The Art and Artifice of Machine Learning Use in Political Research *

Laura M. Moses & Janet M. Box-Steffensmeier

September 30, 2020

Abstract

Machine learning is becoming increasingly prevalent in political science research. It has the ability to improve the accuracy of outcomes, refine measurements of complex processes, address non-linearities in data and introduce new kinds of data are all advantages of machine learning. Our work is intended to provide a clearer understanding of the growing number of applications that currently exist and encourage greater use by emphasizing how to use these tools and examining the pitfalls. This article provides a practical guide to machine learning beyond the textbook treatment and illustrates how these methods can advance political science research along the way, including a fuller example of how machine learning can help understand voter turnout through an application of these methods with election survey data.

*Janet M. Box-Steffensmeier is the Vernal Riffe Professor of Political Science & Sociology (Courtesy), Ohio State University Distinguished Scholar, Director of the Program in Statistics & Methodology (box-steffensmeier.1@osu.edu). Laura Moses is a Ph.D. candidate (moses.96@osu.edu). Ohio State University, Department of Political Science 2140 Derby Hall, 154 N. Oval Mall, Columbus, OH 43210

Machine learning (ML), which sits at the intersection of statistics and computing is revolutionizing parts of political science. Instead of manually constructing computational systems to learn something from data, ML systems *learn* programs from data, making them a flexible solution for analyzing complex, large data increasingly used in research. These modeling approaches are changing the way political science can address fundamental questions of causation and allow researchers to leverage novel types of data to study social phenomena (Grimmer, 2015). ML tools already improve the measurement of critical and complex concepts like ideology or legislative effectiveness (Abi-Hassan et al., 2019), (Casas, Denny and Wilkerson, 2020). Applications of ML are already widely used for text analysis (Monroe, Colaresi and Quinn (2008), Grimmer and Stewart (2013), Roberts et al. (2014)). ML is also used for classification of events (Jones and Lupu, 2018) and video or image data analysis (Dietrich (2020), Casas and Williams (2019) Dietrich et al. (2019)). While there are excellent texts on the statistical underpinnings of ML (e.g., Bishop (2006) and Murphy (2012)). Yet, there are important aspects of using ML to consider beyond the statistical form of a model for ML that need to be considered for these tools to be successfully developed and implemented in political science that are not covered in textbooks.[1]

This article presents these considerations of ML needed to leverage implement ML models successfully. This includes artifices to be wary of, important factors for selecting an ML method and specific concerns for political science research. In the first section, we discuss the composition of ML learners, provide an overview of methods in the literature and the pitfalls models and learning objectives can encounter. Then, we discuss the importance of prediction and some of the challenges ML can create. Moreover, our objective is to introduce political science to practical lessons for using ML beyond understanding the statistical form of any particular learner. We argue that, in certain situations, ML learners are an appropriate standard choice for analysis. In sections five and six, we conduct analysis with ML with synthetic data and data from the CCES survey that demonstrate both some major

[1]ML can be done with software packages in common programming languages like python and R. In R, a number of packages like `mlr` (Bischl et al., 2016), `caret` (Kuhn, 2008) are a useful place to begin.

advantages and their limitations. First, we conduct analysis with simulated data that illustrates the conditions when ML is appropriate and valuable. We then apply the methods to the CCES data to generate estimates of voter turnout. This replicates and extends analysis in Kim, Alvarez and Ramirez (2020) to analyze a large collection of survey data with correlated feature sets. We conclude with general words of caution about fairness, bias and data considerations for ML use in political science.

1 The Big Picture of Building Learners

Determining which learning algorithm to use out of the thousands available can be a daunting challenge for social scientists new to ML. Unpacking how ML models are constructed can help situate what may be of importance for any given application. In short, learning are built with two things: data and some desired output. The most prevalent ML approaches are supervised and unsupervised.

In unsupervised learning, the learner finds patterns in the data, without any prior knowledge of the dependent variable. The goal of unsupervised models is to find the notable structures in the data by modeling density estimations using only independent variables. Not having outcomes defined beforehand changes the learning objective. Learning comes from assumptions about the structural, probabilistic properties, or algebraic properties of the data (Jordan and Mitchell (2015), Shmueli (2010)). Unsupervised ML allows researchers to make estimates for missing data, finds relevant patterns that allow for dimensionality reduction or latent variable discovery, which lead to finding patterns and new causal mechanisms.

In supervised ML the outcome values are known. Supervised learning can be used for classification or regression. Classifiers take input features and output discrete values, or classes. Think about classifying individual ideology as "liberal" or "not liberal" from survey data, with inputs $x = \{x_1, ...x_9\}$ where x_9 is the ninth survey response. A *learner* receives training set examples (x_i, y_i) where x_i is an observed respondent and y_i is the corresponding outcome value, ideology and then outputs a classifier. The true test for the learned classifier

is its ability to generate accurate predictions $\hat{y}$ for data that the model has not seen. Ordinarily, we think of the outcome y as the *dependent variable*. In this context we introduce the terminology used in the ML literature and refer to y as the *target values*. When y is categorical, these values are sometimes called the *class labels*. Supervised learners are the most ubiquitous and will be the emphasis of our discussion in this paper. However, the strategies, trade-offs, and issues discussed in this paper apply to all types of ML. Table 1 provides a brief description of supervised and unsupervised ML along with some relevant examples.[2]

Table 1: Supervised & Unsupervised ML Learners

Type	**Data Transformation**	**Interpretation**	**Learners**
Supervised	Predicts outcomes or classes on new data by estimating a model and applying the model to unseen, new data to generate predictions	Requires the target variable values be known	Naive Bayes Nearest Neighbors Support Vector Machines CART Neural Networks Regularized Regression
Unsupervised	Generates target outcomes or classes directly from input features	Does not require known target values. The researcher determines what the target values mean in context	Topic Models K-Means Expectation Maximization Support Vector Machines Principal Component Analysis Neural Networks

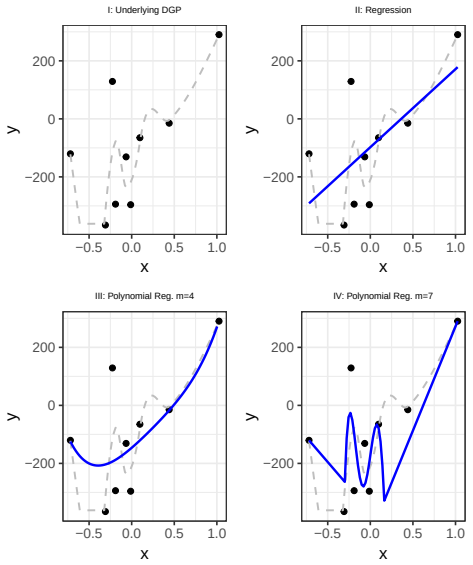
All ML models have three components: representation, evaluation, and optimization. Representation refers to the landscape of a particular modeling approach. Most learners are defined by their representation. For example, support vector machines and neural networks are different types of representations. Different representations will encode and express

[2]See supplementary materials for statistical ML model discussion overview.

models differently. A model needs to be understood by the computer or it cannot be learned. That is, even the most powerful learners can produce poor results if the representation used in not appropriate for the data type (Domingos, 2012). The same ways in which ordinal least squares regression is a poor model for data with a binary outcome, not all ML models are well suited to all data analysis tasks.

The representation also determines the hypothesis space and the possible set of models that can be learned given the type of model being used. The way information is represented affects how well a model can describe the data generation process. How a learner is represented determines the set of candidate models that can be considered. Consider the data

Figure 1: Possible Representations of Data



Panel 1 illustrates the underlying data generating process. Subsequent panels show possible representations of the data.

in Figure 1. The first panel shows the true process underlying the data, represented by the dashed line. Subsequent panels illustrate that there are many different ways of representing the data with a polynomial regression lines. Modeling this data with a linear model using parameters is not expressive enough; the more expressive polynomial illustrated in the third panel captures more aspects of the data generating process, but not perfectly. Adding more

parameters can help, but too many parameters can make the representation too expressive. The model in panel four fits all the data points but does not representing the underlying process well. The representation in the fourth panel will not generalize and will fail to make accurate predictions because it is too expressive. Just as in statistical modeling, the most expressive and inclusive model, may not be the best choice to fit the data. How does a researcher go about selecting the best one? This question leads us to the second building block, the objective function. This function helps the ML learner differentiate among good candidate algorithms and bad ones.

The objective function is used internally by a learner to distinguish good models from bad ones. Objective functions are generally defined as loss functions, which define the penalty for making errors between the true target value y and the predicted target value, $\hat{y}$. The objective function depends on the representation of how the target is modeled. Think again about linear regression. Here, the objective is to find the best prediction of the dependent variable, which is determined by the minimum mean squared error between the prediction $\hat{y}$ and the truth y or by the maximum likelihood of a model given the observed data. In ML there are many choices for evaluating the objective or the loss, beyond minimizing the mean squared error. Frequently used measurements are the predictive accuracy, information gained and relative entropy[3], and minimization of margins. An objective function is a way of scoring each of the possible algorithms the learner is considering; how objectives are defined is dependent on the representation and what is desired as an outcome.

The final component of any ML model is optimization or, how learners search for the best performing objective in the space of represented models. Optimization choices determine the efficiency of the learner and its ability to find the maxima or minima of the objective function. This is essential to determine the outputs produced by the model. In maximum likelihood estimation, the objective is to find the maximizing value of each parameter. Optimization is how to find those values. Many ML tools predetermine how optimization is done, but it is

[3]Relative entropy is also known as the Kullback–Leibler divergence. This measures how one probability distribution is different from a second, reference probability distribution.

important to acknowledge how the search for the best values is conducted as it impacts the model that is produced. Not all combinations of possible representations, evaluations, and optimization are reasonable. For instance, combinatorial optimization techniques may not be the most efficient way to evaluate continuous data and some types of objective functions, like maximizing the likelihood, are not possible to use with ML models that make no distributional assumptions about the data(Domingos, 2012). Just as no single modeling choice or specific algorithm is optimal for all learning tasks (Wolpert and Macready, 1997), some ML methods are better for certain types of tasks than others. In many cases, models with different represntations are used together or in parallel to do analysis. In subsequent sections, it will become clear that representation decisions may not be paramount in an ML project. The next section touch on some of the key issues to consider when selecting a learner.

2 Prediction is Key

Prediction is central in ML. Unlike applied statistical models, which aim to explicate how the features relate to the outcome, the primary goal of ML is to *generalize* models can accurately predict outcomes for data that was not used to fit the model. A traditional statistical model is concerned with inferring the relationship between the inputs and the dependent variable. These models and ML models are both rooted in the practice of answering questions by estimating a model from the data to make statements about the outcome of interest. For example, a linear regression model,$f(x) = x^T\beta$, uses data to learn an assumed linear model $f(x)$, by estimating the coefficients we select to use in our model. These parameters tell us something about how the features $\mathbf{x}$ relate the outcome y. In ML both the model and the parameters are learned.

Like generative modeling, ML requires thinking statistically and thinking about knowledge in a statistical form. "Congress is 535 men and women" is knowledge. Statistical knowledge is: all Members of Congress are men or women, but only 24% are women. The same way a generalized linear model uses data to learn a model $f(x)$, where the model parameters tell us something about how the features $\mathbf{x}$ affect the outcome y, ML can map $\mathbf{x}$

to the outcome y, with a function or a process. Linear regression can be used as ML, the difference is, in ML the aim is to find models that make accurate predictions, de-emphasizing how well coefficients may explain the sample. Though, if features related to the outcome often imply associational relationships.[4] Predictive modeling requires a framework that allows for a model to be learned on one set of the data known as the *training set*, which is used to fit the model, and the *test set*, which is withheld observations not used to learn the model. Test set data are used to determine the performance of the model, often measured by its ability to predict results out-of-sample and make generalized claims about the data generating process.

ML relies on these two connected principles, prediction and generalization. Generalization is the learner's capacity to adapt to previously unseen data and create accurate predictions. ML makes better estimates when models are able to generalize from the data or extrapolate the patterns that are necessary to make accurate predictions. Any model that is a good representation of the data can create accurate predictions on a training set easily, the learner just has to memorize the training examples and assign model weights.[5] Even when reserving data in a *test set*, artifice can occur in results. Once a model is created and tested on test data, if that test data is then used to tune or change model parameters too much, it can lead to biased results. Since the data has been used to fit the model in some capactity, it is no longer purely out-of-sample. To avoid this, cross-validation should be used during training. Cross-validation randomly partitions data into subsets, holding out a different subset while training on the rest, then testing each learned classifier on the examples it did not see, and averaging the results. In essence, cross-validation selects test and training subsets with replacement from the training data. This can also help avoid overfitting.

The predictive properties of ML allow us to create better models by validating the outcomes and generalizing from examples. ML facilitates the verification of theoretical causal

[4]Cranmer and Desmarais (2017) prove an in-depth discussion on predictive modeling in political science.

[5]A common ML pitfall for beginners is to not split the data and test predictive accuracy on the training data, used to create an ml model, and have the perception of accurately predicted outcomes.

claims using the principle of generalization. The principles of generalization and accuracy that ML tools are predicated on can be tailored to address a wide range of political science research questions. The flexible ways of representing data that ML facilitates can enable political science to solve problems of measurement, identify underlying mechanisms, map interdependencies, and process large quantities of highly dimensional data.

With prediction at the center, ML models rely on the training error to inform the researcher if the model is a good fit for the data. Training error is a stand-in for the test error until the model is refined enough to try on new, unseen test data. Unlike in general linear modeling, where the objective is to optimize the likelihood for the parameters of the function, in ML there is not a given function to optimize. Prediction and generalization have another implication, which is that the knowledge about how to represent the data is necessary. How each survey response is coded or the ways different relationships in social networks, images or text is represented determines what can be learned from the data.

Accurate learners can produce poor predictions when the data are not appropriately represented. An important criterion for choosing the right representation is considering what kinds of information are easily expressed in the representation, and what is known about the data. Consider trying to predict ideology using a survey containing 75 yes and no questions with a training set containing 100,000 respondents. Even with this large dataset, with 2^{75} possible inputs, you have observed effectively *zero* ($\approx 2^{-51.7}$) percent of the hypothesis space, hence generalization is key.[6] Creating an accurate model that can predict beyond the training set can quickly become guess work. Prior knowledge or assumptions can help determine an accurate model. If there is not a lot of certainty about the probabilistic dependencies in what determines ideology, but we do know that some conditions lead to more liberal or conservative views, a tree model or finding a separating boundary between liberals and conservatives may be a better approach than a graphical model or one that makes many

[6]There are 2^{75} possible inputs and $10^5 \approx 2^{16.7}$ accounted for in the training set, assuming each respondent answered in a unique way. As a proportion of the input space this is $\frac{2^{16.7}}{2^{75}} = 2^{-58.3}$.

distributional assumptions.

Unfortunately, it is never the case that a single model will be optimal across all ML tasks and there is not a universally optimal method for converting data in to something discernible for ML. For example, when using text or image data, specific choices about vectorizing or converting the data into numerical representation, or how to represent categorical or continuous features may need to be made. Even when there is limited information about the data in a political context, some general assumptions like the smoothness of the approximate density function, or that the complexity of the components is limited or independent, can often be enough to yield good performance of a model. Learners are ultimately inductively taking in a small amount of data and outputting a larger amount of information. This inductive process emphasizes the importance of how data is represented. Transforming the data to meet assumptions for statistical models is standard practice, and is done for regression and other generalized linear models. With ML there are more approaches to data representation and hand-tuning to transform the data (Bishop, 2006).

3 Pitfalls: Overfitting & Dimensionality

To successfully use ML, practitioners need to understand the pitfalls of model implementation. The two most common challenges are overfitting and understanding dimensionality relative to the model.

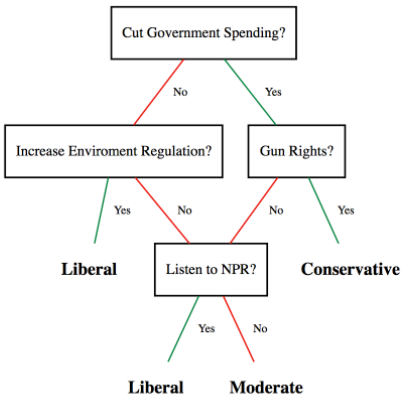
Overfitting

When learners latch on to patterns in the data that are not reflective of the underlying data generating process, the model has overfit the data. Overfitting is a central issue for all learners, especially for weak learners like k-nearest neighbors or decision trees. This does not mean complex models are immune to overfitting problems.

Consider a simple decision tree example in Figure 2. A decision tree acts as a flow chart for the data to predict the targets.[7] Decision trees decide which features in the data are best at predicting the target or target class by starting with features that provide more

[7]See supplementary materials for an in-depth review of decision trees.

Figure 2: Decision Tree for Predicting Vote Ideology



information to create initial partitions in the data. Decision trees use a split function to decide what is the best feature and the best value for that feature. No other modeling choices are made once we determine how the splits of the data are made. The decision tree recursively splits the data until targets are properly sorted.

Overfitting happens when data includes many, weakly informative features unneeded for prediction. When a model includes extraneous or uniformative features, there is more variance in the data. One way to combat this is to remove features that only weakly predict an outcome, resulting in a parsimonious model that will better generalize and be less prone to overfitting. Decision trees are often "pruned" to avoid overfitting by calculating if further splits of the data are necessary. This is done by removing all possible sub-trees and re-evaluating the classification accuracy after the removal of sub-trees. By doing this, parts of the model that provide less information for accurate predictions are removed.

Overfitting is worsened by noise in the data, or when observations have outcomes that are difficult to predict obscure the signature pattern in our data we are searching for. If we have an individual in our four-question survey example who is for gun rights and tax cuts, but still identified as a liberal, our tree would have a difficult time separating that individual from conservatives. Unless our questions fully accounted for all the reasons that an individual identified with a certain party, we will likely have noise in the data that leads

to misclassifications or errors.

How can we tell if a learner is overfitting? When models fail to generalize well because they latch on to the variance in the data, they overfit. The errors made in predictions can be decomposed into bias errors that result from model assumptions, variance errors resulting from sensitivity to small changes in the training data, and random noise.[8] Evaluating the training set errors is an important step in understanding a learner's performance. However, the absolute test for determining a learner's fit to the data is to test the learner on the test set. A learner that can generalize beyond the training set accurately is how we determine the generalizability of the learner's explanation of a data generating process. Predicting outcomes with high accuracy on data that has not been seen by the learner is a high standard for performance and will indicate how well the learner generalizes beyond the sample used to learn the model. If the patterns the learner hypothesized about being true from the training set hold on the test set, you can be confident that these patterns exist. When trying to optimize a model to represent social phenomena, we do not ever actually see the function we are aiming to model. Using training errors and test data predictive accuracy to determine how well our model works, sheds light on the important social science questions using prediction and explanation together. Pragmatically, this illustrates why it is crucial to divide data into training and test sets.

The Curse of Dimensionality

In addition to overfitting, the loss of intuition and interpretation for models as dimensionality increases is a concern.

Many ML tools that work well in low dimensions become intractable when the inputs are in high dimensions.

Further, the data's dimensions increase, the data occupies less of the hypothesis space. Even if all the features are informative, in high dimensions, the similarity difficult to define.

[8]We can also think of overfitting in terms of bias and variance in the errors. When models create predictions that are far from the true values, models suffer from bias and "underfit". The aim is to balance between these two competing forces to minimize errors overall.

This can cause a model to be intractable and lead to the inaccurate predictions, because the training set covers a smaller portion of the total hypothesis space.

This is referred to as the "curse of dimensionality." The dimensionality increases exponentially as data contains more features, more information or more observations. Increased dimensionality of data makes ML tools appealing; but at the same time, the unintuitive nature of increased dimensions poses a challenge. High-dimensionality can also cause models to perform poorly because of the increased noise introduced to the model. This can happen even when all the features are contributing to prediction since the observations have increased similarity, making them closer to one another along different dimensions of the data.

All ML models are affected by dimensionality to some extent. Weak learners, or learners with few assumptions or hyperparameters, are particularly sensitive. For example, nearest neighbor learners have weak assumptions. The lack of assumptions can be useful for times when there are many unknown patterns in the data. Nearest neighbor models find the most similar observation in the entire dataset for any new observation. These models do not weight by the features, allowing the data to "speak of itself" but also makes them susceptible to overfitting. Models with weak assumptions like nearest neighbors, can fail to determine which observations are most alike in high-dimensional settings.[9]

Consider again the example of determining ideology through a series of yes and no questions. More questions on a survey provide information, but also makes the hypothesis space larger. As the number of dimensions increase, the number of training examples required to locate the boundary between liberals and conservatives also goes up exponentially. Thinking about separating the classes in two dimensions is intuitive; we can think about drawing a decision boundary, or line to separate the classes along the x and y axes. In twenty or more dimensions, understanding where the decision boundary between liberals and consevatives

[9]There is a weighted version of k-nearest neighbor that can be used to weight some features more heavily than others. Weighting can also mitigate prediction error for noisy data or data with substantial missing values.

occurs is less intuitive. If there are only twenty questions, each with two possible answers, there are one million possible examples. Each additional question added to the survey would increase the complexity. Further, if new questions only weakly inform ideology, then the added information may create irrelevant dimensions in the data. This noise can make prediction even more challenging. Luckily, in many ML applications, examples are not uniformly distributed throughout the instance space but are concentrated in a lower-dimensional area. A strategy for managing dimensionality is to reduce the number of features to only include the important values.

4 Getting Useful Learners

All learners have hyperparameters that determine the form of the model. For example, choosing k in the number of neighbors to use, or how many trees to learn in a random fores are set by assumptions and determined before estimating the learner's parameters. Unlike model parameters, hyperparameters are set manually, determined theoretically or optimized through a search procedure. Modest changes to hyperparameters values can change the outcomes and the accuracy but also influence the model's parameter values and the final model form. Finding or selecting optimal hyperparameters values for a given data set requires creating a new version of the model each time. This process is sometimes called "tuning" and can lead to better predictions or measurements. If adjustments to hyperparameters are done on the test set or the entire data set after the model's parameters are estimated, adjustments of these parameters can result in false predictive accuracy and not generalize well to cases beyond the initial analysis. In cases where the dataset is not large enough to divide, cross-validation can be used during model training. Selecting hyperarameters and assessing performance on subsets generated through cross-validation ensure that the learned model generalizes well to out-of-sample data.

Using many ML learners or combinations of learners is another approach to getting useful ML models. Combining learners can minimize overfitting and maintain generalization. The performance of many different types of learners can tell us something new about the data or

political process of interest. Learning with combinations of different models is called *ensemble learning*. The composition of an ensemble can contain several learners of the same type, like using three random forests, or different ones, like a random forest, support vector machine (SVM) and naive bayes model. The simplest approach is a homogeneous ensemble method called *bagging*. Bagging is bootstrapped aggregation of the model, where random variation is introduced to the training set by re-sampling. The model is applied to each re-sampled training set. Each re-sampled model makes predictions for the target values. We can think of these as votes for each observations' predictive target value. The target predictions with the most votes for any individual observation are reported as the best prediction. Bagging can reduce variance without increasing the bias of a model. The difference between bagging and cross-validation is that bagging averages predictions of an ensemble and cross-validation evaluates a number of potential models assuming that they are equivalent.

Another approach is *boosting*, which provides weights to training examples that are incorrectly predicted, thus increasing their importance in the next iteration. These predictions are combined through weighted sums or votes to determine the best prediction. In boosting, the base learner is trained in sequence on weighted versions of the data, where as bagging uses random re-samples of the data. This allows boosting to leverage the dependencies between the base learners. Another way to think about these techniques is, bagging decreases variance, while boosting decreases bias.

Bagging and boosting methods can be used or combined with ensembles. This is a common approach for highly dimensional data like text. Grimmer, Westwood and Messing (2015) use a heterogeneous ensemble composed of an elastic net, random forest and SVM to classify press releases that contain credit claiming to constituents by members of Congress. Ensembles of different learners are created when any individual model is accurate at some type of prediction with the data, but the diversity of models included creates better accuracy across different dimensions. While none of these strategies resolve the balance between bias and variance in the errors of a learner, they ensure that the model is a generalization of data

generating process beyond the samples collected for analysis.

5 Illustration & Interpretation

In this section, we walk though two examples to showcase ML. First, we evaluate the performance of common learners: Classification and Regression trees (CART), Random Forest (RF), Multi-layer Perceptron Neural Network (MLP), Naive Bayes (NB), an SVM, and AdaBoost, and compare the results.[10] Second, we demonstrate what ML can tell us about the 2016 presidential election. Assessing the numerous theories of voter turnout in major presidential elections requires using many features resulting in complex data. This example illustrates how ML can be advantageous to address often studied questions, like voter turnout. By uncovering the features important for predicting turnout without knowing which features are most meaningful, the ML approach can resolve debates and long-standing questions in political science.

Synthetic Data Example

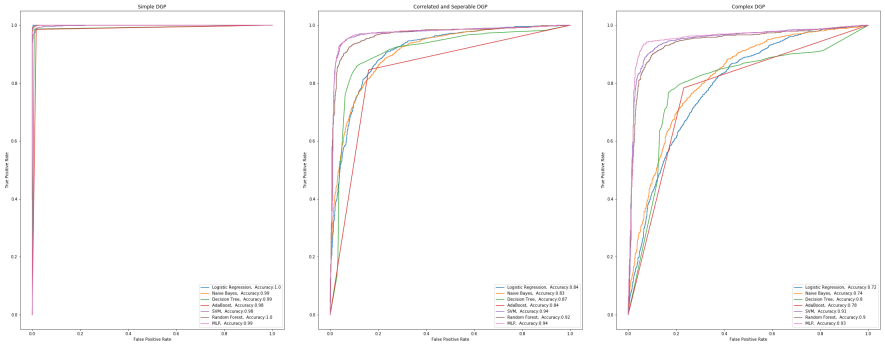
We begin by illustrating ML on three synthetic datasets. The simplest dataset is linearly separable with ten potential features. The simple dataset includes a separable set of target values, with little noise and no redundancy or correlation between features that are important for classification.

The complex datasets have 25 symmetric, and asymmetric features. These datasets include strongly correlated, weakly correlated and independent features.[11] These data introduce more variation in the target values as well as noise and redundancy in the feature values. We split the data into test and training sets. Using cross-validation, we fit the different models to the training data. To evaluate the relative accuracy of each ML model, we calculate the average out-of-sample accuracy from fitting each model using a cross-validated split of training dataset and then evaluating the predictive accuracy on the test set. These seven types of models are fit to the three sythetic datasets described above.

[10] See supplementary materials for model details and specifications.

[11] Additional details for data simulation are available in the supplementary materials.

Figure 3: ROC Curve Model Comparisons



Supervised learners are often evaluated by plotting Receiver Operating Characteristic (ROC) curves or Precision-Recall (PR) Curves. Both tools are restricted to a binary classification predictive modeling problem but are often modified to be useful in multiple class scenarios. ROC curves characterize how well a model can trade off its ability to identify positive class members (or the true positive rate) with its ability to not classify negative class members as positive (the false positive rate). PR curves show the trade-off in a model's ability to identify all positive class members (recall) with its ability to predict only positive class members (precision). PR curves are favored over ROC curves for imbalanced datasets. Since we are evaluating a binary outcome, voting or not voting, on relatively balanced data, we use an ROC curve. Figure 3 illustrates the ROC curves for the out-of-sample prediction performance of the seven different models. The closer a curve gets to the upper left corner of the graph, the more accurately the learner is classifying the test set. The left panel shows the results of the simple DGP set where a majority of the data informs the outcome and there is low variance in the distribution of target values. All models perform with high accuracy when the data are linearly separable. In this case, the assumptions for many different ML models are easily met. Models that are easy to interpret and implement, like logistic regression, do as well as the ML models. Across the other panels, as the data increase in complexity, more complex learners perform better and generate accurate predictions for the target values. In the complex cases, the random forest and neural network are more useful

than the decision tree or logistic regression. ML models perform well under a variety of data-generating processes, especially in complex data, which mirrors the data complexities of real-world political science data. Next, we consider how ML can support political science research using an illustrative example of voter turnout.

Predicting & Explaining Voter Turnout with ML

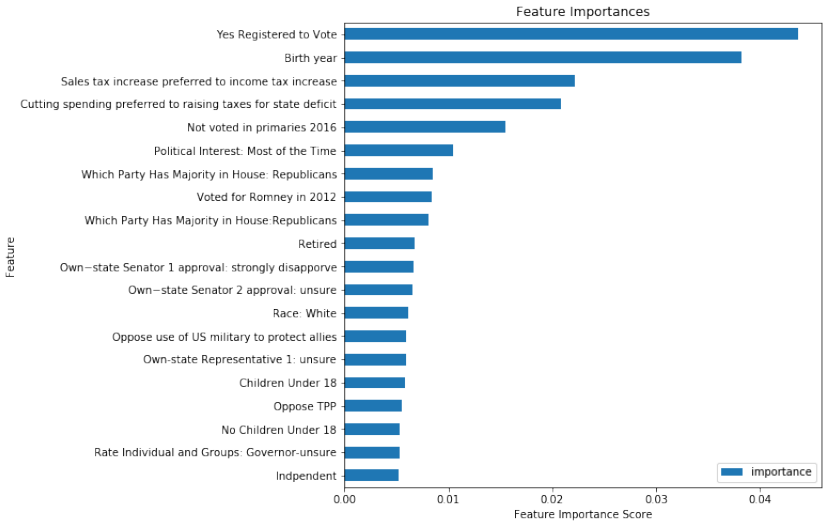
In order to make valid claims about political processes from ML results, researchers are faced with choices about including important features, and how to address competing theoretical claims with their analysis. Consider voting behavior in American politics. There is a broad literature that offers explanations as to why voters participate in national elections. How do we determine which factors were important for voter turnout in the 2016 elections? The role of campaigns, issue importance, institutions, and the individual may all matter, but what mattered most in this election cycle remains unclear. With the unique political actors participating in 2016, additional features beyond those with established theoretical importance could be important factors in predicting voter behavior in 2016. With such a large number of theories and the highly dimensional, correlated representation it yields, ML help develop an inclusive, comprehensive understanding of voting behavior in 2016.

We use the Cooperative Congressional Election Study (CCES) data from the 2016 general election. The CCES is a national stratified sample survey that validates respondents' voter behavior by matching voter files to their survey data. In 2016, there were 64,600 individuals surveyed. Our sample is limited to respondents who answered both the pre- and post-survey waves. We validated voters and define non-voters as both matched non-voters and non-matched respondents. This yields a total sample size of 43,871 respondents available for analysis. We use 97 variables from the survey to capture features important to voting behavior like age, race, education, income, political interests, and policy questions.[12] We one-hot encode the dataset, meaning that we transform categorical variables to binary variables,

[12]This follows a similar approach that Kim, Alvarez and Ramirez (2020) takes to address a similar question using Fuzzy Trees.

resulting in 449 variables. One-hot encoding is a process that ensures categorical responses to survey questions are not treated as ordinal variables by a model by binarizing. Binarizing categorical or ordinal variables in to integer representations of the information by making each categorical response a binary variable. This approach eliminates the ordering issues that may not be retained by some ML models, but adds many dimensions to the data.

Figure 4: Feature Importance, Top 20 Features for Voter Turnout



We fit a random forest model and a naive Bayes model since and explore the results of the more accurate random forest model. Since we are interested in interreting the model and the features in addition to generating accurate predictions about voter turnout behavior, we select ML tools that allow us to interpret model components. We assess what the model predictions tell us about voter turnout behavior. As with all ML models, the hyperparameters impact the final accuracy and results of a model. To address this, we optimize the hyperparameters parameters using grid search and cross validate.[13]

Figure 4 shows the importance plot for the top 20 features. The model identified which features are the most informative for predicting turnout. The importance of each feature is calculated using the sum of the information decrease across every tree of the forest every time that variable is chosen to split a node. The sum is divided by the number of trees in

[13]See Supplemental Materials for more details.

the forest to give an average of information gained from the feature. The scale of importance is irrelevant but the relative values between features are useful. The important features are consistent with research on voter turnout. Respondents who are not registered to vote are less likely to turnout of an election. The age of the respondent, interest in politics and past voting behavior are all relatively important in predicting turnout. Additionally, if a person did not vote in the primaries, which is highly correlated with not turning out for the 2016 general election.

In political science, ML models are most useful when the model is interpretable. Predicting voter turnout with these data can be done with SVMs or neural networks. A neural network trained on this data, is as accurate as the random forest but, has few explainable model components.

The outcomes from the random forest allow us to understand how features used for predictions. In addition to validating theory, the interpretable model allows erroneous predictors to be identified and better target the sources of errors in a model. Other types of ML models may generate more accurate predictions, but at the expense of explanation derived from the model. This does not make less interpretable ML models less useful. For instance, SVMs, which rely on defining margins between the target classes in the data can be useful for locating the source of misclassifications and provide new avenues of study. It does require the researcher to consider the importance of interpreting the model in the ML application. If the voter turnout outcome were to be used in subsequent analysis, complex and accurate models like neural networks can be valuable for generating accurate measurements. ML to be a useful tool for predictions and how predictions can be used as a starting point for understanding political processes or for confirming existing theories.

6 Hints and Tips on Data & Model Considerations

The goals of computer science, where much of ML was initially developed, emphasize the advancement of efficiency, accuracy, and generalizability, which are quite separate from the goals of social science (Wallach, 2018). So adopting these learners require additional

considerations. Interpretation of a learner allows for an explanation of decisions and choices made in identifying an accurate model. Doshi-Velez et al. (2017) offers three key qualities to consider when using ML: fairness, reliability, and trust, which are equally important for political science applications. ML emphasizes prediction and generalization, whereas social science emphasizes explanation. ML models for an explanation must be interpretable. Their structure needs to relate to the explanation of interest and be grounded in existing theory. Interpretable learners establish trust in the predictions and a better understanding of how a model behaves. Some learners are inherently easier to interpret than others. For example, the learned weights in a Naive Bayes model or features used in a decision tree can easily be interpreted. The interpretation of more complex models, like the many parameters generated by a Neural Network, cannot explain the relationship between the features and the targets. This "black box" quality to how models arrive at the accurate prediction leads to skepticism. To trust a model's predictions and to trust that the model is behaving appropriately, interpreting the reasons for the model predictions is critical. Unfortunately, it is model complexity the gives ML learners the capacity to make accurate predictions about complex processes. Local or global approximations can be a solution to interpreting some of the complexity of the original model. Local approximations make a simpler model from the original learner for a single example (Ribeiro, Singh and Guestrin, 2016). In neural networks, specific layer activations and weights can be directly examined for inputs of interest (Zhou et al. (2018) Kim et al. (2018)).

Benchmarks for interpretability in ML is an unresolved debate. Typically, interpretation is considered in the context of being able to quantify a proxy or illustrate a local example with the data. Political scientists should make sure that their models and evaluations match the type of model checking and interpretability evaluations they elect to use. ML models that further our understanding of political processes needs to be evaluated beyond generalization errors, variance, and bias to account for the context of the application. Even in the case of clustering, outside of pattern exploration, the ultimate conclusions should still reflect a

human-verified evaluation of the learner's findings.

Selecting learners that highlight explanations in social science is critical. Learners that allow for contrast, or are relational to compare outcome predictive values or the relationship of the inputs and outputs, are those that are the most useful for political science interpretation. Additionally, political science must find base rates for performance that make sense for the particular application. Some learners like CART models, Random Forests, Naive Bayes, regularized regressions, etc. are highly interpretable but may not perform as well as more complex ML models such as Neural Networks, ensembles, or Support Vector Machines. This balance between interpretability and performance is a definite challenge. For political science, learning how to apply interpretable models should be considered first.

Once a learner has been selected and tuned, the learner is source of knowledge about the underlying political or social process it models. Model interpretation can help the researcher understand how or why a model makes its predictions and to draw political insights from the knowledge captured in the outputs of the model. Interpretation of ML models can also ensure confidence in the model's predictive performance and a better understanding of when a model is failing.

Political science analysis with ML must also be cognizant of social bias. Studies using social or political data, which is about people's behavior, reactions, or institutions, must be diligent in understanding what is represented with the data and the social biases implicit in the data. These social biases impact analysis and these data bias concerns are not addressed with hyperparameters or modeling choices. Biases can be embedded in data because of implicit biases and discrimination evident in the social world. This bias also occurs when data is missing. Consider using ML to determine the opinions of non-voters. Assume analysis was principled and the result was a learner with a predictive accuracy rate of 95% for predicting the topics non-voters discuss. The learner is accurate in determining the non-voter opinions 95% of the time *for the data.* If white males over 50 are over-represented in the data and minority women are not well represented in the data, the learner may be accurate for the

majority of the data, but perform very poorly at predicting topics important to minority women, who were rarely observed in the data used to train the model. If the same learner had a 35% predictive accuracy for predicting the opinions of minority women, the accuracy of the model is called in to question.

Conducting error analysis to determine if misclassifications or incorrect predictions are truly stochastic or evidence of systemic patterns in the data are critical. Text documents, survey questions, experiments, etc. are generated by people, all of whom are susceptible to biases. For instance, the words "women" and "home" are closely associated while "man" is closer to "math" (De-Arteaga et al., 2019). This bias is not always readily apparent but has important consequences. When learners have seemingly accurate predictions, yet the errors are systematic there are important implications for misrepresentations of social processes or exclusion of particular group patterns and behaviors. Bias can also arise from the absence of missingness in data. This is exemplary of the other kind of bias that arises. This often occurs when the data set does not encompass all the possible variations of the outcome you hope to quantify. In this case, the bias arises from a lack of varied data in the training. This lack of data in social data sets has created some public backlash, like when Amazon had to pull their AI for recruitment when it yielded sexist recommendations, because of gender imbalances in the training set. How to systematically assess the construct validity and reliability of measurements, particularly in natural language processes, is an ongoing debate. Some techniques such as measurement modeling can uncover latent constructs that capture human bias in texts, these techniques allow researchers to adjust their models to account for theoretical bias (Alvarez-Melis et al., 2019).

Some types of error are more readily apparent and easier to diagnose. Detection of rare events like conflicts, or ensuring minority groups are represented in surveys can be ensured with techniques like up-sampling, down-sampling, weighting or pooling to create balance in the data across classes.[14] Biases can be prevented by maintaining an explicit representation

[14]In down-sampling, training datasets are constructed disproportionately on a low subset of the majority class examples, so that the over-represented classes have a similar number of observations relative to the

of uncertainty in the model. By maintaining the uncertainty of an estimate, the output of the model allows not only for substantive interpretation, but also for an opportunity to identify decisions that are contrary to the generalized case or demonstrate the strength of correct predictions with weak information as inputs. Understanding when learners misclassify or mispredict facilitates the discovery of new patterns and helps account for normative societal implications.

Data informs us of what representation is most appropriate or easily expressed. For example, if we are unsure of linearity or the complexity of interactions, an instance-based learner such as a decision tree or random forest may be appropriate.[15] Instance-based learners compare new data instances with data from the training set, which have been stored in memory to make predictions. In the case of smaller datasets or when using multiple sources of data, like matching survey responses on the economy and district level economic data, a generative model, like naive Bayes, may provide better performance than a discriminative alternative (Ng and Jordan, 2002). Just as there is no universally good model, there is no prescriptive application for different machine learners. The most useful representation of the data and learner choices is when the knowledge is easily expressed and the assumptions about the underlying processes can be explicitly stated and incorporated into the learner.

Conclusion: ML & Social Datasets

ML is cutting-edge and fast becoming a standard for quantitative analysis in scientific research across the social sciences. Indeed, ML has evolved into a sub-discipline unto itself, and like all disciplines, there are evolving principles guiding research and specialized sub-fields. This article expounds upon some of the most salient practical guidelines for use of ML in political science.

Interpretable learners can be a useful alternative to statistical modeling, while powerful predictive models can help refine measurements and discover new patterns in complex data.

other classes. In up-sampling, minority classes are re-sampled to create balance

[15](See: Montgomery and Olivella (2016) for a detailed discussion of tree-based models)

As these methods become increasingly prevalent, and for them to have maximum potential to positively move political science, practitioners and readers must understand not only the particular functionality of specific ML models, but also the implications of the data and modeling challenges that come with ML.

Learners have the flexibility to answer new questions and uncover new data sources. This expansion in political science has already begun with the adoption of natural language processing to analyze political texts, speeches and the like. However, the potential applications expand well beyond text analysis. This paper has focused on introductory examples to highlight the process and methods for using ML and we conclude by emphasizing three points about the promises and pitfalls for the use of ML.

First, learners rely on the data to make predictions and build models. How the data is represented, what is present in the data, what is missing, and what social bias may permeate the data can impact what predictions learners can make. This makes the researcher's understanding of the data and how it is coded imperative in the ML process. Just like any method, learners are still limited by the data for accurate and generalizable estimations.

Second, it is often the case that more complex learners are better at making accurate predictions, often at the cost of interpretability. Not all good learners are interpretable, but it is likely the case that complex methods are better at capturing complex or abstract concepts, making them predictable, but not easily understood. Despite this, what we can gain from learners is much greater and gives us a new framework beyond a statistical model to think about problems, modeling data and representing political processes. Learners are well equipped to process and manage a variety of data sources that have many features. ML approaches that enable feature selection or dimensionality reduction enable researchers to sort through millions of attributes from complex data to determine what is important in understanding political processes.

Lastly, it worth reemphasizing the role of prediction. Learners make predictions, but their accuracy may not justify a causal argument. Just as in statistical modeling, correlation

does not equal causation. Good learners find strong correlations, but do not guarantee a causal model. ML methods are not a replacement for good research designs and thoughtful theory building. ML methods can allow researches to model complexities in large datasets, generate accurate measurements from complex processes and use new types of data not well suited to traditional methods of analysis. Despite this, there are many potential uses for ML models. In the discussion above, we demonstrate how ML might be incorporated into political science tasks such as measurement and inference. The role and importance of prediction can help elevate the accuracy of political science theories and broaden the impact of research by making it valuable not only for the study of political science, but for the normative outcomes like changes in policies, prediction of important events and detection of social and political patterns.

As the role of automation and computation continues to evolve, ML will play an increasingly important role in not only how we collect and interpret data, but how we orient research and how research is conducted for years to come. The potential for the use of ML in political science can be as creative as the research questions being posed. Our goal is for this discussion and illustration to provide political scientists a framework for understanding ML modeling and encourage exploring social datasets with these skills.

References

Abi-Hassan, Sahar, Janet M. Box-Steffensmeier, Dino Christenson, Aaron Kaufman and Brian Libgoe. 2019. Large-Scale Estimation of Interest Group Ideal Points. In *Annual Meeting of the Southern Political Science Association*. Austin, TX: .

Alvarez-Melis, David, Hal Daumé, Jennifer Wortman Vaughan and Hanna Wallach. 2019. Weight of Evidence as a Basis for Human-Oriented Explanations. In *Workshop on Human-Centric Machine Learning at the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*. Vancouver, Canada: .

Bischl, Bernd, Michel Lang, Lars Kotthoff, Julia Schiffner, Jakob Richter, Erich Studerus, Giuseppe Casalicchio and Zachary M. Jones. 2016. "Mlr: Machine learning in R." *Journal of Machine Learning Research* 17:1–5.

Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning*. Springer.

Casas, Andreu, Matthew J. Denny and John Wilkerson. 2020. "More Effective Than We Thought: Accounting for Legislative Hitchhikers Reveals a More Inclusive and Productive Lawmaking Process." *American Journal of Political Science* 64(1):5–18.

Casas, Andreu and Nora Webb Williams. 2019. "Images that Matter: Online Protests and the Mobilizing Role of Pictures." *Political Research Quarterly* 72(2):360–375.

Cranmer, Skyler J. and Bruce A. Desmarais. 2017. "What can we learn from predictive modeling?" *Political Analysis* 25(2):145–166.

De-Arteaga, Maria, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi and Adam Tauman Kalai. 2019. "Bias in Bios: A case study of semantic representation bias in a high-stakes setting." *FAT 2019 - Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency* pp. 120–128.

Dietrich, Bryce J. 2020. "Using Motion Detection to Measure Social Polarization in the U.S. House of Representatives." *Political Analysis* .

Dietrich, Bryce J., Matthew Hayes, Diana Z O Brien and Diana Z. O'Brien. 2019. "Pitch

Perfect: Vocal Pitch and the Emotional Intensity of Congressional Speech on Women." *American Political Science Review* 113(4):941–962.

Domingos, Pedro. 2012. "A few useful things to know about machine learning." *Communications of the ACM* 55(10):78–87.

Doshi-Velez, Finale, Mason Kortz, Ryan Budish, Christopher Bavitz, Samuel J. Gershman, David O'Brien, Stuart Shieber, Jim Waldo, David Weinberger and Alexandra Wood. 2017. "Accountability of AI Under the Law: The Role of Explanation." *SSRN Electronic Journal* pp. 1–15.

Grimmer, Justin. 2015. "We are all social scientists now: How big data, machine learning, and causal inference work together." *PS - Political Science and Politics* 48(1):80–83.

Grimmer, Justin and Brandon M. Stewart. 2013. "Text as data: The promise and pitfalls of automatic content analysis methods for political texts." *Political Analysis* 21(3):267–297.

Grimmer, Justin, Sean J. Westwood and Solomon Messing. 2015. *The Impression of Influence How Legislator Communication and Government Spending Cultivate a Personal Vote*. Princeton University Press.

Jones, Zachary M. and Yonatan Lupu. 2018. "Is There More Violence in the Middle?" *American Journal of Political Science* 62(3):652–667.

Jordan, M I and T M Mitchell. 2015. "Machine learning: Trends, perspectives, and prospects." 349(6245).

Kim, Been, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas and Rory Sayres. 2018. "Interpretability beyond feature attribution: Quantitative Testing with Concept Activation Vectors (TCAV)." *35th International Conference on Machine Learning, ICML 2018* 6:4186–4195.

Kim, Seo young Silvia, R. Michael Alvarez and Christina M. Ramirez. 2020. "Who Voted in 2016? Using Fuzzy Forests to Understand Voter Turnout." *Social Science Quarterly* 101(2):978–988.

Kuhn, Max. 2008. "Building Predictive Models in R Using the caret Package." *Journal of*

Statistical Software 28(5).

Monroe, Burt L., Michael P. Colaresi and Kevin M. Quinn. 2008. "Fightin' words: Lexical feature selection and evaluation for identifying the content of political conflict." *Political Analysis* 16(4 SPEC. ISS.):372–403.

Montgomery, Jacob M. and Santiago Olivella. 2016. "Tree-Based Models for Political Science Data." *American Journal of Political Science* 62(3):729–744.

Murphy, Kevin. 2012. *Machine Learning: a Probabilistic Perspective*. Cambridge, MA: MIT Press.

Ng, Andrew Y. and Michael I. Jordan. 2002. "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes." *Neural Processing Letters* 28(3):169–187.

Ribeiro, Marco Tulio, Sameer Singh and Carlos Guestrin. 2016. ""Why should i trust you?" Explaining the predictions of any classifier." *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 13-17-August:1135–1144.

Roberts, Margaret E., Brandon M. Stewart, Dustin Tingley, Christopher Lucas, Jetson Leder-Luis, Shana Kushner Gadarian, Bethany Albertson and David G. Rand. 2014. "Structural topic models for open-ended survey responses." *American Journal of Political Science* 58(4):1064–1082.

Shmueli, Galit. 2010. "To explain or to predict?" *Statistical Science* 25(3):289–310.

Wallach, Hanna. 2018. "Viewpoint: Computational social science ? computer science + social data." *Communications of the ACM* 61(3):42–44.

Wolpert, David H. and William G. Macready. 1997. "No free lunch theorems for optimization." *IEEE Transactions on Evolutionary Computation* 1(1):67–82.

Zhou, Bolei, Yiyou Sun, David Bau and Antonio Torralba. 2018. "Interpretable basis decomposition for visual explanation." *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11212 LNCS:122–138.