Considerations for Machine Learning Use in Political Research with Application to Voter Turnout

Laura Moses and Janet M. Box-Steffensmeier

Abstract

Machine learning is becoming increasingly prevalent in political science research. Improving the accuracy of outcomes, refining measurements of complex processes, addressing non-linearities in data and introducing new kinds of data may be achieved using machine learning. Despite the possible uses of machine learning, a clear understanding of how to use these tools and their pitfalls is still needed. This article provides a foundational guide to machine learning and illustrates how these methods can advance political science research. We address the pitfalls of these methods as well as the specific concerns for using machine learning with social data. Finally, we demonstrate how machine learning can help understand voter turnout through an application of methods with survey data on the 2016 election.

Machine learning (ML) is an intersection of statistics and computing. 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 that is increasingly available for political science research. ML is changing the way political science addresses fundamental questions of causation and allows researchers to leverage new and different types of data to study social phenomena (Grimmer, 2015). These tools improve the measurement of critical and complex concepts like ideology or influence (Abi-Hassan et al., 2019). Already, ML applications have been used widely for text analysis (for example; Monroe, Colaresi and Quinn (2008), Grimmer and Stewart (2013), Roberts et al. (2014)). More recently, ML has been 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)). There are excellent texts on the statistical underpinnings of ML (e.g., Bishop (2006) and Murphy (2012)).[1] Yet, there are important aspects of ML beyond the statistical form of a model that need to be considered for ML applications to be successfully developed and implemented for political science that are not explicit in these texts. Our goal is to present the principles of ML, strategies to develop ML applications and address data specific considerations when using these tools to advance the study of political phenomena.

The most prevalent and relevant ML approaches for political science are supervised and unsupervised ML. In unsupervised ML, 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. The learning comes from assumptions about the structural, probabilistic properties, or algebraic properties of the data. The kinds of knowledge discovery that unsupervised ML enables researchers to make estimates for missing data and finds relevant patterns that allow for

[1] ML is easily implemented 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).

dimensionality reduction which can uncover patterns and new causal mechanisms (Jordan and Mitchell (2015), Shmueli (2010)).

In supervised ML the outcome values are known. ML can be used for regression or classification. Classifiers are learning systems that take input features and output discrete values or classes. Consider classifying individual ideology as “liberal” or “not liberal”, 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 and $\hat{y}$ 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.* These learners are the most ubiquitous and adopted processes for political science 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]

In this paper, we focus on the general principles of ML necessary for application in political science. In the next section, we discuss the composition of all ML learners and the implications that different representations and learning objectives can have on the model results. Then, we discuss the importance of prediction and some of the modeling difficulties this framework creates. Importantly, we next demonstrate ML and how to consider interpretablilty for models in which prediction is central. With these examples, we conclude by discussing how data in ML applications requires some careful considerations, but ultimately enables flexible solutions for political science that are often roadblocks for classical methods.

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

Table 1: Supervised & Unsupervised ML Learners

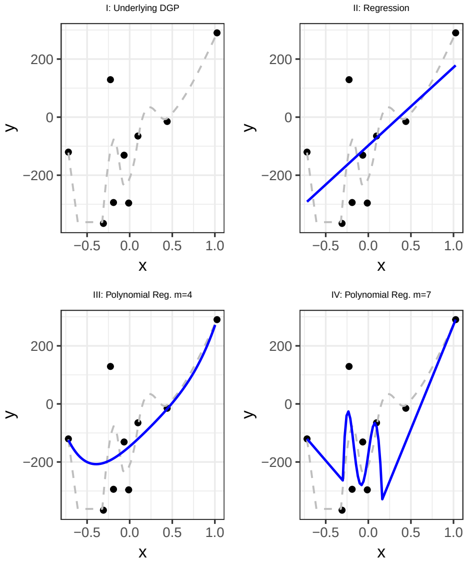
Type	**Data Transformation**	**Interpretation**	**Learners**
Supervised	Predicts outcomes or classes on new data by estimating a model and applying the model to unseen, new data to generate predictions	Requires the target variable values be known	Naive Bayes Nearest Neighbors Support Vector Machines 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

1 Building Learners

Determining which learning algorithm to use out of the thousands available can be a daunting challenge. Unpacking how all ML models are constructed can help situate what may be of importance for any given application. Learners use two things, data and some desired results. The model created has 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 or Neural Networks are different types of representations. Different representations will encode and express models differently. A model needs to be understood by the computer or it cannot be learned. Even the most powerful learners can produce poor results if the representation used in not appropriate for the data type (Domingos, 2012).

Representation also determines the hypothesis space or possible set of algorithms that can be learned given the type of model being used. How we represent information impacts how well a model describes the data generation process. How a learner is represented determines the set of candidate learners that can be considered. Referring to 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 line. Modeling this process with a linear model that uses two 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. While the model fits all the data points in panel four, it is not representing the underlying process. The representation in panel four would not generalize well and fails to make accurate predictions because the representation is too expressive. How does a researcher go about selecting the best one? This question leads us to the second building block of the learner, the objective function. This function helps the learner differentiate among good candidate

Figure 1: Possible Representations of Data



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

algorithms and bad candidate algorithms.

The objective function is an evaluation function that learners use during training, based on how the data is being represented. 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. Using the widely familiar example of linear regression, the objective is to find the best prediction of the outcome, 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. Some other choices for measuring the objective or the loss include predictive accuracy and error, precision, recall, information gained, Kullback–Leibler (KL) divergence, and minimization of margins. The objective function is simply 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.

Optimization is 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 objectives. The optimization is essential to determine the outputs produced by the model. In maximum likelihood models, we aim to find the maximizing value of each parameter, and optimization is how we find those values. In most cases, ML tools predetermine how optimization is done, but it is 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 instance-based learning techniques that make no distributional assumptions (Domingos, 2012). While 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. The next sections touch on some of the key issues to consider when selecting a learner. In subsequent sections, it will become clear that other decisions in an ML project can be more important than the choice of learner.

2 Prediction is Paramount

Prediction is central to ML. Unlike applied statistical models, which often aim to explicate how the features relate to the outcome, the primary goal of ML is to *generalize* so that the ML model can create accurate predictions for data that was not used to fit the model. Both generative models and ML predictive models are rooted in the practice of answering questions by estimating a model from the data to make statements about the outcome of interest. A generative model is concerned with explaining how inputs relate to the outputs. For example, a linear regression model,$f(x) = x^T\beta$, 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. The coefficients minimize the sum of squared residuals, which can be thought of as a *objective function*, to ensure that the coefficient estimates give the best model fit in the data sample to predict the values y, but not necessarily the best predictions out-of-sample.

The goal of ML is to find models that make accurate predictions, de-emphasizing how well coefficients may explain the sample. This requires a framework that allows for a model to be learned on one set of the data, the *training set*, which is used to train the model to find patterns, and a *test set*, which is composed of withheld observations not used to fit a model. The test set is used to determine how accurate the model is at predicting out of sample, beyond the training data. Like generative modeling, ML also 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.

ML modeling relies on these two connected principles, prediction and generalization. Generalization is the learner’s ability to adapt to previously unseen data and create accurate predictions. Learners are better able to make predictions based on their ability to generalize from data or extrapolate the patterns and features 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 that best account for the bias in the data.[3]

Even when reserving data in a *test set*, pitfalls and misleading results from a classifier can happen. Once a model is created and tested on test data, if that test data is then used to tune parameters too much, it can lead to biased results. To avoid this, cross-validation, or randomly partitioning data into subsets holding out each one while training on the rest, then testing each learned classifier on the examples it did not see, and averaging the results can help avoid overfitting, which we will discuss in detail. The predictive properties of models allow us to create better models by validating the outcomes and generalizing from examples. ML allows verification of theoretical causal claims using the principle of generalization. Beyond model validation and confirmation, the questions 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 being central, ML models rely on the training error to inform the researcher if the model is a good fit for the data as a stand-in for the test error until test time. 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

[3] 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.

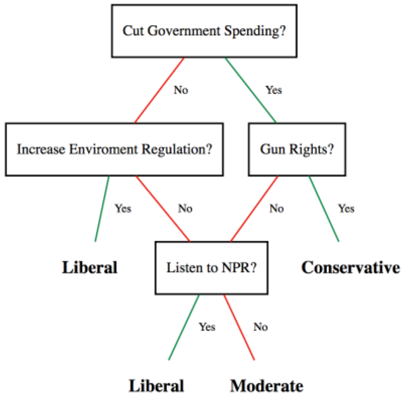
data.

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.[4] Creating a good model that can predict beyond the training set can quickly become only guessing when it comes to predicting ideology for survey responses not yet seen by the model. Prior knowledge or assumptions can help determine a useful 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 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 vectorizing a data set. However, 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

[4] 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 $\frac{2^{16.7}}{2^{75}} = 2^{-58.3}$.

Figure 2: Decision Tree for Predicting Vote Ideology



(Bishop, 2006).

3 Pitfalls: Overfitting and High Dimensions

To successfully utilize ML, practitioners need to understand their potential in the context of the difficulties in implementation. The two most common pitfalls in need of exposition 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 (KNNs) 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 determine the targets.[5] Decision trees determine which features in the data are best at predicting the target or target class by starting with features that provide more 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 modeling choices were made once we determine the splits of the data. The decision tree continues to recursively

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

split the data until targets are properly sorted. 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.

Overfitting happens when data includes many, weakly informative features unneeded for prediction. The learner would be overfitting because our learner is using features that weakly predict the outcome and there is more variance in the data. One way to combat overfitting 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 subtrees. By doing this, sections of the model that provide less information for accurate predictions are removed.

How can we tell if a learner is overfitting? The errors made in predictions can be decomposed into bias errors that result from wrong model assumptions, variance errors resulting from sensitivity to small changes in the training data, and random noise.[6] 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 new data or 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

[6] 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". When models fail to generalize well because they latch on to the variance in the data, they overfit. The aim is to balance between these two competing forces to minimize errors overall.

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

After overfitting, the loss of intuition and interpretation as dimensionality increases is the biggest pitfall in using ML. Many ML tools that work fine in lower dimensions become intractable when the input in high dimensions, this is often referred to as the "curse of dimensionality." Dimensionality increases as data contain more features, more information, more observations, and the data occupies a smaller and smaller percent of the hypothesis space. Even if the features are informative, in high dimensions, the similarity is more difficult to define, causing a model to be intractable and leading to the challenge of creating accurate predictions, because the training set covers a smaller and smaller fraction of the hypothesis space. Increased dimensionality of data, or very large data, makes ML necessary; but the unintuitive nature of increased dimensions makes it challenging. High-dimensionality can cause models to break because of the increased noise introduced to the learner. This can also happen when all the features are contributing to prediction because the observations will have increased similarity, making them closer to one another along different dimensions of the data.

All learners are affected by dimensionality to some extent. Weak learners are particularly sensitive. Nearest neighbor learners have weak assumptions, which is a double-edged sword as weak assumptions are useful 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 in the dataset, which can lead to failure in determining which observations are most alike in high-dimensional datasets.[7]

Consider again trying to determine ideology through a series of yes or no questions. As the number of dimensions increase, the number of training examples required to locate the boundary between classes also goes up exponentially. Thinking about separating liberals and conservatives in two dimensions is intuitive; we can think about drawing a decision boundary between the two classes. In twenty or more dimensions though, understanding what is happening is less intuitive, which can make finding a good ML model challenging. Even with only 20 questions, each with two possible answers, there are one million possible examples. Each additional feature increases complexity. If each question is only weakly informative, the added information may add irrelevant dimensions in the data and become more difficult to predict the outcome accurately. 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 Good Learners

All learners have hyperparameters that determine the form of the model. The key test of a learner's ability to generate accurate predictions is to use the learner to predict data *not* used to train the model. Regardless of the size of a dataset used to train a learner, those exact cases are unlikely to appear again in new data. Hyperparameters 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 the values of these hyperparameters can not only change the outcomes and the accuracy but can also influence the learner's parameter values and the

[7] 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.

final form of the model. The selection of which hyperparameters are optimal for a given data set requires a new version of the model each time. This process is sometimes called "tuning" to better capture a prediction or measurement. If adjustments to hyperparameters are done on the test set or the entire data set after the model's parameters are estimated, multiple adjustments of these parameters can result in false predictive accuracy and not generalize well to cases beyond the initial analysis. 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. 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.

Another strategy to support good learning is by using many learners or combinations of learners. The performance of many different learners may tell us something new about the data or political process of interest. Combining learners can minimize overfitting and maintain generalization. Combinations, called ensemble learning, are notional algorithms that combine several learners, either of the same type or different types. 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. Then the model is applied to each re-sampled training set and the best fit for the data is determined through voting. Bagging can reduce variance without increasing the bias of a model. *Boosting* 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 voting to determine the best prediction. Unlike bagging, the base learner is trained in sequence on weighted versions of the data, not random re-samples of the data. This allows boosting to leverage the dependence between the base learners. While bagging decreases variance, boosting decreases bias.

These methods can also be extended to use different classes of models together in one ensemble to create better predictions or classifications. This is a common approach for highly dimensional data and can be used to process text information. Grimmer, Westwood and Messing (2015) use a heterogeneous ensemble 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 often 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 beyond the samples collected for analysis.

5 Illustration & Interpretation

We provide two examples to highlight the advantages of ML while also illustrating their relative strengths and challenges. First, we evaluate the performance of commonly used learners: classification and regression tree (CART), Random Forest (RF), Multi-layer Perceptron Neural Network (MLP), Naive Bayes (NB), a Support Vector Machine (SVM)[8], and AdaBoost. Then, we compare the results with the other methods. Second, we illustrate what ML can tell us about who participated in the 2016 general election. There are numerous theories about voter turnout in major presidential elections, and it requires using many features to assess the theories that may explain voter turnout. How these theories translate to elections during non-presidential cycles remain even murkier. This example illustrates

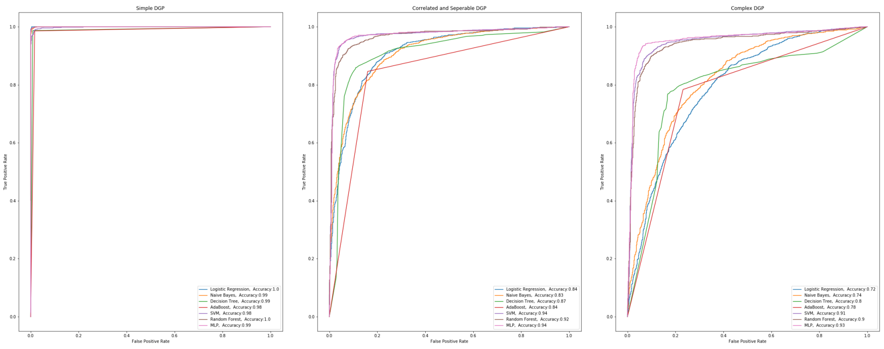
[8] The SVM is fitted with a radial basis function kernel, see supplementary materials for further details.

how ML can be advantageous to address often studied questions, like voter turnout. The advantage of ML in this example is its prediction objectives uncover the features important for predicting turnout without knowing which features are most meaningful. This approach leverages quantitative evidence created to resolve debates and long-standing questions.

An Example Using Synthetic Data

We begin by creating three datasets, a simple dataset that is linearly separable with 10 potential different features, and two more complex datasets with 25 symmetric, and asymmetric features. These more complex datasets include strongly correlated, weakly correlated and independent features.[9] The simplest classification dataset includes a fairly separable set of target values, with little noise and no redundancy or correlation between features that are important for classification. More variation in the target value is introduced for the second dataset, and finally, noise and redundancy of feature values are introduced to create a complex classification specification in the third. We split the data into test and training sets. We use cross-validation to fit the different ML models to the training data. To evaluate the relative accuracy of each ML learner, 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 different data generation processes. ROC curves are an important evaluation metric for model per-

Figure 3: ROC Curve Model Comparisons



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

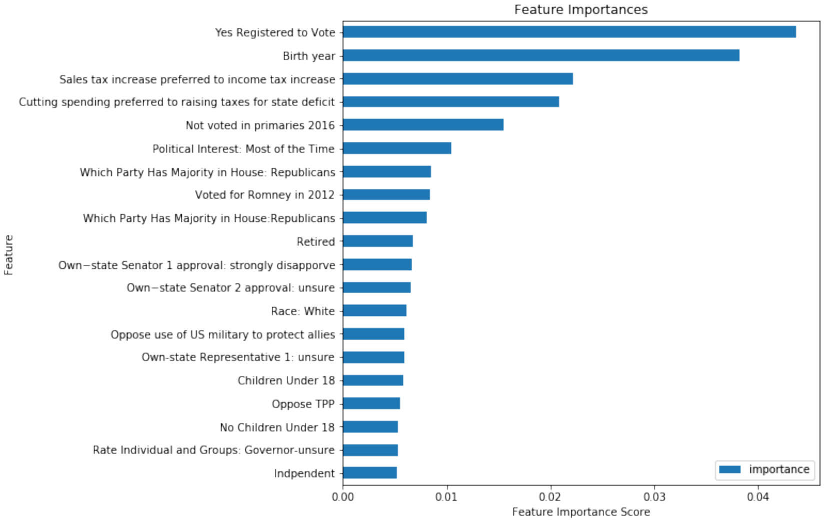
formance. Figure 3 illustrates the ROC curves for the out-of-sample prediction performance of the seven different models. The closer a curve gets to the upper left corner of the graph, the more accurately the learner is classifying the test set. The left panel shows the results of the simple DGP set 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 easily linearly separable and assumptions for many different ML models can be satisfied. Models that are easy to interpret and implement, like logistic regression, do as well as the ML models. Across the other panels, as the DGPs increase in complexity, more complex learners do better and are better at generating accurate predictions for the outcome of interest. The Random Forest and Neural Network do much better than a decision tree or logistic regression. Overall, 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. We next look at an example of how ML can support political science research.

Predicting & Explaining Voter Turnout with ML

In order to make valid causal claims about political processes, 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 can be used to develop an inclusive and comprehensive understanding of voting behavior in 2016.

We use the Cooperative Congressional Election Study (CCES), to investigate voter behavior in 2016. 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 CCES that capture features important to voting behavior like age, race, education, and income; political interest, behavior questions and policy questions.[10] We one-hot encode the dataset, meaning that we transform categorical variables to binary variables, resulting in 449 variables.[11]

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



We fit a Random Forest model and a Naive Bayes model since both allow us to interpret features used in predictions and explore the results of the Random Forest, which was more accurate. We assess what Random Forest model predictions can tell us about voter turnout behavior. As with all ML models, the hyperparameters may impact the final accuracy and

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

[11] This is a process to ensure better performance, common in ML applications. It also ensures categorical responses to survey questions are not treated as ordinal variables.

results of a model, so we optimize the hyperparameters parameters using grid search.[12] We then select the top twenty features from the Random Forests, seen in Figure 4 which shows the feature importance plot. The ML model identifies which features are the most important in predicting turnout. For each feature, the sum of the information decrease across every tree of the forest is accumulated every time that variable is chosen to split a node. The sum is divided by the number of trees in the forest to give an average. While the scale is irrelevant the relative values between features are useful. These variables are consistent with research on voter turnout. If a respondent is not registered to vote, they are less likely to turnout of an election. Also, the age of the respondent, interest in politics and past voting behavior are all relatively important in predicting turnout. In addition, if they did not vote in the primaries, which is highly correlated with not turning out for the 2016 general election.

ML models are most useful when the model is interpretable. Predicting voter turnout with these data can also be done with SVMs or neural nets. We also train a Neural Network on this data, which was as accurate as the Random Forest but, has few explainable model components. We illustrate the outcomes produced from the Random Forest, as they allow us to understand what features are used as predictors. In addition to validating theory, having an interpretable model allows erroneous predictors to be identified and better target the sources of errors in a model. While other types of ML models may generate more accurate predictions, there is less explanation derived from the model. This does not make other less interpretable ML models less useful. 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. If the voter turnout outcome were to be used in subsequent analysis, complex and accurate models like Neural Networks may also be valuable for generating accurate measurements. This illustrates how ML can 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.

[12] See Supplemental Materials for more details.

6 Data & Model Considerations

Interpretation also allows for an explanation of decisions and choices made in identifying an accurate learner. 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 help to 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 can highlight explanations in social science is critical. Learners that allow for contrast, or relational to compare outcome predictive values or the relationship of the inputs and outputs are most needed for ML to be interpretable for political science. 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 before others should be considered.

Once a learner has been selected and tuned, the learner can also be a 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. 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.

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 error is apparent and can be simple to diagnose. Detection of rare events like conflict or ensuring minorities are represented in surveys can be ensured with techniques like up-sampling or down-sampling, to create balance in the data across classes. Biases can be prevented by maintaining an explicit representation 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 (See: Montgomery and Olivella (2016) for a detailed discussion of tree-based models). 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 fast becoming a standard for quantitative analysis in scientific research across disciplines. ML has evolved into a discipline itself, and like all disciplines, there are evolving principles guiding research and specialized sub-fields. This article expounds upon some of the most salient items 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. As these methods become increasingly prevalent, and for them to have maximum potential to impact political science, practitioners 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 for the study of political science. 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 to 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 interpretable. 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 analysis methods. 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 application of ML in political science can be as creative as the research questions being posed. Our hope is that this discussion and illustration will 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 and Nora Webb Williams. 2019. "Images that Matter: Online Protests and the Mobilizing Role of Pictures." *Political Research Quarterly* 72(2):360–375.

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.