and Recreation	61
Taxation	5884
Transportation and Public Works	2010
Water Resources Development	811
	20

I combine manually tuning hyper-parameters, as well as an automatic tuning using a random search function on a pre-defined grid. Hyper-parameters are not parameters the

Figure 4.1: PAP Major Topics to CRS Subject Area



model can learn on its own, but rather parameters that affect how the model learns. Using an optimized combination of parameters can substantially increase model performance. The final model for classifying major topics used symmetrical splits, 9,234 iterations (trees), max depth of 10 (the number of features to split on before a decision is made—up to an interaction of 10 features), learning rate of 0.025 (how weak of a learner each tree is), l2_leaf_reg of 0.75 (regularizing the loss function to improve learning), rsm 0.2 (size of a random sample of features to use in each iteration, can address the greedy nature of the trees) and a loss function of MultiClassOneVsAll.

My description so far summarizes the selection and creation of features, mostly through the use of unsupervised learning algorithms, and the training of a supervised learning model using the Catboost algorithm. These steps encompass my strategy for addressing the complexity of the problem. The final question to address is the data itself.

4.2.3 Data

The data used in training the model for classifying major topics included the last congresses in the data: From the 108th Congress to the 114th. I am essentially ignoring a few hundred thousand observations from earlier congresses in favor of using only the most recent data.

The first step required correcting misclassifications in the data. Reviewing small samples of data suggested one type of error in the data that is both easy to identify and fix: Bills that are almost identical in title but have been coded into separate major and/or minor topics. I wrote a simple iterative search algorithm. At each iteration, it randomly sampled one bill. Then, it identified all bills that shared at least 80% of their terms in common with the sampled bill. If they were all coded into the same topic, I removed them from the pool of bills to sample. If they were coded into separate topics I moved them to a pool that required manual review. I continued with this search until no bills were left in the original pool. I performed the search only on the second half of the dataset, beginning in the 93rd Congress. The point was not to correct any and all errors in the entire dataset, but rather to correct a specific type of error in the data most likely to be used in training.

This method yielded 6,802 groups and a total of 50,210 bills to be reviewed. Table 4.4 illustrates one such group. The bill, which should be coded as major topic "Health" (3) and minor topic "Children and Prenatal Care" (332), was miscoded once at the major topic level into "Law, Crime and Family Issues" (12) and twice at the minor topic level into "Public Health and Disease Prevention" (331).

Reviewing these bills resulted in the correction of 10,725 bills at the major topic level and an additional 3,081 bills at the minor topic level.[26]

[26]See the online appendix for a series of plots that illustrate the corrections. For each original major topic, I plot the distribution of corrections into other major topics. A special thank you to Jacob Fridakis for taking a major role in reviewing these bills.

Table 4.4: Example of Misclassified Bills

BillID	Title	Major	Minor
109-HR-1709	To expand access to preventive health care services that help reduce unintended pregnancy, reduce the number of abortions, and improve access to women's health care.	3	332
109-S-20	A bill to expand access to preventive health care services that help reduce unintended pregnancy, reduce the number of abortions, and improve access to women's health care.	3	332
109-S-844	A bill to expand access to preventive health care services that help reduce unintended pregnancy, reduce the number of abortions, and improve access to women's health care.	3	332
110-HR-819	To expand access to preventive health care services that help reduce unintended pregnancy, reduce abortions, and improve access to women's health care.	12	1208
110-S-21	A bill to expand access to preventive health care services that help reduce unintended pregnancy, reduce abortions, and improve access to women's health care.	3	331
111-HR-463	To expand access to preventive health care services that help reduce unintended pregnancy, reduce abortions, and improve access to women's health care.	3	331

Once I completed this series of corrections, I moved to selecting the data to include in training. As in the previous chapter, the data here are imbalanced as well, although they are spread out between 20 categories, rather than just two. After several attempts at addressing this imbalance (random sampling, oversampling of smaller categories and undersampling of larger categories), it appeared the best performance was acheived using the entire data as-is.

As is customary in many machine learning practices (Raschka, 2015), I use a random

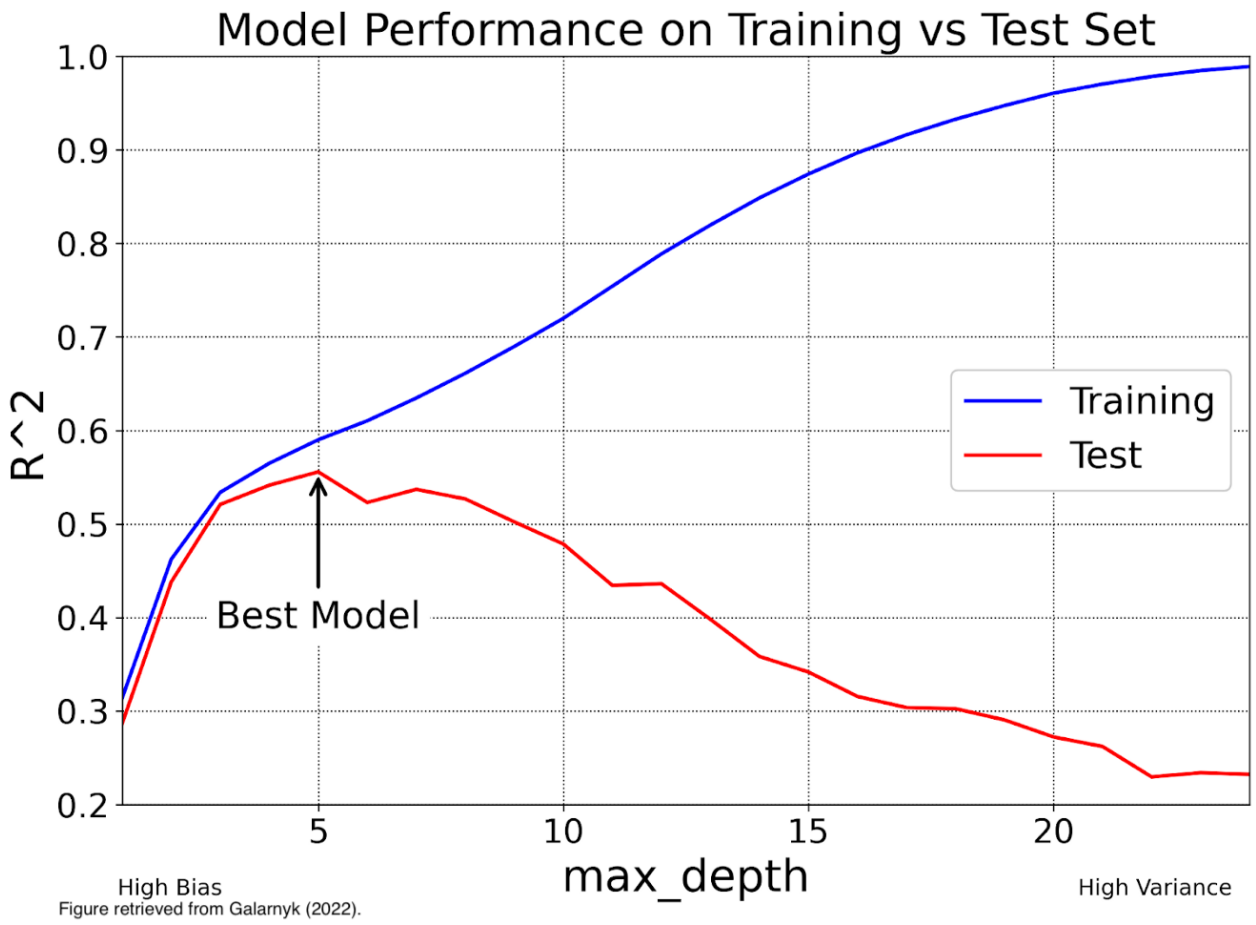
split of the data; 80% used for the training set. I split the remaining 20% into two equally sized sets. The first, I used as an evaluation set, which served two purposes. First, Catboost allows the use of an evaluation set while training the model to prevent overfitting. As the model measures the loss function from one iteration (tree) to the next on the training set, it does so on the evaluation set as well. If at any point, the loss function stops improving for a predefined number of consecutive iterations (in this case, 50), the model stops training and is "shrunk" to the best performing number of iterations. In this way, it learns patterns from the training set and applies them to the evaluation set to detect and prevent overfitting.

In a recent article by Galarnyk (2022), the author provides an excellent illustration of overfit and the optimal point at which we hope to be in when training. Adapted from his own article, Figure 4.2 illustrates the performance of a linear model using R^2. Measured on both the author's training set and test set, he illustrates how a max depth of 5 gives the best performance on the test set, while maintaining a small difference in performance between training and test. Increasing the depth beyond 5 increasingly improves model performance on the training set, but simultaneously decreases model performance on the test set—the best test of model performance we have. What this image illustrates is a classic case of overfitting: Increasing depth beyond 5 causes the model to learn patterns that are unique to the training set and cannot be generalized well to other data. Using the evaluation set in Catboost for overfit detection and early stopping is designed to prevent this from occurring and to converge on meaningful and generalizable patterns.[27] Note, the image also provides an excellent illustration of the bias-variance trade-off. At the left extreme, where differences between the training set and test set are small, we encounter high bias (greater model error in both sets) but low variance (small differences); at the right extreme, we encounter high variance (large differences in performance with high bias in the test set only and very low bias, i.e. model error, in the training set). The best model balances minimal variance and

[27]Reminder, depth is a hyper-parameter that cannot be learned by the machine and the figure by Galarnyk (2022) illustrates an attempt to tune it. My use of an evaluation set refers to the patterns the machine can learn, but the principle is the same.

bias.

Figure 4.2: Optimal Point in Training



Second, in the first few attempts at training, I reviewed the errors the model was making in the evaluation set. Understanding errors helps to identify terms than need to be dropped completely or included separately as stand-alone features. In the process, I also identified an additional 538 bills[28] that had originally been miscoded (and the trained model's classifications were in fact correct). Finally, I was able to determine when the model training had approached the maximum possible improvement. About two thirds of the errors at this point were ones I would not expect the model to handle. For example, short or obscure bill titles that did not include sufficient information, or "arbitrary" rules in the codebook that cannot be gleaned from the text of an observation, for example using the lower number of two codes when it

[28] An oddly meaningful number in the context of Congress.

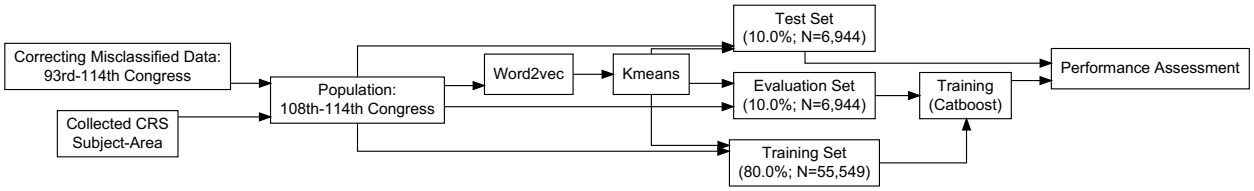
appears both codes are equally dominant in a given observation.

All that remained was a test set of 10% of the original data. This test set was used only for the purpose of assessing model performance, with no attempt to use it as part of the training or to learn from the model's errors in this set. It is crucial to keep a separate set for this purpose to ensure a good estimate of the performance on unseen data.

4.2.4 Strategy: A Summary

Figure 4.3 summarizes the model training strategy. After correcting for previously misclassified data, I limited the population of interest to the last six meetings of Congress. I pre-processed the data using Word2vec and K-means clustering to produce term-clusters as features. I also added a categorical feature of subject-area as coded by the CRS. Next, I split the population of interest into a training set (80%), an evaluation set (10%) and a test set (10%). I trained a supervised Catboost model on the training set and used the evaluation set to prevent overfitting. Once satisfied, I assessed model performance on the test set.

Figure 4.3: Model Training Strategy



I used this approach for the model predicting major categories. I replicated this approach for each model predicting the minor topics within each major topic, mostly with minor changes (e.g. deviating from the 80-10-10 split, when the data did not permit this ratio for reliable training/evaluation). One important way in which the minor topic models deviate from the major topic model is that for the former, I used all data from the 93rd Congress onward. While at the major topic level, relying on data from the most recent 5 or 6 congresses provided a sufficiently large and representative dataset of the most recent examples from each topic, at the minor topic level I was faced with more limited data, in both size and range. Hence my decision to add additional, older data, for the minor topic models.

Table 4.5: Major Topic Model Accuracy

Set	Pre Corrections	Post Corrections	CRS Subject Area
Training set	0.804	0.972	0.975
Evaluation set		0.872	0.898
Test set	0.753	0.875	0.896

4.3 Classification of the Test Set: Pre-Labeled Data

In Table 4.5 I list the accuracy of the model trained to classify major topics at three stages: prior to correcting misclassified data, after correcting the data and after adding the categorical CRS subject area feature to the model. Prior to correcting the data I used only twosets —a training set and a test set—with a standard document-term-matrix (terms as features) and a Random Forest algorithm. After correcting the data, I also opted for a model that uses a Catboost algorithm with clusters of terms following an implementation of Word2vec and K-means clustering.

The improvement is remarkable. With much fewer incorrect data to learn from, and a more advanced modelling strategy, the model is able to learn from the training set in an almost perfect capacity. Applying this model to the evaluation set (used during training to detect and prevent overfit) shows a huge improvement from accurately classifying only 3 of every 4 observations (0.753) to accurately classifying nearly 9 of every 10 observations. This is further improved with the addition of the categorical CRS subject area feature. The most important testament of the model's performance is that these accuracy measures hold when applied to the test data—data that was not used in training in any way.

What these measures mean is that if I were satisfied with a minimal accuracy of 0.9, the model could reliably classify almost all observations in the set and very few would require human review. For a minimal accuracy of 0.95 (my chosen level of accuracy), I use the evaluation set to identify a probability threshold. The evaluation set suggests all observations with a probability of 0.590 or higher, which is assigned to the class the model predicts, will yield 0.95 accuracy. I then confirm that this threshold holds in the test set. Overall, the

model may reliably classify 88% of the test set with an accuracy of 0.95. Within only a few seconds, an overwhelming majority of the data are reliably classified by the model and only a small portion require human review.

In Table 4.6 I list the accuracy thresholds of each of the models trained on the data leading up to the 114th Congress. The first row lists results for the major topic model (corresponding to the final column in Table 4.5) and each of the remaining rows relate to a model classifying the minor topics within a given major topic. In all of the training sets, model accuracy is near perfect with a precision of 0.98 or higher.

Table 4.6: Model Precision by Topic

Model	Training Precision	Evaluation Precision	Test Precision
major_topic_model	0.98	0.9	0.9
Macroeconomics	0.99	0.95	0.95
Civil_Rights	1	0.91	0.96
Health	0.98	0.88	0.87
Agriculture	0.99	0.92	0.91
Labor	1	0.95	0.94
Education	0.99	0.91	0.91
Environment	1	0.9	0.89
Energy	1	0.91	0.91
Transportation	1	0.92	0.92
Law_Crime_Family	0.99	0.88	0.89
Social_Welfare	0.98	0.94	0.93
Housing	0.99	0.93	0.9
Banking	0.98	0.92	0.93
Defense	0.99	0.89	0.89
SSTC	1	0.91	0.89
Trade	0.99	0.93	0.91
International_Affairs	0.99	0.86	0.88
Government_Operations	0.98	0.93	0.93
Public_Lands	1	0.96	0.95

Precision in the evaluation and test sets is usually lower—as is to be expected—but usually still high. Several topics still exhibit high precision of 0.95 or greater and most are greater than 0.90. Only a small number of topics yield lower precision rates (Health, Environment,

Law_Crime_Family, Defense, SSTC & International_Affairs), but even they remain high (minimal precision >= 0.86) considering past success rates. Thus, the models show mostly low variance (slightly higher error rate in test sets compared to training).

4.4 Classification of Unseen Data: Unlabeled Data

The estimated performance illustrated in the previous section is only valid if the unseen data are similar enough to the evaluation set and test set. Sampling the results of classifying the unseen data from the 115th Congress, it became apparent that the data are in fact not as similar as I had hoped. To understand model performance and identify thresholds for 95% precision, I manually reviewed large samples of data from the 115th Congress. Sorting the data by probability, from the lowest to the highest, I reviewed each predicted category separately, correcting the predictions where necessary. Within each category I continued reviewing predictions until I identified at least 100 consecutive observations, 95 of which the machine accurately predicted.

This method is incredibly rigorous and required reviewing more than half of the machine's predictions. Sometimes, especially in early iterations of model training, it is necessary to invest the time and effort in manually reviewing the data. Not only is it important in order to avoid making too many (more than 5% in this case) false predictions, but it is also a useful way of learning about next steps in training a new model. In a multi-classification problem with 20 categories, this is especially crucial and samples that require review based on the first models one trains are often quite large. To improve models such as these, feedback is imperative.

After reviewing nearly 59% of the data, an estimated 80% of the machine's classifications were correct. Considering the challenge of training a model like this and the fact that all training and test data were from Congresses prior to the 115th, this is an excellent outcome. Recall that performance on the test set was estimated at 89% precision—only 9 points higher. Despite the encouraging outcome, to leave these data without manual review would leave too

Table 4.7: Thresholds in 115th Congress

Topic	Threshold	Percent Above	Percent Reviewed
Macroeconomics	0.377	60.411	69.208
Civil Rights	0.688	25.935	99.002
Health	0.453	84.723	21.642
Agriculture	0.159	56.410	86.325
Labor	0.000	100.000	62.500
Education	0.581	72.507	40.970
Environment	0.181	61.336	58.907
Energy	0.227	89.401	33.641
Immigration	0.000	100.000	32.680
Transportation	0.548	23.575	100.000
Law, Crime, Family	0.154	62.632	54.912
Social Welfare	0.050	88.584	57.078
Housing	0.058	80.556	100.000
Finance	0.245	58.681	50.000
Defense	0.151	53.577	63.062
SSTC	0.000	100.000	100.000
Trade	0.000	100.000	100.000
International Affairs	0.171	40.415	100.000
Government Operations	0.664	26.139	79.019
Public Lands	0.574	49.917	58.417
Total		57.898	58.621

many mistaken classifications in the new dataset, and these would be carried forward as I progress to training a model for classifying the 116th Congress.

Through my review I was able to correct nearly 20% of the model's predictions (at the lower end of the probabilies) and I identified a different threshold per topic that better represents these data. In total, I could rely on 58% of the predictions to be accurate at 95% precision or higher, but had to review 59% of the data to guarantee 95% accuracy overall. See thresholds and the number/percent of observations reviewed in each topic in Table 4.7.

Reviewing such a large portion of the data provides rare insights into understanding previously unlabeled data. In Figure 4.4 I plot the density of model score (probability) per major topic, separating between false predictions and true predictions (the latter includes all predictions that I reviewed and found to be true with the addition of all predictions above

the relevant threshold, estimated to be true).

Topics such as Health, Education, Energy and to a lesser degree, Immigration are wonderful illustrations of the distribution we aim for in a model like this. In each of this topics, the two groups hardly overlap, if at all, and each is centered at an opposite end. Other topics indicate a good ability to identify what does not belong in that topic, but yield uniform probabilities for the observations that do belong in that topic. These include Macroeconomics, Civil Rights, Agriculture, Environment, Transportation, Law/Crime/Family, Social Welfare, Housing, Finance, Government Operations and Public Lands. The model performs worst in topics such as Labor, Defense, Trade, International Affairs and SSTC. Note how in Trade and International Affairs, the two groups overlap nearly perfectly, and the in SSTC, the bulk of the true predictions are in fact at the lower end of the scale.

Finally, in Figure 4.5 I plot the proportion of bills in each topic on the congressional agendas. For the sake of comparison, I plot the 114th and 115th Congresses side-by-side. Despite the model's difficulty with a lot of the observations the two consecutive meetings of Congress held a similar agenda. The topics of the day did not change much between the two Congresses.

4.5 Summary

The problem in this chapter is incredibly complicated to address. Any multi-classification problem on its own is more complicated than predicting a dichotomous label. Having to rely almost entirely on the terms used in bills descriptions adds to the complexity of the problem. Considering the challenge, results are very promising but retraining this model to classify the 116th Congress will provide much further work.

The use of K-means clustering on numeric representations of terms substantially improved model performance and reduced its dimensionality. With the addition of the CRS subject area as a single non-textual categorical feature, Catboost was able to perform very well in making good predictions. Although I manually reviewed a large portion of the data, the data

Figure 4.4: Density Plots of Model Scores by Prediction Accuracy

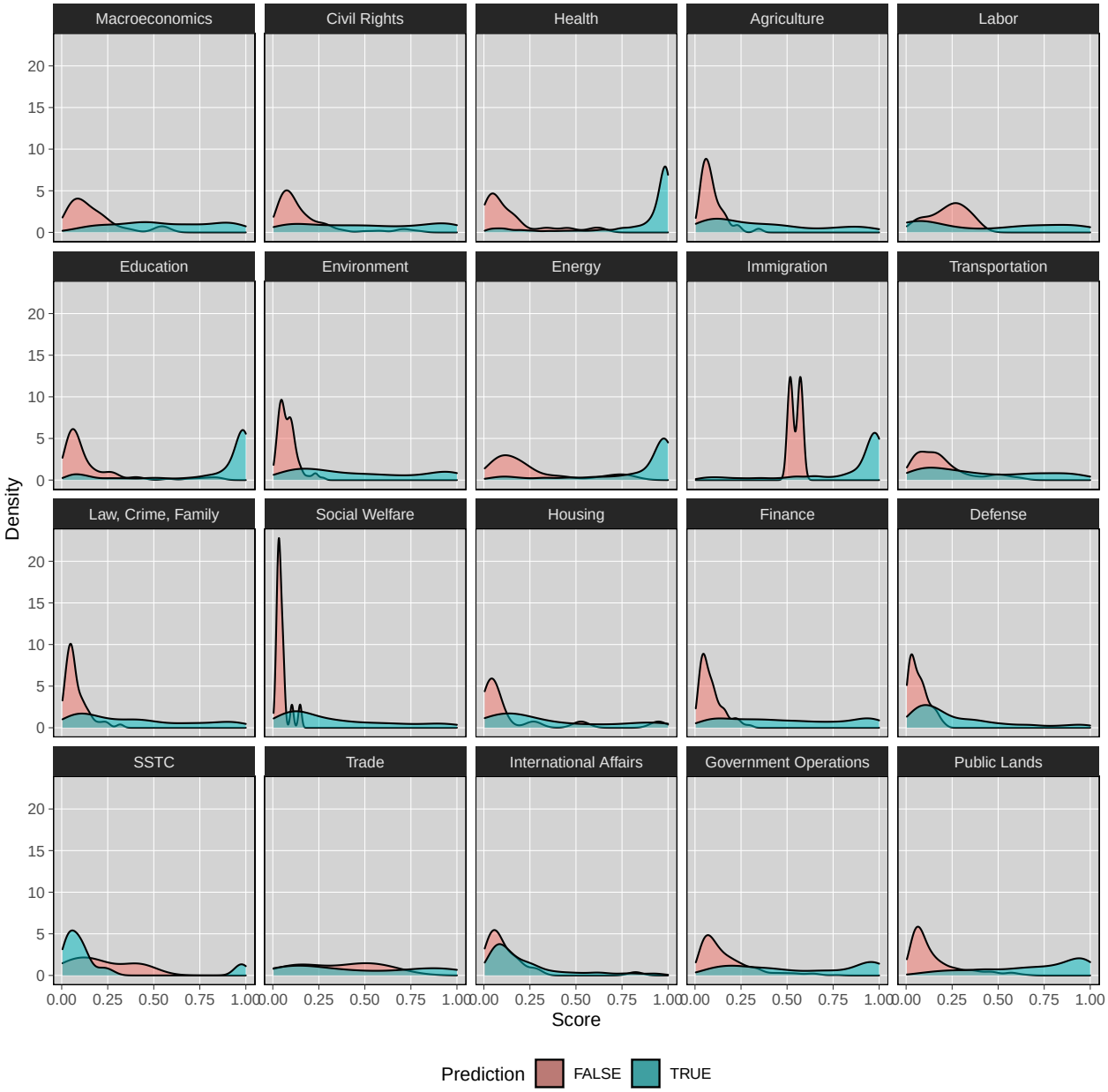
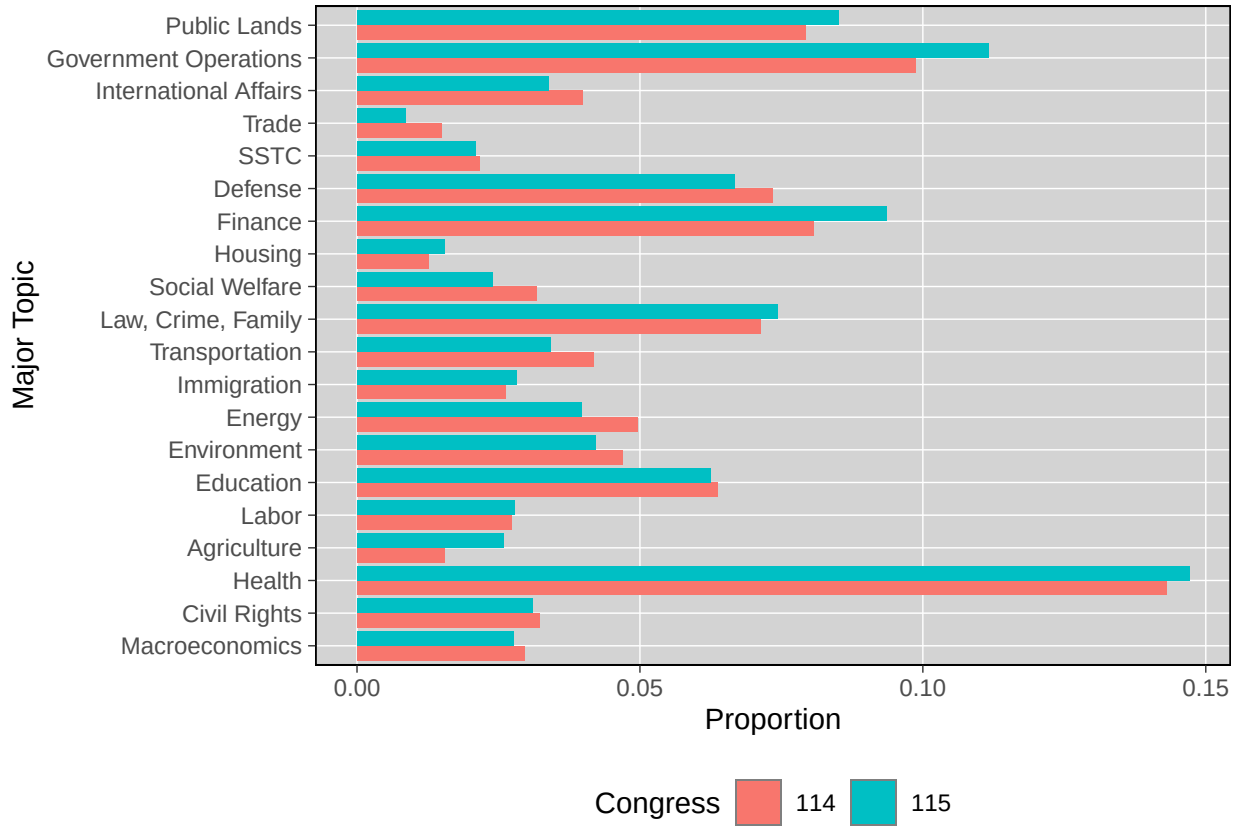


Figure 4.5: The 115th Congressional Policy Agenda



did illustrate how beyond a certain threshold, Catboost did an excellent job in distinguishing between classes. The challenge moving forward is to lower that threshold as much as possible.

Reviewing the data, one thing became apparent to me. Relying on a representative sample to train on may have been a mistake. The outcome of this decision was that some terms were associated with larger categories, not because they were better predictors of that category, but simply because that category appeared more frequently in the training data. In other words, I overfitted the model to the larger categories. As I progress to the 116th Congress, I plan on using a balanced sample with the same number of observations from each topic to address this mistake.

I move to the the next and final chapter, with this notion of moving forward and what lies ahead for machine learning in political science.

5 The Road Ahead: Machine Learning in Political Science

Summarizing the purpose of the preceding chapters, I wish to make the following points. Political science as a whole, and congressional research in particular, have experienced an explosion of data. Such large scale data hold exciting promise for research but present researchers with a difficult challenge of creating reliable measurements. Several scholars have turned to machine learning as a suitable solution and for good reason. The two empirical chapters illustrate just how far machine learning can take us. With relative ease, I was able to train machine learning models for two complicated problems, saving hours of human labor and even improving on existing human-coded data.

In this conclusions chapter, I am interested in the question: Where do we go from here? My dissertation can help answer that crucial question.

5.1 Bridging the Gap: Textbook vs. Machine Learning in Practice

The first wave of machine learning literature in political science introduced to the discipline machine learning algorithms and their potential. For good reason, it did not cover all of the intricacies of machine learning and the challenges that real data present. Instead, articles described what machine learning is, outlined several widely used algorithms and demonstrated potential uses for it in political science.

I believe we are now entering a second wave of machine learning literature, which attempts to unlock the potential of machine learning given the limitations and challenges that real, political, data present. This wave has given birth to articles such as that of Barberá et al. (2021) on the practical use of machine learning and the considerations that go with it, or the groundbreaking work of Grimmer et al. (2022) that demonstrate the contribution of machine learning in every step of empirical research in political science.

In a way, the shift from the first wave to the second wave marks a shift from textbook

data to real data. In textbook data, our data are neatly organized and well balanced to facilitate our own learning.[29] Algorithms perform well on textbook data and it is easy for the user to achieve good results, for example when clusters of data points are easily (and visually) identifiable. Real data—the data we are able to collect for research purposes—are far more complicated. Too little data presents a challenge for statistical inference. Too much data presents a challenge for measurement. Real data for one reason or another have missing observations. In real data, classes may be severely imbalanced. Text as data methods might yield too many features. Training data and unseen data may be distinct from one another in several ways. Real data have mistakes. Clusters in real data sometimes overlap; patterns are not mutually exclusive. Deciding how best to sample data and split between training and test sets is a complicated decision, infused by theoretical and methodological considerations. The list goes on.

I see my dissertation as firmly placed within the start of this second wave. Chapter 3 illustrated several aspects that researchers may face when applying machine learning methods to political data, as well as listed some possible solutions: Balanced sampling for training to handle severely imbalanced data; feature engineering in the face of two politically distinct periods for training and unseen data; the trade-off between precision and recall; and the combination of features from different sources, including text and non-text based features.

Chapter 4 illustrated how to combine supervised and unsupervised methods to reduce model dimensionality and improve prediction using clusters of similar words. It demonstrated how a single problem might be broken up into several different models, each providing a solution to only part of the problem. It also serves as an excellent example of sacrificing model performance for the sake of providing the discipline with a widely-used dataset, while minimizing the risk of endogeneity when used in research. Finally, it demonstrated the challenge of a multi-classification scenario, a scenario that is all to common in a discipline that favors qualitative measurements.

[29]For those familiar, think of datasets such as "iris," "diamonds," "cars," "mtcars," etc.

The shift from textbook data to real data is in fact a leap; one that is not easy to make. Without proper guidance on the solutions to the problems that real data present, we may find ourselves ill-applying machine learning methods, receiving poor results and abandoning a promising method; or perhaps even worse accepting poor machine-based predictions by assuming that that they are good. Consider for example the notion of an 80/20 split offered by textbooks. With real data, we may sometimes be better off using only a small part of the 80% of the potential training data, e.g. in order to balance classes or use only data of the highest certainty in their labels. Or, perhaps other ratios might be preferable to guarantee sufficient data to learn from in training and sufficient data to test on (for example, the minor topic model for Housing in Chapter 4, required a 90/10 ratio).

Even the notion of what should be labeled for successful machine learning projects, could be further developed in the literature. Recently a colleague of mine coded the first 10 years of a dataset spanning 100 years. Her hope was that she could train a model based on the labeled data to predict the remaining 90 years but it performed poorly. Instead, I suggested drawing a random sample of the data, stratified by year/decade in order to yield a sample that better represents the data across time. Labeling such a sample and training a model based on it would provide much better results.

Finally, I have emphasized throughout the dissertation my argument that we should treat the use of machine learning as a process, composed of multiple, consecutive and iterable choices, the consequences of which are evident in both model performance and usage within a research setting. With the increasing trend for data transparency and reproducibility, I expect authors will be required to document and convey such processes in greater detail.

5.2 Trade-Offs in the Academic Practice of Machine Learning

Adopting and importing methods often entail tailoring them to one's needs (e.g. my point about using such methods on real data in political science) and addressing the unique challenges that arise from one's constraints. I have iterated that a machine learning project

consists of trade-offs. In every project, we must choose between precision and recall; we must balance bias and variance in our sets; we must forgo some of our data—sometimes, nearly all of it—to achieve better performance; we must choose between adding features that improve performance and risk introducing endogeneity into subsequent research.

This last trade-off may be somewhat unique to the academic use of machine learning. Usually, especially in the industry, the use of machine learning to make certain predictions is itself the project's goal. When Google uses machine learning to complete users' search texts, when navigation apps predict the best traffic route or when algorithms estimate the likelihood of cancer in test results, the process stops there. In these examples, we're rarely concerned with the downstream effect that such predictions have on causal or relational research. In academia, prediction is used primarily to measure variables in large scale data, that are to be subsequently used in inferential research. We build a model that predicts a in order to subsequently understand the relationship between a and b.

As we incorporate machine learning methods into our research, researchers need to be cognizant of this trade-off at both ends of the spectrum. Those designing machine learning models need to consider how they intend to make use of their outcomes. Those that consume the outcomes need to understand the process that produced them and the risks they pose to their research. Anticipating that researchers might be interested in analyzing differences in the congressional agenda by e.g. party, I excluded such variables from my model. Similarly, a study that explores patterns of agency-creation in Congress, should be careful of examining differences across chambers, because, in my model, I used chamber to make predictions about agency-creation.

5.3 Moving from Specific to All-Purpose Models

Working on this dissertation, I became aware of another trade-off, one I hadn't previously considered. A colleague of mine recently asked if I think she could use the models I trained to predict policy topics on congressional bills to predict policy topics in European party

manifestos. My response was that my models are unlikely to do well in that area for three main reasons. Each of them illustrates the trade-off between a model that can be easily generalized to multiple datasets vs. a model that is trained to do well on a specific type of dataset. The former sacrifices overall performance for the possibility of using it to make predictions in a wider variety of projects. The latter limits the predictions to a specific project, but with much better performance on that particular project.

The first challenge arises from differences across space: The geographical and institutional characteristics of the data that the model was trained on (bills in the American Congress) versus the target data (party manifestos in Europe). The greater the differences between the data the model was trained on and the target, the more we can expect a reduction in performance. One of the greatest achievements of the comparative agendas project was to be able to replicate the coding system offered by the policy agendas project and provide a universal framework for comparative analyses of agendas. But, a unified coding system does not mean that the terms used to describe the issues within each policy topic are the same across countries.

Second, bills and manifestos are different at the data level. That is, they represent different units, target different audiences and may differ linguistically. It is impossible to guarantee performance on a type of data that is different from the data on which the model was trained (but we can at least empirically test performance).

The third challenge is rooted in differences across time. At the major topic level, I chose to train a model based on the 108th-114th (2003-2016) in an effort to predict data from the subsequent 115th Congress. Relying on the most recent data in the bills dataset rested on the assumption that they are more likely to be similar to the unseen data, than previous meetings of Congress. If the manifestos data are drawn from earlier years, my model may be at a disadvantage in predicting policy topics. The minor topics models relied on data reaching back to the 93rd Congress (1973) so differences across time are less pronounced at the minor topic level.

Finally, in a successful effort to boost model performance, I included a single non-textual feature in my models: The CRS subject area associated with each bill. Adding this feature introduced the largest limitation I did not think of ahead of time. The problem that this feature presents is that it limits my ability to use the model on any data that do not have a CRS subject area. Within Congress, the CRS might assign subject areas to other sources aside from bills, but they are primarily used for bills. Outside of Congress, not to mention, the United States, I don't expect CRS subject areas to exist at all. Thus, even if we are able to get past the first three challenges (all of which can be empirically tested) to use the model on manifestos data would mean setting this feature to NA in all observations, yielding reduced performance.

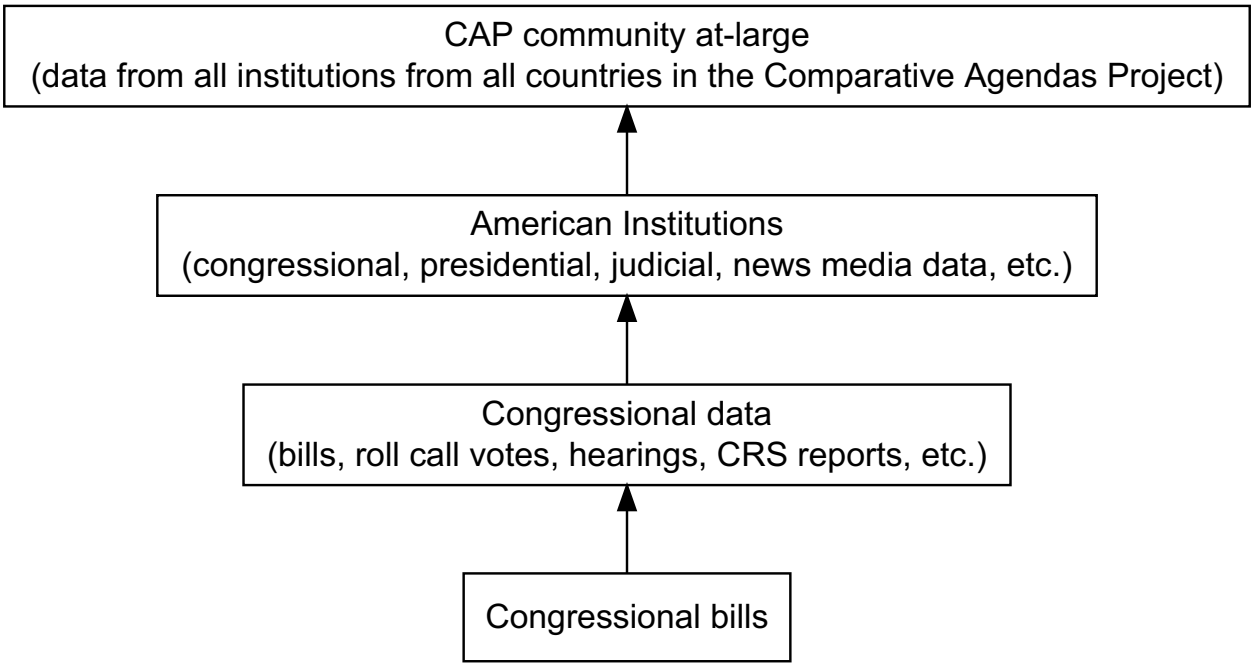
For a model with the purpose of doing well on predicting congressional bill topics, the process I outlined in Chapter 4 is an excellent choice. The bills project is a large enough and complicated enough project to justify its own specific model. Its downside is that my choice of strategy limited its applicability to practically bills only. To design a model that might provide good predictions of policy topics, regardless of the data type, unit of analysis, space or time, requires not only a more diverse collection of data to train on, but also a selection of features that can be widely and easily measured in most datasets.

So far, the use of machine learning algorithms in political science mostly converges on the latter approach—designing dedicated models to address a specific problem. As we accumulate more data, the discipline could benefit from training all-purpose models. For instance, rather than training a model for predicting American policy topics in congressional bills, we may choose to train a more generalized model. Including different types of data in training could allow to to train a model that can be generalized to several different types of data.

The diagram in Figure 5.1 illustrates a hierarchy of generalizeable models. At the bottom is a model predicting policy topics that was trained on congressional bills data only (much like the model I trained in Chapter 4, and can therefore be reliably generalized to congressional bills only.

Adding additional types of congressional data, such as roll call votes, hearings, CRS reports and others we can train a model that may be capable of predicting policy topics in most types of congressional data. Any new dataset that includes congressional data could be reliably classified using this type of model. The price of training such a model would be the collection of additional data, the engineering of features that are easily applicable to most types of congressional data and forgoing any features that are specifically engineered for bills

Figure 5.1: From Specific to All-Purpose Models



Moving up one more step in the hierarchy, we may decide to train a model that predicts policy topics in data from several American institutions. As before, the price would be the collection of additional data, for instance party platforms, State of the Union speeches, executive orders, Supreme Court decisions or news articles and the need to use more easily generalizable features.

At the final step, we could conceive of a model that is an expert at predicting policy topics across geographical space by training on data from all, or several, countries that are members of the Comparative Agendas Project (CAP).

As we move up in the hierarchy we face the same trade-off: Designing a model that

applies more broadly to more types of data, but may perform more poorly on each dataset alone because it cannot account for the unique features that make up each dataset. The benefit could be a standard, uniform model for predicting policy topics across a wide range of data, setting the stage for comparative big-data research sharing the same level of reliability (i.e. even if mistakes exist in the data, they should be consistent across countries/institutions).

As the second wave of machine learning research in political science advances, we may begin to move up the hierarchy and collaborate on these wide-spanning models.

5.4 The Ease and Accessibility of Machine Learning via Code

As quantitative methods have become ingrained into political research, so too has using statistical programming languages become a prominent skill among students of politics. The resources that the personal computer affords today, together with open source statistical programming languages have made machine learning models accessible to whomever may be interested. R packages such as "caret"[30] (Kuhn, 2020) and "tidymodels"[31] (Kuhn & Wickham, 2020) have transformed the ease with which we may train models using R, providing a flexible, powerful and comprehensive framework for models training. The code chunk attached in Appendix A to this chapter, displays example R code for one of the GBM models trained in Chapter 3 (including text-based features only), using the caret package. In less than 100 lines of code, I was able to read in the data, pre-process it, split it into training and test set, train a model and evaluate its performance.

Although R is incredibly popular in political science, advancements in machine learning usually occur first in Python. Students wishing to use the most state-of-the-art algorithms likely need to turn to Python. The package scikit-learn[32] is without out a doubt the most widely used machine learning package in Python today. Several algorithms are accompanied

[30] https://topepo.github.io/caret/
[31] https://www.tidymodels.org/
[32] https://scikit-learn.org/stable/user_guide.html

by their own package, e.g. XGBoost[33] or Catboost[34] and offer comprehensive, flexible and intuitive use of their algorithms, along with exceptional online documentation. In Appendix B to this chapter, I have provided sample Python code, corresponding to the models in Chapter 4. Relying on existing packages, I created classes for identical pre-processing of my training set, evaluation set and test set. Next, I calculated word vectors based on the population data, estimated K-means clustering to form features and trained the model in Catboost.

Machine learning code is widely available online and users are likely to face the problem of having too many code sources to sift through to find relevant solutions to their coding challenges. The natural step, as academia progresses in this direction, is to match source code with methods. As we adopt machine learning methods, and tailor them to our needs—those specified by our data—complementing them with our code will provide the foundation for students to practice machine learning in their own research. Dataverses that store code (and data) for published work will no doubt become useful sources for this purpose, but perhaps more importantly, using sources such as Github to host, share and collaborate on code (a practice that is commonly used in other disciplines and in the industry) could open the door to making machine learning code even more accessible.

5.5 The Model Training Game Plan

As we embark on the second wave of machine learning, I expect to see a growth in the number of students adopting this method. Summing up the lessons I discovered in this dissertation, I leave the reader with the following questions as guidance.

What is the purpose of the model you're training? Training an all-purpose model will require collecting data from various sources, of various types to provide reliable predictions in a variety of datasets. Such a model will rely on features that can be engineered based on the lowest common denominator that these various data sources share, and can therefore be

[33] https://xgboost.readthedocs.io/en/stable/
[34] https://Catboost.ai/en/docs/

quite limiting. A more specific model will likely produce more accurate results for a *specific* type of data and will allow more fine-grained features. However, the ability to generalize it to more data will be severely limited. Such models are likely used only once.

How do unseen data affect your model? Obviously our models never learn from unseen data and instead, we make use of models to make sense of unseen data. We do need to be cognizant of what is available to us in unseen data. For instance, are all data points used to engineer features in the training data, readily measurable in unseen data? Additionally, one of the most challenging aspects of political data is their temporal nature. Temporal changes can limit the use of some features, while offering a range of new features.

What data should your model learn from? Usually, we think the answer to this question is 'all labeled data we have' but the truth is we may prefer only a subset of that data. The best data to learn from should be those whose labels are of the highest certainty. Additionally, we should aim to include observations that make it easier for the machine to learn useful patterns, rather than confuse it (e.g. near-identical observations with different labels). Finally, we should prioritize the data that are likely to provide the best outcome on unseen data. In an all-purpose model, this prioritization may results in training on a variety of data sources. In a specific setting, we may prefer data that are most similar to whatever unseen data I plan on labeling.

What feedback mechanism can you provide the model? One of the challenges in using machine learning models is that once we've used them to label unseen data, we don't have an indication of ground-truth and we sometimes treat the machine's predictions as ground-truth. This approach can lead to misclassified observations and deteriorated model performance down the line. Some areas have built-in feedback mechanism, where ground-truth is supported externally. In most projects I expect researchers to use machine learning, I don't think this is the case. Researchers should be prepared to manually review large portions of the data. As the process of machine learning progresses, we may train a better model, requiring less human review and manual feedback. Feedback can also be useful for detecting a change in

the population emphasizing the need for retraining the model. For example, we may find that a model trained to predict policy topics in congressional bills performs well on three meetings of Congress in a row, but a change in the agenda or balance of power in the fourth meeting, reduced model performance and required a fresh take.

Finally, what downstream effects on research do your choices within the model training process have? My emphasis throughout this dissertation on treating the model training stage as a process reflects the numerous considerations we are faced with when applying machine learning models. We now have the ability of training very powerful models for handling the overload of data in political science. The choices we make along the way affect what each model can and cannot do and how they can be used in research. Addressing these questions can help improve model performance, but more than that, they help us in understanding how to make the most of such models.

Appendices

Appendix A

R demo code for training a model to classify agency-creation in congressional hearing data (corresponds to Chapter 3).[35]

```
# install and load libraries
if(!("xfun" %in% installed.packages())){
  install.packages("xfun")
}
xfun::pkg_attach2(c("readr","tm","tidyverse","tidytext","stringr","caret",
                    "SnowballC","ggplot2","mlbench","plotROC","MLeval",
                    "fmsb","rvest","zoo","gbm"))


# import data ----
hearings <- read_csv("US-Legislative-congressional_hearings-19.4.csv")


# remove missing cases
hearings <- filter(hearings, filter_Agency %in% c(0,1))


# create unique id
hearings$myid <- 1:nrow(hearings)


# pre-process data ----
fulldtm <- as.data.frame(hearings) %>%
  filter(grepl("[a-z]",description, ignore.case = T)) %>%
  unnest_tokens(output = word, input = description) %>%
```

[35] See full code at https://github.com/freedmanguy/agency/blob/main/agency_creation.R.

```
  filter(!str_detect(word, "^[0-9]*$")) %>% # remove numbers
  anti_join(stop_words) %>% # remove stop words
  mutate(word = SnowballC::wordStem(word)) # stem the words

# create documen-term-matrix ----
fulldtm <- fulldtm %>%
  count(myid, word) %>% # count of each word in each observation
  cast_dtm(document = myid, term = word, value = n) # no weights

# remove observations with insufficient text
hearings.text <- hearings %>%
  filter(myid %in% as.numeric(as.character(fulldtm$dimnames$Docs)))

# test set ----
# create data frame of test set (random sample with ration 5:1)
set.seed(2400)
testset2 <- hearings.text %>%
  filter(filter_Agency==1) %>%
  slice_sample(n=419)
testset2 <- hearings.text %>%
  filter(filter_Agency==0) %>%
  slice_sample(n=419*5) %>%
  bind_rows(testset2, .) %>%
  arrange(myid)

# reduce population data to exclude test set
popdata2 <- anti_join(hearings.text,testset2)
```

```
# training set ----
# create data frame of training set (random sample with ration 5:1)
set.seed(2404)
trainingset2 <- popdata2 %>%
  filter(filter_Agency==1)

trainingset2 <- popdata2 %>%
  filter(filter_Agency==0) %>%
  slice_sample(n = nrow(trainingset2)*5) %>%
  bind_rows(trainingset2) %>%
  arrange(myid)

# remove sparce terms
length(fulldtm$dimnames$Terms)
fulldtmS <- fulldtm
fulldtmS <- removeSparseTerms(fulldtmS, sparse = .999)
length(fulldtmS$dimnames$Terms)
fulldtmdf <- as.data.frame(as.matrix(fulldtmS))

# create training set in the form of dtm
mytrain <- fulldtmdf %>%
  mutate(myid = as.numeric(as.character(rownames(.)))) %>%
  filter(myid %in% trainingset2$myid)

# remove observations with insufficient text
trainingset2 <- filter(trainingset2, myid %in% mytrain$myid)
```

```
# remove id variable
mytrain <- mytrain %>%
  select(., -myid) %>%
  as.matrix()

# create test set in the form of dtm
mytest <- fulldtmdf %>%
  mutate(myid = as.numeric(as.character(rownames(.)))) %>%
  filter(myid %in% testset2$myid) %>%
  arrange(myid) %>%
  select(., -myid) %>%
  as.matrix()

# Train model (with final specifications) ----
mygbm50 <- train(x = as.matrix(mytrain), # training set
                 y = factor(trainingset2$filter_Agency, # DV
                            levels = c(0,1),
                            labels = c("NotAgency","Agency")),
                 method = "gbm", # use "ranger" for RF
                 # resampling:
                 trControl = trainControl(method = "repeatedcv",
                                          number = 10,
                                          repeats = 3,
                                          classProbs = T,
                                          savePredictions = T),
                 # tuning parameters:
```

```
                tuneGrid = data.frame(n.trees = 50,
                                      n.minobsinnode = 2,
                                      interaction.depth = 10,
                                      shrinkage = .1))


# Predicted probabilities in training set
training_pred <- mygbm50$pred %>%
  mutate(model = "GBM 50") %>%
  group_by(obs, rowIndex, model) %>%
  summarise(Agency_mean = mean(Agency),
            Agency_sd = sd(Agency),
            NotAgency_mean = mean(NotAgency),
            NotAgency_sd = mean(NotAgency))  %>%
  ungroup()
training_pred$predAgency <- ifelse(
  training_pred$Agency_mean>=quantile(training_pred$Agency_mean, .6),
  "Agency",
  "Not Agency"
  )


# density plot
training_pred %>%
  mutate(Observed = factor(obs,
                           levels = c("NotAgency","Agency"),
                           labels = c("Not Agency","Agency"))) %>%
  ggplot(aes(x = Agency_mean, fill = Observed)) +
  geom_density(alpha = .5) +
```

```
  labs(x = "Pr(Agency==1)") +
  theme(legend.position = "bottom")


# ROC curve
evalm(mygbm50)$roc
```

Appendix B

Python demo code for training a model to classify PAP topics in congressional bills data (corresponds to Chapter 4).[36]

```
# packages
import Catboost
import gensim.models
import gap_statistic
import nltk
import re
import copy
import pandas as pd
import numpy as np
import gensim.downloader as wv
from gensim import utils
from scipy.stats import randint
from sklearn.model_selection import RandomizedSearchCV
from sklearn.pre-processing import StandardScaler
from nltk.stem import PorterStemmer
from nltk.corpus import stopwords
```

[36] Code available at https://github.com/freedmanguy/cbp/blob/main/Example.ipynb.

```
from nltk import download
from nltk.cluster import KMeansClusterer
download('stopwords')

# class for pre-processing dta - stemming, tokenizing, removing stop words
porter = PorterStemmer()

class PrepareSentance():
    def __init__(self, df, text_column):
        self.df = df
        self.text_column = text_column
        self.processed_df = []

    def tokenize(self, stem=True, remove_stopwords=True):
        df = self.df.copy()
        text_column = self.text_column
        processed_df = [utils.simple_pre-process(t) for t in df[text_column]]
        if remove_stopwords:
            stop_words = set(stopwords.words('english'))
            for i in range(len(processed_df)):
                processed_df[i] = [
                  w for w in processed_df[i] if not w in stop_words
                  ]
        if stem:
            for i in range(len(processed_df)):
                processed_df[i] = [porter.stem(p) for p in processed_df[i]]
        self.processed_df = processed_df
```

```
        return processed_df



# classes for pre-processing training/test sets
class TabularDescription():
    def __init__(self, dataset, text_column, word_clusters,
    single_words=None):
        self.set = dataset
        self.text_column = text_column
        self.word_clusters = word_clusters
        self.single_words = single_words
        self.x = None
        self.y = None
        self.training_features = None
        if 'congress_gov_major_topic' in list(dataset.columns):
            congress_subject_area = dataset[
              ['billid','congress_gov_major_topic']
              ]
            congress_subject_area = congress_subject_area.set_index(
              keys='billid'
              )
            self.congress_subject_area = congress_subject_area
        else:
            self.congress_subject_area = None

    def get_dataset(self):
        return self.set
```

```
    def get_text_column(self):
        return self.text_column

    def get_word_clusters(self):
        return self.word_clusters

    def get_single_words(self):
        return self.single_words

    def get_congress_subject_area(self):
        return self.congress_subject_area

    def get_training_features(self):
        tsf = copy.deepcopy(self.training_features)
        return tsf


class TabularDescriptionTrain(TabularDescription):
    def __init__(self, dataset, text_column, word_clusters,
    single_words=None):
        super().__init__(dataset, text_column, word_clusters,
        single_words=single_words)

    def prepare_set_for_training(self, stem=True, remove_stopwords=True):
        ts = self.get_dataset()
        tc = self.get_text_column()
```

```
        ts_pp = PrepareSentance(df=ts, text_column=tc)
        ts_t = ts_pp.tokenize(stem=stem, remove_stopwords=remove_stopwords)

        billid = [
          [ts.billid[b]] * len(ts_pp.processed_df[b]) for b in range(
            len(ts_pp.processed_df)
            )
          ]
        ts_train = pd.DataFrame({
            'billid' : [item for bill in billid for item in bill],
            'term' : [term for title in ts_pp.processed_df for term in title]
        })

        word_clusters = self.get_word_clusters()
        ts_train = ts_train.merge(right=word_clusters, how='left')
        ts_train = ts_train.astype(str)
        ts_train['cluster_name'] = 'c_'
        ts_train.cluster_name = ts_train.cluster_name.str.cat(
          ts_train.cluster
          )

        ts_dtm = ts_train.groupby(
          ['billid', 'cluster_name']
          ).size().reset_index()
        ts_dtm = ts_dtm.rename(columns={0:'n'})
        ts_dtm = ts_dtm.pivot(
          index="billid", columns="cluster_name", values="n"
```

```
      ).fillna(0)
    ts_dtm = ts_dtm.drop(labels='c_nan', axis=1)

    sw = self.get_single_words()
    if sw is not None:
        ts_sw = ts_train.merge(right=sw, how='inner')
        ts_sw = pd.DataFrame(ts_sw.groupby(['billid', 'term']).size())
        ts_sw = ts_sw.reset_index()
        ts_sw.columns = ['billid', 'term', 'n']
        ts_sw = ts_sw.pivot(index="billid", columns="term", values="n")
        ts_dtm = ts_dtm.merge(right=ts_sw, left_index=True,
                              right_index=True, how='left').fillna(0)

    sa = self.get_congress_subject_area()
    if sa is not None:
        ts_dtm = ts_dtm.merge(right=sa, left_index=True,
                              right_index=True, how='left')

    y = pd.DataFrame({'billid':ts_dtm.index}).merge(
      right=ts[['billid', 'minor']], how='left', on='billid')
    y = y.astype('str')
    y = list(y['minor'])

    self.x = ts_dtm
    self.y = y
    self.training_features = list(ts_dtm.columns)
    return ts_dtm, y
```

```
class TabularDescriptionTest(TabularDescription):
    def __init__(self, dataset, text_column, word_clusters,
    training_features, single_words=None):
        super().__init__(dataset, text_column, word_clusters,
        single_words=single_words)
        self.training_features = training_features

    def prepare_set_for_evaluation(self, stem=True, remove_stopwords=True):
        ts = self.get_dataset()
        tc = self.get_text_column()
        ts_pp = PrepareSentance(df=ts, text_column=tc)
        ts_t = ts_pp.tokenize(stem=stem, remove_stopwords=remove_stopwords)

        billid = [
          [ts.billid[b]] * len(ts_pp.processed_df[b]) for b in range(
            len(ts_pp.processed_df)
            )
          ]
        ts_train = pd.DataFrame({
            'billid' : [item for bill in billid for item in bill],
            'term' : [term for title in ts_pp.processed_df for term in title]
        })

        word_clusters = self.get_word_clusters()
        ts_train = ts_train.merge(right=word_clusters, how='left')
```

```
        ts_train = ts_train.astype(str)
        ts_train['cluster_name'] = 'c_'
        ts_train.cluster_name = ts_train.cluster_name.str.cat(
          ts_train.cluster
          )

        ts_dtm = ts_train.groupby(
          ['billid', 'cluster_name']
          ).size().reset_index()
        ts_dtm = ts_dtm.rename(columns={0:'n'})
        ts_dtm = ts_dtm.pivot(index="billid", columns="cluster_name",
                              values="n").fillna(0)
        ts_dtm = ts_dtm.drop(labels='c_nan', axis=1)

        sw = self.get_single_words()
        if sw is not None:
            ts_sw = ts_train.merge(right=sw, how='inner')
            ts_sw = pd.DataFrame(ts_sw.groupby(['billid', 'term']).size())
            ts_sw = ts_sw.reset_index()
            ts_sw.columns = ['billid', 'term', 'n']
            ts_sw = ts_sw.pivot(index="billid", columns="term", values="n")
            ts_dtm = ts_dtm.merge(right=ts_sw, left_index=True,
                                  right_index=True, how='left').fillna(0)

        ts_columns = set(ts_dtm.columns)
        training_features = set(self.get_training_features())
        missing_features = training_features.difference(ts_columns)
```

```
        number_of_columns = len(missing_features)
        if number_of_columns > 0:
            number_of_rows = len(ts_dtm)
            missing_features_a = np.zeros((number_of_rows, number_of_columns))
            missing_features_df = pd.DataFrame(missing_features_a)
            missing_features_df.columns = missing_features
            missing_features_df.index = ts_dtm.index
            ts_dtm = pd.concat([ts_dtm, missing_features_df], axis = 1)

        columns_to_keep = copy.deepcopy(self.get_training_features())
        sa = self.get_congress_subject_area()
        if sa is not None:
            columns_to_keep.remove('congress_gov_major_topic')
        ts_dtm = ts_dtm[columns_to_keep]

        if sa is not None:
            ts_dtm = ts_dtm.merge(right=sa, left_index=True,
                                  right_index=True, how='left')

        y = pd.DataFrame({'billid':ts_dtm.index}).merge(
          right=ts[['billid', 'minor']], how='left', on='billid')
        y = y.astype('str')
        y = list(y['minor'])

        self.x = ts_dtm
        self.y = y
        return ts_dtm, y
```

```
# function for evaluating model performance
def get_classification_results(cbm_model, x, y=None):
    pred_prob_set = pd.DataFrame(cbm_model.predict_proba(x))
    pred_prob_set.columns = cbm_model.classes_
    pred_prob_set['probability'] = pred_prob_set.max(1)
    pred_prob_set['predicted'] = cbm_model.predict(x)
    if y is not None:
        pred_prob_set['observed'] = y
        pred_prob_set['match'] = [
          True if pred_prob_set['observed'][i] == \
          pred_prob_set['predicted'][i] else False \
          for i in pred_prob_set.index
          ]
    pred_prob_set['billid'] = list(x.index)
    return pred_prob_set


# files (prepared in advance)
population_csv_file = 'population_93_114.csv'
training_csv_file = 'training_80.csv'
evaluation_csv_file = 'evaluation_set_80.csv'
test_csv_file = 'test_set_80.csv'
# data
population = pd.read_csv(population_csv_file)
training_set = pd.read_csv(training_csv_file)
evaluation_set = pd.read_csv(evaluation_csv_file)
test_set = pd.read_csv(test_csv_file)
```

```
# word vectors on population data
population = population.query('congress > 107')
population['title'] = [re.sub('united states code|other purposes', '', x,
                      flags=re.IGNORECASE) for x in population['title']]

population_pp = PrepareSentance(df=population, text_column='title')
population_t = population_pp.tokenize()
for i in population_t:
    for j in ['amend', 'act', 'bill', 'oper', 'implement', 'program', 'titl',
              'administration', 'american', 'institut','department',
              'secretari', 'offic']:
        try:
            i.remove(j)
        except:
            pass

population_t
population_gm = gensim.models.Word2Vec(sentences=population_t, min_count=3,
                                       vector_size=300)
dfwv = pd.DataFrame(population_gm.wv.vectors)
dfwv.index = population_gm.wv.index_to_key

# K-means clustering based on word vectors
optimalK = gap_statistic.OptimalK(n_jobs=4, parallel_backend='joblib')
n_clusters = optimalK(dfwv, cluster_array=np.arange(1, 350))
n_clusters
```

```
optimalK.plot_results()
X = population_gm.wv.vectors
NUM_CLUSTERS=n_clusters
kclusterer = KMeansClusterer(NUM_CLUSTERS,
                             distance=nltk.cluster.util.cosine_distance,
                             repeats=25)
assigned_clusters = kclusterer.cluster(X, assign_clusters=True)
word_clusters = pd.DataFrame({'term':population_gm.wv.index_to_key,
                              'cluster':assigned_clusters})
word_clusters.to_csv('word_clusters_edit.csv')

# list of terms to use as single words, outside of clusters
single_words = pd.DataFrame({'term':['safeti', 'transit', 'secur', 'effici',
                                     'job', 'vehicl', 'reimburs', 'construct'
                                    'research', 'school', 'compet', 'youth',
                                    'young', 'clean', 'production',
                                    'power']})

# training
training_ins = TabularDescriptionTrain(dataset=training_set,
                                       text_column='title',
                                       word_clusters=word_clusters,
                                       single_words=single_words)
ts_x, ts_y = training_ins.prepare_set_for_training()
training_ins.get_training_features()

# evaluation
```

```
evaluation_ins = TabularDescriptionTest(
  dataset=evaluation_set,
  text_column='title',
  word_clusters=word_clusters,
  training_features=training_ins.get_training_features(),
  single_words=single_words
  )
ev_x, ev_y = evaluation_ins.prepare_set_for_evaluation()


# test set
test_ins = TabularDescriptionTest(
  dataset=test_set,
  text_column='title',
  word_clusters=word_clusters,
  training_features=training_ins.get_training_features(),
  single_words=single_words
  )
test_x, test_y = test_ins.prepare_set_for_evaluation()


# train Catboost
major_model_80 = Catboost.CatboostClassifier(
  iterations=18000, max_depth=10,
  learning_rate=0.025,
  l2_leaf_reg=0.75,
  loss_function='MultiClassOneVsAll',
  rsm=0.2,
  cat_features=['congress_gov_major_topic']
```

```
)


major_model_80_mt = major_model_80.fit(
  X=training_80_dtm, y=y_training_80,
  early_stopping_rounds=50,
  eval_set=(validation_80_dtm, y_validation_80)
)


# shrink number of iterations based on early detection of overfit
major_model_80_mt.shrink(9234)


# accuracy
ts_set_acc = major_model_80_mt.score(X=training_80_dtm, y=y_training_80)
eval_acc = major_model_80_mt.score(X=validation_80_dtm, y=y_validation_80)
test_set_acc = major_model_80_mt.score(X=test_set_80_dtm, y=y_test_set_80)


print(f'training set accuracy: {ts_set_acc}')
print(f'validation set accuracy: {eval_acc}')
print(f'test set accuracy: {test_set_acc}')


# predictions
pred = get_classification_results(cbm_model = model,
                                  x=test_set_80_dtm, y=y_test_set_80)
```

References

Abram, M., & Cooper, J. (1968). The rise of seniority in the house of representatives. *Polity*, *1*(1), 5285.

Adler, E. S., & Wilkerson, J. D. (2008). Intended consequences: Jurisdictional reform and issue control in the US house of representatives. *Legislative Studies Quarterly*, *33*(1), 85112.

Adler, E. S., & Wilkerson, J. D. (2013). *Congress and the politics of problem solving*. Cambridge University Press.

Aitchison, J. (2001). *Language change: Progress or decay?* Cambridge university press.

Aldrich, J. H. (1999). Political parties in a critical era. *American Politics Quarterly*, *27*(1), 932.

Aldrich, J. H., & Niemi, R. G. (2018). *The sixth american party system: Electoral change, 19521992* (p. 87109). Routledge.

Aldrich, J. H., & Rohde, D. (2001). *The logic of conditional party government: Revisiting the electoral connection* (L. C. Dodd & B. I. Oppenheimer, Eds.; 7th ed.). CQ Press.

Amethiya, Y., Pipariya, P., Patel, S., & Shah, M. (2021). Comparative analysis of breast cancer detection using machine learning and biosensors. *Intelligent Medicine*.

Anastasopoulos, L. J., Badani, D., Lee, C., Ginosar, S., & Williams, J. (2016). Photographic home styles in congress: A computer vision approach. *arXiv Preprint arXiv:1611.09942*.

Anastasopoulos, L. J., & Bertelli, A. M. (2020). Understanding delegation through machine learning: A method and application to the european union. *American Political Science Review*, *114*(1), 291301.

Antwarg, L., Miller, R. M., Shapira, B., & Rokach, L. (2021). Explaining anomalies detected by autoencoders using Shapley Additive Explanations. *Expert Systems with Applications*, *186*, 115736. https://doi.org/10.1016/j.eswa.2021.115736

Argersinger, P. (1992). *Structure, process, and party: Essays in american political history*. ME Sharpe Inc.

Bachrach, P., & Baratz, M. S. (1962). Two faces of power. *American Political Science Review*, *56*(4), 947952.

Barberá, P., Boydstun, A. E., Linn, S., McMahon, R., & Nagler, J. (2021). Automated Text Classification of News Articles: A Practical Guide. *Political analysis*, *29*(1), 19–42.

Baumgartner, F. R., Breunig, C., & Grossman, E. (2019). *Comparative policy agendas: Theory, tools, data*. Oxford University Press.

Baumgartner, F. R., De Boef, S. L., & Boydstun, A. E. (2008). *The decline of the death penalty and the discovery of innocence*. Cambridge University Press.

Baumgartner, F. R., & Jones, B. D. (2015). *The politics of information: Problem definition and the course of public policy in america*. University of Chicago Press.

Baumgartner, F. R., Jones, B. D., & MacLeod, M. C. (2000a). The evolution of legislative jurisdictions. *Journal of Politics*, *62*(2), 321–349.

Baumgartner, F. R., Jones, B. D., & MacLeod, M. C. (2000b). The evolution of legislative jurisdictions. *Journal of Politics*, *62*(2), 321–349.

Binder, S. A. (1999). The dynamics of legislative gridlock, 194796. *American Political Science Review*, *93*(3), 519533.

Black, E., & Black, M. (2009). *The rise of southern republicans*. Harvard University Press.

Bloch Rubin, R. (2013). Organizing for insurgency: Intraparty organization and the development of the house insurgency, 19081910. *Studies in American Political Development*, *27*(2), 86110.

Bonica, A. (2018). Inferring roll-call scores from campaign contributions using supervised machine learning. *American Journal of Political Science*, *62*(4), 830848.

Boydstun, A. E. (2013). *Making the news: Politics, the media, and agenda setting*. University of Chicago Press.

Brady, H. E. (2019). The challenge of big data and data science. *Annual Review of Political Science*, *22*, 297323.

Brewer, M. D., & Stonecash, J. M. (2009). *Dynamics of american political parties*. Cambridge University Press.

Buerki, A. (2019). Furiously fast: On the speed of change in formulaic language. *Yearbook of Phraseology*, *10*(1), 538.

Burnham, W. D. (1965). The changing shape of the american political universe. *American Political Science Review*, *59*(1), 728.

Canon, D. T., & Stewart, C. H. (2001). *The evolution of the committee system in congress* (L. C. Dodd & B. I. Oppenheimer, Eds.; 7th ed., pp. 163–190). CQ Press.

Cardie, C., & Wilkerson, J. (2008). *Text annotation for political science research*.

Cavari, A., & Freedman, G. (2021). *American public opinion toward israel: From consensus to divide*. Routledge.

Chen, N.-C., Drouhard, M., Kocielnik, R., Suh, J., & Aragon, C. R. (2018). Using machine learning to support qualitative coding in social science: Shifting the focus to ambiguity. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, *8*(2), 120.

Chen, Y., Garnett, R., & Montgomery, J. M. (2022). Polls, context, and time: A dynamic hierarchical bayesian forecasting model for US senate elections. *Political Analysis*, 121. https://doi.org/10.1017/pan.2021.42

Cohen, R. E. (1999). *Rostenkowski: The pursuit of power and the end of the old politics*. University of Chicago Press.

Collingwood, L., & Wilkerson, J. (2012). Tradeoffs in accuracy and efficiency in supervised learning methods. *Journal of Information Technology & Politics*, *9*(3), 298318.

Cooper, J. (1988). *Congress and its committees*. Garland.

Cooper, J., & Brady, D. W. (1981). Institutional context and leadership style: The house from cannon to rayburn. *American Political Science Review*, *75*(2), 411425.

Cox, G. W., & McCubbins, M. D. (2005). *Setting the agenda: Responsible party government in the US house of representatives*. Cambridge University Press.

Cranmer, S. J., & Desmarais, B. A. (2017). What Can We Learn from Predictive Modeling? *Political analysis*, *25*(2), 145–166.

D'Orazio, V., Landis, S. T., Palmer, G., & Schrodt, P. (2014). Separating the Wheat from the Chaff: Applications of Automated Document Classification Using Support Vector Machines. *Political analysis*, *22*(2), 224–242.

Dahl, R. A. (1961). *Who governs?: Democracy and power in an american city*. Yale University Press.

Davidson, R. H. (1990). The advent of the modern congress: The legislative reorganization act of 1946. *Legislative Studies Quarterly*, *15*(3), 357373.

Deering, C. J., & Smith, S. S. (1997). *Committees in congress*. CQ Press.

DeGregorio, C. (1994). Professional committee staff as policymaking partners in the US congress. *Congress & the Presidency*, *21*(1), 4965.

Denny, M. J., & Spirling, A. (2018). Text preprocessing for unsupervised learning: Why it matters, when it misleads, and what to do about it. *Political Analysis*, *26*(2), 168189.

Diermeier, D., Godbout, J.-F., Yu, B., & Kaufmann, S. (2012). Language and ideology in congress. *British Journal of Political Science*, *42*(1), 3155.

Dorogush, A. V., Ershov, V., & Gulin, A. (2018). CatBoost: Gradient boosting with categorical features support. *arXiv Preprint arXiv:1810.11363*.

Dowding, K., & Miller, C. (2019). On prediction in political science. *European Journal of Political Research*, *58*(3), 10011018.

Downs, A. (1957). *An economic theory of democracy*. Harper & Row.

Dun, L., Soroka, S., & Wlezien, C. (2021). Dictionaries, supervised learning, and media coverage of public policy. *Political Communication*, *38*(1-2), 140158.

Epp, D. A. (2018). Policy Agendas and Economic Inequality in American Politics. *Political Studies*, *66*(4), 922–939. https://doi.org/10.1177/0032321717736951

Erlich, A., Dantas, S. G., Bagozzi, B. E., Berliner, D., & Palmer-Rubin, B. (2021). Multi-Label Prediction for Political Text-as-Data. *Political analysis*, 1–18.

Evans, C. L. (1999). Legislative structure: Rules, precedents, and jurisdictions. *Legislative Studies Quarterly*, 605642.

Fagan, E., & Shannon, B. (2020). Using the comparative agendas project to examine interest group behavior. *Interest Groups & Advocacy*, *9*(3), 361372.

Fenno, R. F. (1966). *The power of the purse: Appropriations politics in congress*. Little, Brown.

Fenno, R. F. (1977). US house members in their constituencies: An exploration. *American Political Science Review*, *71*(3), 883917.

Foner, E. (1988). *Reconstruction: America's unfinished revolution, 18631877*. Harper & Row.

Fong, C., & Tyler, M. (2021). Machine Learning Predictions as Regression Covariates. *Political analysis*, *29*(4), 467–484.

Galarnyk, M. (2022). *Understanding train test split (scikit-learn + python)*. https://towardsdatascience.com/understanding-train-test-split-scikit-learn-python-ea676d5e3d1

Gamson, W. A., & Modigliani, A. (1989). Media discourse and public opinion on nuclear power: A constructionist approach. *American Journal of Sociology*, *95*(1), 137.

Geisser, S. (1975). The predictive sample reuse method with applications. *Journal of the American Statistical Association*, *70*(350), 320328.

Giudici, P., & Raffinetti, E. (2021). Shapley-lorenz eXplainable artificial intelligence. *Expert Systems with Applications*, *167*, 114104. https://doi.org/https://doi.org/10.1016/j.eswa.2020.114104

Green, D. P., & Kern, H. L. (2012). Modeling heterogeneous treatment effects in survey experiments with bayesian additive regression trees. *Public Opinion Quarterly*, *76*(3), 491511.

Greene, K. T., Park, B., & Colaresi, M. (2019). Machine Learning Human Rights and Wrongs: How the Successes and Failures of Supervised Learning Algorithms Can Inform the Debate About Information Effects. *Political analysis*, *27*(2), 223–230.

Grimmer, J. (2015). We are all social scientists now: How big data, machine learning, and causal inference work together. *PS: Political Science & Politics*, *48*(1), 8083.

Grimmer, J., & King, G. (2011). General purpose computer-assisted clustering and conceptualization. *Proceedings of the National Academy of Sciences*, *108*(7), 26432650.

Grimmer, J., Messing, S., & Westwood, S. J. (2012). How words and money cultivate a personal vote: The effect of legislator credit claiming on constituent credit allocation. *American Political Science Review*, *106*(4), 703719.

Grimmer, J., Roberts, M. E., & Stewart, B. M. (2021). Machine learning for social science: An agnostic approach. *Annual Review of Political Science*, *24*, 395419.

Grimmer, J., Roberts, M. E., & Stewart, B. M. (2022). *Text as data: A new framework for machine learning and the social sciences*. Princeton University Press.

Grossman, J., & Pedahzur, A. (2020). Political science and big data: Structured data, unstructured data, and how to use them. *Political Science Quarterly*, *135*(2), 225257.

Hajare, P., Kamal, S., Krishnan, S., & Bagavathi, A. (2021). *A machine learning pipeline to examine political bias with congressional speeches*. 239243.

Hardin, G. (1968). The tragedy of the commons. *Science*, *162*(3859), 12431248. http://www.jstor.org/stable/1724745

Hartigan, J. A. (1975). *Clustering algorithms*. John Wiley; Sons.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series c (Applied Statistics)*, *28*(1), 100–108.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (Vol. 2). Springer.

Hawley, E. W. (2015). *The new deal and the problem of monopoly*. Princeton University Press.

Hetherington, M. J. (2009). Putting polarization in perspective. *British Journal of Political Science*, *39*(2), 413448.

Heuillet, A., Couthouis, F., & Díaz-Rodríguez, N. (2021). Explainability in deep reinforcement learning. *Knowledge-Based Systems*, *214*, 106685. https://doi.org/10.1016/j.knosys.2020.106685

Hillard, D., Purpura, S., & Wilkerson, J. (2008). Computer-assisted topic classification for mixed-methods social science research. *Journal of Information Technology & Politics*, *4*(4), 3146.

Imai, K., & Khanna, K. (2016). Improving ecological inference by predicting individual ethnicity from voter registration records. *Political Analysis*, *24*(2), 263272.

Jacobson, G. C. (2000). *2000. Party polarization in national politics: The electoral connection* (J. R. Bond & R. Fleisher, Eds.; p. 930). Congressional Quarterly Press.

Jacoby, W. G. (2000). Issue framing and public opinion on government spending. *American Journal of Political Science*, 750767.

Jones, B. D. (2016). The comparative policy agendas projects as measurement systems: Response to dowding, hindmoor and martin. *Journal of Public Policy*, *36*(1), 3146.

Jones, B. D., & Baumgartner, F. R. (2004). Representation and Agenda Setting. *Policy Studies Journal*, 25.

Jones, B. D., Baumgartner, F. R., & Talbert, J. C. (1993a). The destruction of issue monopolies in congress. *American Political Science Review*, *87*(3), 657671.

Jones, B. D., Baumgartner, F. R., & Talbert, J. C. (1993b). The destruction of issue monopolies in congress. *American Political Science Review*, *87*(3).

Jones, B. D., Theriault, S. M., & Whyman, M. (2019). *The great broadening: How the vast expansion of the policymaking agenda transformed american politics*. University of Chicago Press.

Jones, D. R. (2001). Party polarization and legislative gridlock. *Political Research Quarterly*, *54*(1), 125141.

Karol, D., & Hershey, M. R. (2014). *Parties revised and revived: Democrats and republicans in the age of reagan, 1980-2000*. CQ Press.

Kastellec, J. P. (2010). The statistical analysis of judicial decisions and legal rules with classification trees. *Journal of Empirical Legal Studies*, *7*(2), 202230.

Kaufman, A. R., Kraft, P., & Sen, M. (2019). Improving Supreme Court Forecasting Using Boosted Decision Trees. *Political analysis*, *27*(3), 381–387.

Key, V. O. (1959). Secular realignment and the party system. *The Journal of Politics*, *21*(2), 198210.

King, D. C. (1997). *Turf wars: How congressional committees claim jurisdiction*. University of Chicago Press.

King, G., Lam, P., & Roberts, M. E. (2017). Computer-assisted keyword and document set discovery from unstructured text. *American Journal of Political Science*, *61*(4), 971988.

King, G., Pan, J., & Roberts, M. E. (2013). How censorship in china allows government criticism but silences collective expression. *American Political Science Review*, *107*(2), 326343.

Kuhn, M. (2020). *Caret: Classification and regression training*. https://CRAN.R-project.org/package=caret

Kuhn, M., & Wickham, H. (2020). *Tidymodels: A collection of packages for modeling and machine learning using tidyverse principles.*

Lawrence, E. D. (2013). The publication of precedents and its effect on legislative behavior. *Legislative Studies Quarterly*, *38*(1), 3158.

Layman, G. C., Carsey, T. M., & Horowitz, J. M. (2006). PARTY POLARIZATION IN AMERICAN POLITICS: Characteristics, Causes, and Consequences. *Annual Review of Political Science*, *9*(1), 83–110. https://doi.org/10.1146/annurev.polisci.9.070204.105138

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436444.

Lee, F. E. (2008). Agreeing to disagree: Agenda content and senate partisanship, 19812004. *Legislative Studies Quarterly*, *33*(2), 199222.

Lee, F. E. (2018). *Bicameral representation* (G. C. Edwards, F. E. Lee, & E. Schickler, Eds.). Oxford University Press.

Leuchtenburg, W. E. (1963). *Franklin d. Roosevelt and the new deal, 1932-1940*. Harper & Row.

Lewallen, J. (2020). *Committees and the decline of lawmaking in congress*. University of Michigan Press.

Lewallen, J., Theriault, S. M., & Jones, B. D. (2016). Congressional dysfunction: An information processing perspective: Congressional dysfunction and hearings. *Regulation & Governance*, *10*(2), 179–190. https://doi.org/10.1111/rego.12090

Llanos, M., & Nolte, D. (2003). Bicameralism in the americas: Around the extremes of symmetry and incongruence. *The Journal of Legislative Studies*, *9*(3), 5486.

Lundberg, S. M., Erion, G. G., & Lee, S.-I. (2019). Consistent Individualized Feature Attribution for Tree Ensembles. *arXiv:1802.03888 [Cs, Stat]*. http://arxiv.org/abs/1802.03888

Lundberg, S. M., & Lee, S.-I. (2017). *31st Conference on Neural Information Processing Systems*. 10.

Mann, T. E., & Ornstein, N. J. (2006). *The broken branch: How congress is failing america and how to get it back on track*. Oxford University Press.

Marcilio, W. E., & Eler, D. M. (2020). *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. https://doi.org/10.1109/SIBGRAPI51738.2020.00053

Marcílio, W. E., & Eler, D. M. (2021). Explaining dimensionality reduction results using Shapley values. *Expert Systems with Applications*, *178*, 115020. https://doi.org/10.1016/j.eswa.2021.115020

Mayhew, D. R. (1974). *Congress: The electoral connection*. Yale university press.

Mayhew, D. R. (1991). *Divided we govern*. Yale University New Haven.

McGrath, R. J. (2013). Congressional Oversight Hearings and Policy Control: Congressional Oversight. *Legislative Studies Quarterly*, *38*(3), 349–376. https://doi.org/10.1111/lsq.12018

Mettler, S., & Lieberman, R. C. (2020). *Four threats: The recurring crises of american democracy*. St. Martin's Press.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space*. Google Inc.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). *Distributed representations of words and phrases and their compositionality*. 3111–3119.

Mikolov, T., Yih, W., & Zweig, G. (2013). *Linguistic regularities in continuous space word representations*. 746–751.

Molina, M., & Garip, F. (2019). Machine learning for sociology. *Annual Review of Sociology*, *45*, 2745.

Monroe, B. L. (2013). The five vs of big data political science introduction to the virtual issue on big data in political science political analysis. *Political Analysis*, *21*(V5), 19.

Monroe, B. L., Pan, J., Roberts, M. E., Sen, M., & Sinclair, B. (2015). No! Formal theory, causal inference, and big data are not contradictory trends in political science. *PS: Political Science & Politics*, *48*(1), 7174.

Montgomery, J. M., & Olivella, S. (2018). Tree-based models for political science data. *American Journal of Political Science*, *62*(3), 729744.

Muchlinski, D., Siroky, D., He, J., & Kocher, M. (2016). Comparing random forest with logistic regression for predicting class-imbalanced civil war onset data. *Political Analysis*, *24*(1), 87103.

Mullainathan, S., & Spiess, J. (2017). Machine learning: An applied econometric approach. *Journal of Economic Perspectives*, *31*(2), 87106.

Nelson, T. E. (2011). *Issue framing* (G. C. Edwards, L. R. Jacobs, & R. Y. Shapiro, Eds.; p. 189203).

Nyrup, J., & Bramwell, S. (2020). Who governs? A new global dataset on members of cabinets. *American Political Science Review*, *114*(4), 13661374. https://doi.org/10.1017/S0003055420000490

Ornstein, N. J., Mann, T. E., & Malbin, M. J. (2009). *Vital statistics on congress 2008*. Brookings Institution Press.

Patterson, S. C., & Mughan, A. (2001). Fundamentals of institutional design: The functions and powers of parliamentary second chambers. *Journal of Legislative Studies*, *7*(1), 3960.

Patty, J. W., & Penn, E. M. (2015). Analyzing big data: Social choice and measurement. *PS: Political Science & Politics*, *48*(1), 95101.

Perlich, C., Dalessandro, B., Raeder, T., Stitelman, O., & Provost, F. (2014). Machine learning for targeted display advertising: Transfer learning in action. *Machine Learning*, *95*(1), 103127.

Peterson, A., & Spirling, A. (2018). Classification accuracy as a substantive quantity of interest: Measuring polarization in westminster systems. *Political Analysis*, *26*(1), 120128.

Pierson, P., & Skocpol, T. (Eds.). (2007). *The transformation of american politics: Activist government and the rise of conservatism*. Princeton University Press.

Polsby, N. W., Gallaher, M., & Rundquist, B. S. (1969). The growth of the seniority system in the US house of representatives. *American Political Science Review*, *63*(3), 787807.

Pool, I. de S. (1983). Tracking the flow of information. *Science*, *221*(4611), 609613.

Poole, K. T. (2005). *Spatial models of parliamentary voting*. Cambridge University Press.

Poole, K. T., & Rosenthal, H. (2017). *Ideology & congress: A political economic history of roll call voting*. Routledge.

Purpura, S., & Hillard, D. (2006). *Automated classification of congressional legislation*. 219225.

Quinn, K. M., Monroe, B. L., Colaresi, M., Crespin, M. H., & Radev, D. R. (2010). How to analyze political attention with minimal assumptions and costs. *American Journal of Political Science*, *54*(1), 209–228.

Raschka, S. (2015). *Python machine learning*. Packt publishing ltd.

Rice, D. R., & Zorn, C. (2021). Corpus-based dictionaries for sentiment analysis of specialized vocabularies. *Political Science Research and Methods*, *9*(1), 2035.

Richardson, H. C. (2009). *The greatest nation of the earth: Republican economic policies during the civil war* (Vol. 126). Harvard University Press.

Riker, W. H. (1992). The justification of bicameralism. *International Political Science Review*, *13*(1), 101116.

Roberts, J. M., & Smith, S. S. (2003). Procedural contexts, party strategy, and conditional party voting in the US house of representatives, 19712000. *American Journal of Political Science*, *47*(2), 305317.

Rohde, D. W. (1991). *Parties and leaders in the postreform house*. University of Chicago Press.

Romasco, A. U. (1983). *The politics of recovery: Roosevelt's new deal*. Oxford University Press.

Rudin, C. (2015). Can machine learning be useful for social science. *The Cities: An Essay Collection from the Decent City Initiative*, *9*, 8690.

Russell, A. (2021). *Tweeting is leading : how senators communicate and represent in the age of Twitter*. Oxford University Press.

Russell, M. (2001). What are second chambers for? *Parliamentary Affairs*, *54*(3), 442458.

Salganik, M. J. (2019). *Bit by bit: Social research in the digital age*. Princeton University Press.

Salisbury, R. H., & Shepsle, K. A. (1981). Congressional staff turnover and the ties-that-bind. *American Political Science Review*, *75*(2), 381396.

Schaffer, C. (1993). Selecting a classification method by cross-validation. *Machine Learning*, *13*(1), 135143.

Schaffner, B. F. (2018). *Party polarization* (G. C. Edwards, F. E. Lee, & E. Schickler, Eds.). Oxford University Press.

Schattschneider, E. E. (1960). *The semisovereign people: A realists view of democracy in america*. The Drayden Press.

Schickler, E. (2001). *Disjointed pluralism: Institutional innovation and the development of the u.s. congress*. Princeton University Press.

Schickler, E., & Bloch Rubin, R. (2018). *Congress and american political development* (R. Valelly, S. Mettler, & R. Lieberman, Eds.). Oxford University Press.

Schiff, S. H., & Smith, S. S. (1983). Generational change and the allocation of staff in the US congress. *Legislative Studies Quarterly*, 457467.

Seb, M., & Kacsuk, Z. (2021). The multiclass classification of newspaper articles with machine learning: The hybrid binary snowball approach. *Political Analysis*, *29*(2), 236249.

Shapley, L. (1953). *A value for n-person games: Vol. II* (H. W. Kuhn & Tucker, Albert W., Eds.; p. 307317). Princeton University Press.

Sheingate, A. D. (2006). Structure and opportunity: Committee jurisdiction and issue attention in congress. *American Journal of Political Science*, *50*(4), 844859.

Shmueli, G. (2010). To explain or to predict? *Statistical Science*, *25*(3), 289310.

Silbey, J. H. (2010). *American political parties: History, voters, critical elections, and party systems* (S. Maisel, J. M. Berry, & G. C. Edwards, Eds.). Oxford University Press.

Sinclair, B. (1986). The role of committees in agenda setting in the US congress. *Legislative Studies Quarterly*, 3545.

Slapin, J. B., & Proksch, S.-O. (2014). *Words as data: Content analysis in legislative studies*. Oxford University Press.

Smith, M., & Alvarez, F. (2021). Identifying mortality factors from machine learning using shapley valuesa case of COVID19. *Expert Systems with Applications*, *176*, 114832.

Smith, S. S. (1986). The central concepts in fenno's committee studies. *Legislative Studies Quarterly*, 518.

Soroka, S. N., & Wlezien, C. (2022). *Information and democracy*. Cambridge University Press.

Stewart, B. M., & Zhukov, Y. M. (2009). Use of force and civilmilitary relations in russia: An automated content analysis. *Small Wars & Insurgencies*, *20*(2), 319343.

Stimson, J. A., & Carmines, E. G. (1989). *Issue evolution: Race and the transformation of american politics*. Princeton University Press.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, *36*(2), 111133.

Strahan, R. (1988). Agenda change and committee politics in the postreform house. *Legislative Studies Quarterly*, 177197.

Sundquist, J. L. (1983). *Dynamics of the party system: Alignment and realignment of political parties in the united states*. Brookings Institution Press.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Theocharis, Y., Barberá, P., Fazekas, Z., Popa, S. A., & Parnet, O. (2016). A bad workman blames his tweets: The consequences of citizens' uncivil twitter use when interacting with party candidates. *Journal of Communication*, *66*(6), 10071031.

Theriault, S. M. (2008). *Party polarization in congress*. Cambridge University Press.

Thurber, J. A., & Yoshinaka, A. (2015). *American gridlock: The sources, character, and impact of political polarization*. Cambridge University Press.

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*(2), 411423.

Tsebelis, G., & Money, J. (1997). *Bicameralism*. Cambridge University Press.

Tversky, A., & Kahneman, D. (1973). Availability: A heuristic for judging frequency and probability. *Cognitive Psychology*, *5*(2), 207232.

Verberne, S., D'hondt, E., Van den Bosch, A., & Marx, M. (2014). Automatic thematic classification of election manifestos. *Information Processing & Management*, *50*(4), 554567.

Wallach, H. (2016). Computational social science. *Computational Social Science*, *307*.

Welch, R. E. (1988). *The presidencies of grover cleveland*. University of Kansas.

Wildavsky, A. B. (1986). *Budgeting: A comparative theory of the budgeting process*. Transaction Publishers.

Wilkerson, J., & Casas, A. (2017). Large-scale computerized text analysis in political science: Opportunities and challenges. *Annual Review of Political Science*, *20*, 529544.

Wolfe, P. M. (1972). *Linguistic change and the great vowel shift in english*. University of California Press.

Yee, O. S., Sagadevan, S., & Malim, N. H. A. H. (2018). Credit card fraud detection using machine learning as data mining technique. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, *10*(1-4), 2327.

Ying, L., Montgomery, J. M., & Stewart, B. M. (2021). Topics, Concepts, and Measurement: A Crowdsourced Procedure for Validating Topics as Measures. *Political Analysis*, *FirstView*.

Yu, B., Kaufmann, S., & Diermeier, D. (2008). Classifying party affiliation from political speech. *Journal of Information Technology & Politics*, *5*(1), 3348.

Zelizer, J. E. (2006). *On capitol hill: The struggle to reform congress and its consequences, 1948-2000*. Cambridge University Press.

Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *3*(1), 1130.