Supplemental Materials: Improving Supreme Court Forecasting Using Boosted Decision Trees[1]

Aaron Russell Kaufman
PhD Candidate
Department of Government, Harvard University
1737 Cambridge Street, Cambridge, MA 02138, USA
(818) 263–5583
aaronkaufman@fas.harvard.edu

Peter Kraft
PhD Candidate
Department of Computer Science, Stanford University
353 Serra Mall, Stanford, CA 94305, USA
kraftp@cs.stanford.edu

Maya Sen
Assistant Professor
John F. Kennedy School of Government, Harvard University
79 John F. Kennedy Street, Cambridge, MA 02138, USA
maya_sen@hks.harvard.edu

Abstract

Though used frequently in machine learning, boosted decision trees are largely unused in political science, despite many useful properties. We explain how to use one variant of boosted decision trees, AdaBoosted decision trees (ADTs), for social science predictions. We illustrate their use by examining a well-known political prediction problem, predicting U.S. Supreme Court rulings. We find that our ADT approach outperforms existing predictive models. We also provide two additional examples of the approach, one predicting the onset of civil wars and the other predicting county-level vote shares in U.S. Presidential elections.

Many thanks to Matthew Blackwell, Peter Dilworth, Finale Doshi-Velez, Phillipa Gill, Gary King, Brian Libgober, Chris Lucas, Luke Miratrix, Kevin Quinn, Jeff Segal, Robert Ward, and participants at the Computational Social Science Institute Seminar at University of Massachusetts, Amherst for helpful conversations and valuable feedback. We also thank Josh Blackman, Michael Bommarito, and Dan Katz for comments during early stages of this project.

[1]Replication materials available at the *Political Analysis* Dataverse: https://doi.org/10.7910/DVN/JJCXTH (Kaufman et al., 2018)

1 Appendix A0: Importance of Supreme Court Prediction

As one of the co-equal branches of the federal government, and as the chief interpreter of the U.S. Constitution, the Supreme Court every year has the opportunity to decide cases pertaining to deep, important questions on civil rights, First Amendment law, religious liberty, bankruptcy, taxation, elections and redistricting, separation of powers, Presidential powers, national security, criminal defense rights, and the death penalty. Recent cases have involved the constitutionality of the Affordable Care Act (one of the largest pieces of legislation in recent years), whether the federal government must recognize the marriages of same-sex couples, and the scope and legitimacy of the Voting Rights Act. Because of its ability to impact huge swaths of American policy and politics, the Supreme Court's proceedings are followed closely by many, including hundreds of journalists and scholars and thousands of members of the public.

Despite this public interest, the Supreme Court—unlike the legislative and executive branches—conducts all of its decision making privately. Justices read briefs, deliberate, seek guidance and advice from clerks, and research issues exclusively in the privacy of their own chambers. Oral arguments, which provide the only window into the possible leanings of the Justices, are immediately analyzed—but even these are not open to members of the public nor are they televised. Thus, the public oftentimes has limited information in trying to suss out the leanings of the Justices on these important issues. Moreover, this period of uncertainty can last for months. Oral arguments on important cases often take place in October, but decisions are not handed down until June—often a 10-12 month gap during which people affected by the rulings must proceed under substantial uncertainty.

To give a concrete example of this uncertainty, and why predicting Court decisions is substantively important, we consider *Windsor v. United States* (2013), which concerned the legality of the Defense of Marriage Act (DOMA), which forbade the federal government (including federal agencies, such as the Internal Revenue Service) from recognizing state-sanctioned same-sex marriages. The case was argued in October of 2012, but the *Windsor* ruling striking down DOMA was not handed down until June of 2013. Tens of thousands of marriages have been affected by the ruling, including the marriages of couples who married in the ten-month period between October and June. Indeed, the Supreme Court was actually still privately deliberating during the 2012 tax season, meaning that married LGBT couples submitted their taxes *not knowing whether they were married or not* in the eyes of the federal government. Many couples filed their taxes separately, expecting the Supreme Court to allow the federal government to continue to refuse to

recognize their marriages. The example of the same-sex marriage ruling in *Windsor* illustrates both how the Court decides cases of intense national and personal interest and how prediction is highly important—not just to the parties immediately involved but to the public more broadly.

Within the scholarly community, Supreme Court prediction has long been a topic of intense interest, both scholarly and journalistic, since at least 1948 when judicial politics scholar C. Herman Pritchett showed that ideology was a useful predictor of the way the Justices voted together (or not) during the era of Franklin Roosevelt. Since then, numerous theories have proliferated regarding what covariates are most important for predicting Supreme Court decision making and Justices' voting and, thus, how the Court will rule on important cases. Some of the more prominent papers and studies have noted the particularly important role of ideology in predicting Supreme Court decision making and, thus, aiding scholars in predicting the Court's eventual rulings (e.g., Martin and Quinn, 2002). Other scholars propose institutional constraints, while others rely on personal characteristics of the Justices. Others consider case covariates, while others consider the predictive and influential power of public opinion. However, all of these papers focus on understanding Justices' decision making, insofar as decision making allows us to (1) better understand the Court and its motivations and (2) allows us to predict rulings.

We summarize some of these studies below:

- Pritchett (1948) predicts the rulings of the Roosevelt-era Supreme Court on the basis of ideological divisions.
- Murphy (1964) more generally incorporates judicial strategy in tandem with ideology/partisanship to predict the votes of Supreme Court Justices.
- Roeder (2015) is a journalistic writeup of the CourtCast model, describing the substantive importance of predicting Supreme Court cases.
- Kort (1957) provides of one of the first large-scale quantitative prediction of Supreme Court Justice behavior.
- Segal (1984) predicts how the Supreme Court rules in search and seizure cases using case covariates.
- Aliotta (1987) shows that, in predicting Supreme Court decisions, it is best to use both case features and personal features of the Justices.
- Segal and Reedy (1988) note that among sex discrimination cases, the solicitor general's presence as a litigator is predictive of court outcomes.

- Tate and Handberg (1991) argue that Justice attributes are important predictors even in eras of U.S. history marked by less partisan polarization on the court.
- Spiller and Gely (1992) point out that Congressional policy predicts how the Court will rule on labor relations decisions.
- Segal et al. (1995) show that the ideological leanings of Supreme Court justices, as measured by the content of newspaper editorials, strongly predict the Justices' decisions on economic and civil liberties cases.
- Kearney and Merrill (2000) show that amicus briefs, especially briefs from well-respected sources, are predictive of court decisions prior to 2000.
- Epstein et al. (2001) theorize that the preferences of other institutional actors, such as Congress and the President, may predict the Supreme Court's judicial behavior.
- Bergara et al. (2003) estimate that from 1947 to 1992, one third of all Supreme Court cases were constrained by Congressional preferences.
- Ruger et al. (2004) compare the predictive results of a simple statistical model to that of legal experts during the Court's 2002 term.
- Martin et al. (2004a) show, following median voter theorem work in Congressional studies, that the ideological leanings of the median Supreme Court Justice predict how the court as a whole will rule.
- Shullman (2004) qualitatively shows that oral argument proceedings convey important information about how Justices are likely to rule.
- Cherry and Rogers (2006) use the consensus results from online information markets to predict Supreme Court outcomes.
- Johnson et al. (2006) rely on Justice Blackmun's personal accounts of oral argument quality to show that litigators who perform better during oral arguments are more likely to win a favorable decision.
- Martin and Quinn (2002) develop ideology scores for the Justices using a dynamic Bayesian IRT model. These Martin-Quinn scores have been widely used in the literature to predict Supreme Court decision making and to understand the role of ideology in predicting Justices' decision making.
- Cameron and Park (2009) formulate a new measure of judicial nominee ideology that better predicts how nominees will rule by incorporating additional pre-nomination characteristics.

- Epstein et al. (2010) also use data from oral arguments and analyze patterns of speech in predicting how the Court will rule.
- Casillas et al. (2011) show that public opinion is a predictor of judicial decisions, especially in low salience cases; Giles et al 2008 argue that the mechanism by which public opinion predicts judicial decisions is independent of membership change on the court.
- Black et al. (2011) show, using data from 2004 to 2008, that Justices are more likely to use negatively-charged language directed at the losing litigator.
- Dietrich et al. (2016) use oral argument audio data to predict Justice-level vote outcomes as a function solely of vocal pitch, finding that vocal inflections (with no additional textual data) accurately predict how the Court will rule.

Thus, an extremely rich literature has plumbed Justices' decision making processes, along the way highlighting which factors are particularly predictive in predicting how the Justices will vote and, thus, how the Court will rule.

From a more pragmatic perspective, predictions of how cases might go could offer guidance to lower court judges and lawyers. For example, lower-court judges tend to dislike being overturned by higher courts, and they routinely weigh the probability that the Court will rule a certain way versus another. This could be a particular concern if a lower-court is adjudicating an issue currently being considered by the Supreme Court (in which case a lower-court judge might simply withhold judgement until the Supreme Court reaches its decision). Lawyers, as well, may benefit since a good prediction of how the Court is likely to rule may influence their willingness to settle. Finally, other members of the interested public—including other political actors, investors, and financial markets may benefit from being able to better predict Supreme Court rulings. For investors, Supreme Court decisions (for example, those involving corporate law and transactions) can cause notable financial uncertainty and therefore market volatility; good predictive models may smooth out this volatility and help financial and other industries plan for the future. For political actors, understanding which way the Court is likely to go allows them to respond with legislation or with appeals to the public—both of which have been on display in recent cases such as those involving the Affordable Care Act (*National Federation of Independent Businesses v. Sebelius*) and the Voting Rights Act (*Shelby County v. Holder*).

We also note an inherent tradeoff between predicting a case *early* and predicting it *better*. Earlier predictions allow more time to prepare for policy changes, but less data may be available for making those predictions. Our analysis shows that it is possible to predict a case with 71.3% accuracy given only the case

covariates, which are known as soon as the Supreme Court grants agrees to hear a case via the granting of *certiorari*. However, we can predict a case with 74% accuracy after the Court has heard oral arguments. This time period may be as short as a month or as long as 10 months, during which there could be substantial policy uncertainty.

2 Appendix A1: Petitioner-Wins Baseline Accuracy

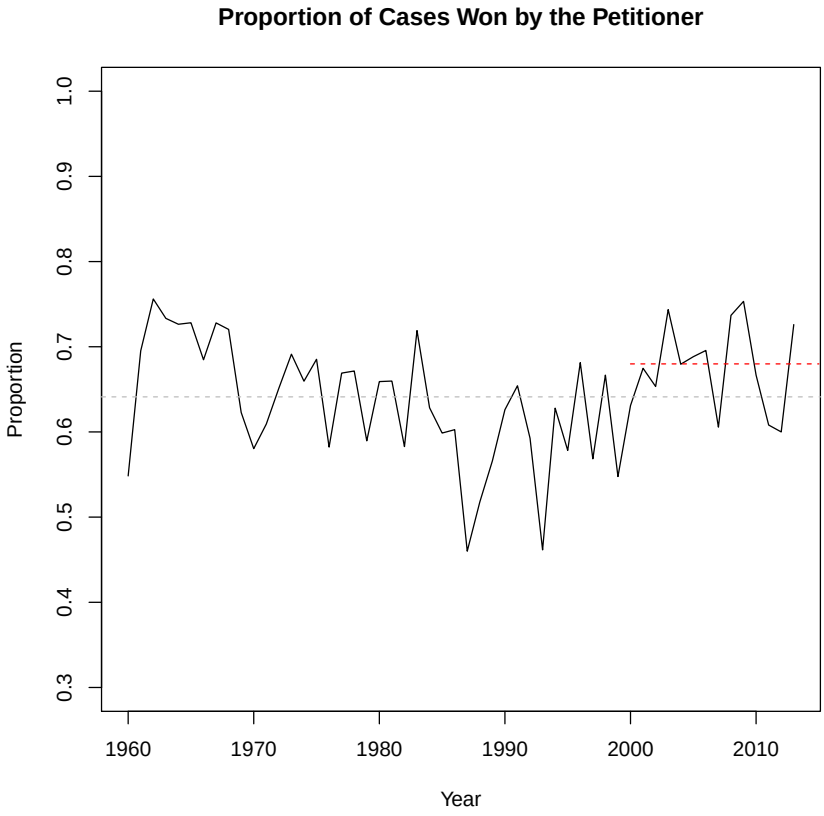

Figure 1: Percentage of Supreme Court cases won by the petitioner. This has averaged 64% since 1960 (gray dashed line) and 68% since 2000 (red dashed line).

3 Appendix A2: Accuracy over time

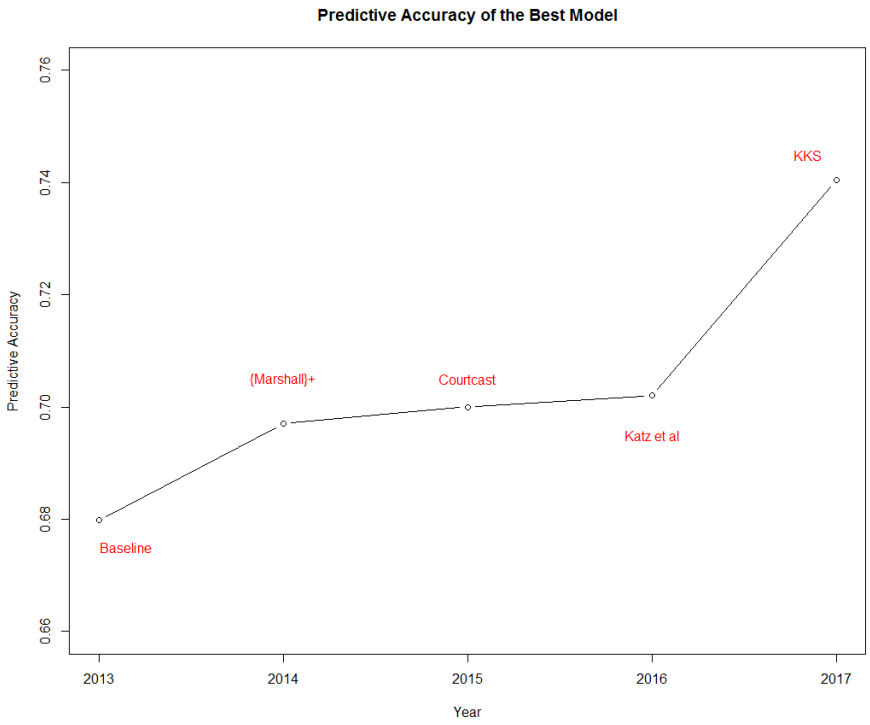

Figure 2: The percentage accuracy of the most accurate Supreme Court prediction model for each year from 2013 to the present.

4 Appendix B: Additional Decision Trees

Below is one decision tree drawn from our KKS AdaBoosted Decision Trees model. Each box represents a feature split, indicated by the first text row in each box. This tree begins with the "lawyers" feature, indicating the relative number of lawyers arguing for the plaintiff and the respondent, split on the value -0.5. The box is very blue, indicating that when that condition holds, it is highly probable that the respondent wins the case. The second row of text, Gini impurity, indicates the probability of incorrect classification based on that node. For the "lawyers" box, this means that classifying court decisions solely on whether "lawyers" ≥ -0.5 would incorrectly classify cases 45.4% of the time. The second row of boxes is the next layer of feature splits. If the condition in the "lawyers" box holds, the tree moves to the left node; otherwise, it moves to the right node. These trees are all three layers deep, though it is possible to

construct decision trees with more or fewer layers. The end points of each decision tree are indicated in the bottom rows. For example, in the first tree, if for a certain case the conditions in the "lawyers" box, "KENNEDY_pet_questions" box, and "SCALIA_cc_ratio_res" box are all true, then the prediction for that case is that the Respondent will win, and empirically, the Respondent wins more than 65% of those cases, as indicated by the Gini impurity value.

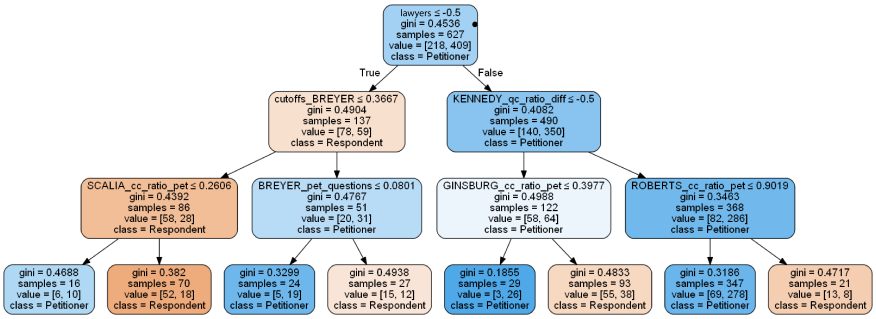

Figure 3: An example decision tree.

5 Appendix C: Variable Selection & Feature List

For each Justice, we compute the following features: questions asked to the petitioner/respondent, words spoken to the petitioner/respondent, interruptions of the petitioner/respondent.[2] We transform these in two ways. First, we create dichotomous indicators for each Justice indicating if that Justice asked more questions, spoke more words, or interrupted more frequently the petitioner or the respondent attorney (27 total variables). Second, we calculate for each Justice the appropriate ratios of speech targeted toward each attorney for words spoken, questions asked, and interruptions.[3] We find that, generally, the most predictive oral argument-derived features are ratios.

Since ADTs are largely black boxes where features enter and predictions are returned, determining which covariates contribute most to the model's success can be difficult. One commonly used method to extract feature importances from tree-based models involves "feature depth" (Archer and Kimes, 2008), which is natively implemented in python's scikit-learn module. Since ADTs consist of decision trees that are ordered variable splits, features that systematically appear earlier in the decision tree are more important to the model. A covariate's feature importance, then, is proportional to the average number of times that feature appears in the decision tree, weighted by how early in the tree it appears; more simply, higher values indicate more strongly predictive features.

We calculate feature importance for our ADT model and present the results below. We find that the most important features derive in equal parts from case-level covariates from the Supreme Court Database and the oral argument transcripts. In order of importance, the features 2, 5, and 9 come from the former, and features 1, 3, 4, 6, 7, 8, and 10 come from the latter. These ten variables together account for more than 30% of the value of the model, strongly suggesting the mutual beneficiality of both data sets.

In Table 1, we indicate the name of the top 20 features by importance. Then, in Table 2, we list all 55 features in alphabetical order. Full descriptions of those features are in footnotes. In Table 1, we bold the importance of features which are also identified as statistically significant in a naive OLS model rather than our ADT model. While the OLS model does identify 4 of the top 5 features, it only indicates as significant 9 of the ADT model's top 20 most important features.

[2]For consistency in comparisons, we compute these measures identically to the CourtCast model.

[3]For example, for interruptions, we calculate for each Justice the ratio of times the Justice interrupted the liberal litigator versus the conservative litigator. If Scalia interrupted the liberal litigator six times but only interrupted the conservative litigator two times, this value would be $(6/8)/(2/8) = 3$.

	Feature	Importance
1	Relative Number of Lawyers[4]	**0.060**
2	Issue Area[5]	**0.031**
3	Kennedy-Petitioner Questions[6]	**0.030**
4	Scalia-Respondent Questions	0.029
5	Case Origin: Circuit	**0.028**
6	Ginsburg-Petitioner Questions	0.027
7	Kennedy QC Ratio[7]	0.027
8	Roberts-Petitioner Questions	0.027
9	Reason for Cert	0.026
10	Interruptions	**0.024**
11	Ginsburg WC Ratio[8]	0.024
12	Scalia WC Ratio	0.024
13	Ginsburg Question Difference	**0.023**
14	Ginsburg Questions to the Respondent	**0.020**
15	Kennedy Cutoff Ratio[9]	0.020
16	Kennedy Question Target[10]	**0.020**
17	Lower Court Disposition Direction[11]	0.019
18	Scalia-Petitioner Questions	**0.019**
19	Breyer Cutoff Ratio	0.019
20	Lower Court Disposition	0.019

Table 1: The 20 features which contribute most to the model's accuracy. Across all features, importances sum to 1. Bolded features are those whose coefficients are statistically significant in a naive OLS model.

	Feature	Importance
1	Administrative Action	0.011
2	Breyer: Comments to the petitioner divided by total comments	0.015

[4]This measure indicates which side had more lawyers present during oral argument proceedings.

[5]This is the issue area of the case, as coded by the Supreme Court Database. Note that while this variable is literally coded after the fact of the case being argued, and thus may be considered post-hoc, we argue that the coding is sufficiently objective to be robust to the outcome of the case.

[6]This is a count of questions asked of the petitioner by Justice Kennedy.

[7]This measure is the difference of ratios of petitioner and respondent questions to total questions. If the petitioner was asked 3 questions and the respondent was asked seven, this ratio is $7/10 - 3/10 = 4/10$.

[8]This measure is the difference of ratios of words spoken to the petitioner and respondent, to total words spoken: If Ginsburg spoke 100 words to the petitioner and 50 to the respondent, this ratio is $100/150 - 50/150 = 50/150$.

[9]This measure is the difference of ratios of the times Kennedy interrupted the petitioner and the times Kennedy interrupted the respondent.

[10]This measure indicates whether Kennedy asked more questions to the petitioner or the respondent.

[11]This measure is whether the lower court ruled in favor of the liberal or conservative side, as determined by the Supreme Court Database.

3	Breyer: Comments to the respondent divided by total comments	0.015
4	Breyer: Respondent comment ratio minus petitioner comment ratio	0.012
5	Case originated in a Circuit Court	0.028
6	Ginsburg: Comments to the petitioner divided by total comments	0.013
7	Ginsburg: Comments to the respondent divided by total comments	0.013
8	Ginsburg: Respondent comment ratio minus petitioner comment ratio	0.016
9	How many questions did Breyer ask the petitioner	0.015
10	How many questions did Breyer ask the respondent	0.017
11	How many questions did Ginsburg ask the petitioner	0.027
12	How many questions did Ginsburg ask the respondent	0.020
13	How many questions did Kennedy ask the petitioner	0.030
14	How many questions did Kennedy ask the respondent	0.015
15	How many questions did Roberts ask the petitioner	0.027
16	How many questions did Roberts ask the respondent	0.014
17	How many questions did Scalia ask the petitioner	0.019
18	How many questions did Scalia ask the respondent	0.029
19	Issue Area	0.031
20	Kenned: Respondent comment ratio minus petitioner comment ratio	0.011
21	Kennedy: Comments to the petitioner divided by total comments	0.014
22	Kennedy: Comments to the respondent divided by total comments	0.014
23	Lower Court Disposition	0.019
24	Lower Court Disposition Directon	0.019
25	Manner in which the court takes jurisdiction	0.002
26	Number of Lawyers: Ratio	0.060
27	Reason for Cert	0.026
28	Roberts: Comments to the petitioner divided by total comments	0.018
29	Roberts: Comments to the respondent divided by total comments	0.018
30	Roberts: Respondent comment ratio minus petitioner comment ratio	0.011
31	Scalia: Comments to the petitioner divided by total comments	0.016
32	Scalia: Comments to the respondent divided by total comments	0.016
33	Scalia: Respondent comment ratio minus petitioner comment ratio	0.011

34	State of Administrative Action	0.006
35	To which litigator did Breyer ask a higher ratio of questions to comments	0.015
36	To which litigator did Ginsburg ask a higher ratio of questions to comments	0.023
37	To which litigator did Kennedy ask a higher ratio of questions to comments	0.020
38	To which litigator did Roberts ask a higher ratio of questions to comments	0.015
39	To which litigator did Scalia ask a higher ratio of questions to comments	0.015
40	Which litigator did Breyer question more	0.012
41	Which litigator did Breyer speak more to	0.017
42	Which litigator did Ginsburg question more	0.015
43	Which litigator did Ginsburg speak more to	0.024
44	Which litigator did Kennedy question more	0.027
45	Which litigator did Kennedy speak more to	0.016
46	Which litigator did Roberts question more	0.008
47	Which litigator did Roberts speak more to	0.013
48	Which litigator did Scalia question more	0.012
49	Which litigator did Scalia speak more to	0.024
50	Which litigator was interrupted more	0.024
51	Which litigator was interrupted more by Breyer	0.019
52	Which litigator was interrupted more by Ginsburg	0.019
53	Which litigator was interrupted more by Kennedy	0.020
54	Which litigator was interrupted more by Roberts	0.011
55	Which litigator was interrupted more by Scalia	0.017

Table 2: All 55 features in our ADT model in alphabetical order.

Note that there are many potential covariates that we exclude from this model. Time-based covariates—for example, the year or month in which the case was heard, or the Court's median ideal point during the case—we found to *harm* our model's predictive accuracy. We also experimented with including a variable indicating whether the Solicitor General was a litigator in the case, but found it to be similarly unpredictive. As well, there are covariates which we would like to include in our model but cannot. The number and text of amicus curiae briefs filed, for example, contain a wealth of information about the case related to public and elite opinion. While several data sets of amicus curiae exist, none include cases during the period in

which we conduct our analysis.

Substantively, this feature importance table does illustrate one point of difference between this approach and, for example, an approach reliant on interpretation of a regression table. We tend to view regression analyses as particularly well suited to causal evaluations and to understanding the causal effects of interventions. By contrast, our analyses are better for identifying variables that are important for prediction—which could include variables that are not particularly causal in nature but are nonetheless quite predictive. This would be less helpful for the parties' litigation strategy, but more of meaningful for risk calculation. (Our analyses also does not attempt to satisfy the assumptions necessarily to isolate a causal effect.) For example, a key feature that has high predictive value is Justice Kennedy's questioning. Although not a causal variable, this fact very interestingly speaks to (1) the influential position occupied by Kennedy as the "median" Justice and (2) the nature the information revealed at oral argument. Indeed, although beyond the scope of this paper, this actually seems to suggest that Kennedy might have an inkling of his intended vote before or during oral arguments and these leanings translate into more (or fewer) questions of one side or another. These are not causal inquiries per se, but they do nonetheless provide important information that is of significant interest to Court observers, judicial politics scholars, and people likely to be impacted by the Court's rulings. Indeed, dozens of newspaper articles pop up after each oral argument seeking to predict how individual Justices will vote based on the nature of the questions asked at oral argument. We also note that this would be a variable that could be "significant" in a regression analysis, but (unless the scholar was specifically studying questions asked in oral arguments) it would be surprising for a researcher to include it in a regression analysis; our machine learning approach, however, automatically picks it up as highly salient in contributing to the model's accuracy.

6 Appendix D: Additional Discussion of K-Fold Cross-Validation

For a data set with n observations, we first partition the data into 10 subsets of size $\frac{n}{10}$. This algorithm first trains a model on partitions 2 through 10, then predicts the outcome measure for the first subset and records the number of correct predictions. Next, a model is trained on subsets 3 through 10 and 1, and then a prediction is generated for subset 2, recording its accuracy. This is repeated for all 10 subsets. The total percentage of correct predictions is treated as the model's out-of-sample predictive accuracy.

K-fold cross-validation is a commonly-accepted metric for model accuracy in computer science and statistics. When performing a single train set and test set split, a data set is randomly partitioned in two, a model is trained on one of the partitions, then used to predict the outcomes for the second partition. Note that a single train set and test set partition is equivalent to a 2-fold cross-validation procedure. This induces a trade-off between model power and accuracy precision: the more observations are reserved for the test set, the fewer may be used to train the model; the fewer observations reserved for the test set, the noisier the measure of out-of-sample predictive accuracy. Ten-fold cross-validation circumvents this trade-off altogether, using 90% of the available data to train the model each iteration, and averaging predictive accuracy across ten folds to increase precision. As K increases to equal N, both model accuracy and accuracy-measurement precision improve. However, computation time increases as well, so in practice, $K = 10$ is common.

7 Appendix E: Thoughts on the Use of Machine Learning in Political Science

We have two primary answers to the question of why political science has been slow to adopt machine learning methods: one is a practical reason, and one is a path-dependent reason. The practical reason is that machine learning works best when there is no measurement error in the outcome variable. Questions like "Is there a cat in this photograph?" are excellent for machine learning for the same reason that "Who will win this Supreme Court case?" is: either there is or is not a cat in a photograph, and either the respondent or the petitioner will win a Supreme Court case. On the other hand, questions like "What is the ideology of this document?" are much harder, because measuring ideology is a nuanced and error-prone endeavor. In short, most of the important dependent variables we care about in political science are noisy and difficult to measure with precision. The path-dependent reason is that political science is often focused on substantive interpretation of covariates, and often with causal implications. Machine learning is ill-suited to this approach, as the functional forms it induces around the data are not amenable to easy linear interpretation. For this reason, we believe that decision trees hold much promise: it is relatively straightforward to examine a decision tree and interpret it.

8 Appendix G: Technical Overview of AdaBoosted Decision Trees

AdaBoosted decision trees combine three powerful machine learning concepts: decision trees, ensembling, and the AdaBoost algorithm. We will discuss each in turn.

8.1 Decision Trees

Decision trees are a flexible non-parametric machine learning method for classification (categorical outcomes) or regression (continuous outcomes). The decision tree grows by optimizing "Gini impurity," measuring how mixed are classes separated by that a given split. A Gini impurity index of 0 indicates that a split perfectly separates classes, while 1 indicates that each branch of a split is evenly divided among classes.

A simple decision tree consisting of one node finds the optimal split as measured by Gini impurity, producing two branches. A decision tree with two layers then performs the same optimal splitting procedure with each branch, resulting in four categories. A decision tree may have arbitrarily many layers, but additional layers increase the risk of overfitting.

Result: Decision Tree with `n` layers

initialization;

for $i \in n$ **do**

  for $l \in 2^{i-1}$ **do**

    Find optimal split in leaf l in layer i by minimizing Gini impurity;

  end

end

8.2 Ensembling

A concern with single-tree models is that they tend to overfit: outliers and dropped or missing values can have an outsized effect on their predictions. Larger trees with many nodes may reduce outlier sensitivity, but are more prone to overfitting. For this reason, ensemble learning methods, which combine many trees in different ways, are popular in practice. The two most common ways to ensemble decision trees are bootstrapping and boosting. Bootstrapping many decision trees leads to a random forest model, while boosting leads to

a boosted decision tree model. The random forest is among the most commonly used machine learning methods, so we briefly outline it below.

In random forests, (1) many trees are constructed simultaneously using bootstrapped samples of the data, (2) each tree's decision rules are generated using random subsets of the covariates, and then (3) the trees' predictions are averaged together (Liaw and Wiener, 2002). The bootstrapping procedure serves to reduce overfitting, while the random covariate selection eliminates systematic correlations between the trees, thereby improving predictive power[12] (Ho, 2002). Random forests are also, by comparison to other ensemble methods, easy to use, with efficient implementations in `R` (Liaw and Wiener, 2002) and STATA.

8.3 AdaBoost

AdaBoosting is an ensembling method for combining multiple models *in sequence*. It is initialized by training a base model, often called a "weak learner," on the full data set. This weak learner may be any model, often a linear or logistic regression, but in this case we use decision trees as the base learner. After the model is trained, residuals are calculated as the difference between predictions and the truth. In the case of a classification problem, these residuals are binary, whereas in regression problems they may be continuous.

In the second iteration, all observations in the data set are re-weighted proportional to the size of their residuals, a new model is run, new predictions, residuals, and weights are calculated, and then the third iteration begins. The number of iterations is at the researcher's discretion, and more is better than fewer; we perform 10,000 boosting iterations in our applications. After T iterations, the result is a series of $M_t \forall t \in T$ models, each of which has a prediction $P_{i,t}$ for each observation in the data set. The final prediction for observation i is the average of all predictions for that observation: $\frac{1}{T}\sum_t P_{i,t}$.

[12] In most machine learning ensemble methods, many weakly predictive models are aggregated together. If the "weak leaners" are weakly correlated at most, then each model picks up a different piece of the model variance, and the overall model will have more predictive power. If, however, the models are all highly correlated, then the ensembling procedure will add very little, and it is sufficient to take any single model by itself.

Data: Covariates x_i, and outcome y_i for $i \in 1, N$; weights $w_{i,t}$

Result: AdaBoosted Predictions after `T` iterations

initialization;

Initialize $w_{i,t} = 1 \forall i$;

for $t \in T$ **do**

Create a model M_t to predict y_i from $x_i w_{i,t}$;

Generate predictions $\boldsymbol{P}(M_t)$;

Calculate residuals $\boldsymbol{r} = \boldsymbol{P}(M_t) - \boldsymbol{y}$;

Calculate $w_{t+1} \propto \boldsymbol{r}$;

end

Calculate final predictions: $\frac{1}{T} \sum_{t=1}^{T} \boldsymbol{P}(M_t)$;

This algorithm makes clear that ADTs scale linearly in the number of boosting iterations, but polynomially in the number of covariates and exponentially in the interaction depth of features.

8.4 Alternative Boosting Algorithms

An important feature of AdaBoost is that it can be reformulated as gradient descent, a standard nonconvex optimization procedure, with an exponential loss function. Modifying that loss function results in a number of alternative boosting algorithms. For example, replacing the exponential loss function with a logistic regression loss function results in an algorithm called LogitBoost. Using a max-margin loss function, similar to support vector machines, results in LPBoost, while adding in a majority-voting rule results in BrownBoost (Mason et al., 2000).

While there is no consensus about which boosting methods perform best, Wu et al., 2010 find that LPBoost outperforms other algorithms in a study of disease mutations. Despite this, we choose to focus on AdaBoost. The more elaborate boosting algorithms are designed to solve problems associated with particular data issues, such as noisy or mislabeled outcome data, or skewed continuous outcomes. Since we are not concerned about noisy or mislabeled Supreme Court outcomes, and our outcomes are strictly binary, we need not turn to more elaborate models. We prefer AdaBoost for its computational simplicity, its intuitive construction, and its ease of implementation: AdaBoost and LogitBoost are the only boosting algorithms readily implemented in scikit-learn, and there is no clear reason to prefer LogitBoost's logistic loss function to AdaBoost's more straightforward exponential loss. As well, many common boosting algorithms are not

amenable to binary classification problems.

8.5 Tuning Parameter Optimization

All machine learning models involve "tuning parameters" that control the behavior of the model. ADTs have two categories of tuning parameters: parameters related to the decisions trees and parameters related to AdaBoost.

The first set of parameters include the minimum number of observations allowed at a "leaf" of the tree, the minimum maximum depth of the tree, the minimum and maximum number of allowed features, the minimum number of observations required to split a node, the criterion by which to measure the quality of a node, whether to bootstrap samples in creating trees, the minimum node impurity required to split a node, and several others.

The second set of parameters are only the learning rate and the number of boosting iterations. The learning rate parameter controls the amount of predictive "weight" each new iteration may add to the final model. The default value is 1; values closer to 0 may improve predictive accuracy, but in turn require more boosting iterations. In our ADT model, we perform 10,000 iterations.

We follow best practices by optimizing both sets of parameters using a grid search (Bergstra et al., 2011). Grid search involves training the same model across many different combinations of parameters and selecting the parameter set that maximizes predictive accuracy. We implement this algorithm using the GridSearchCV module of python's scikit-learn library.

8.6 General Best Practices for using ADTs in Political Science

To summarize, we offer a series of guidelines for applied researchers in how to implemented boosted decision trees in a social science framework. These recommendations are oriented toward minimizing overfitting and maximizing predictive accuracy.

- Have a sufficient sample size. OLS requires only as many observations as variables, but ADTs work best with no fewer than 100 observations. As always, more observations are better than fewer.
- Allow the algorithm to perform variable selection; do not exclude variables that might be useful.
- Select a robust boosting algorithm like capable of generating predictions in the domain of interest. If the prediction problem is a binary classification, AdaBoost and LogitBoost are good choices. If the problem is a continuous regression problem, LPBoost may be more applicable.

- Allow the boosting algorithm to run for as many iterations as you have time for, and reduce the learning rate appropriately to avoid overfitting.
- Perform a grid search (or other search algorithm) to optimize tuning parameters.
- Measure performance against relevant benchmarks. This includes structural benchmarks like the predictive accuracy for always guessing the modal outcome, and benchmarks from the literature.
- If possible, reserve a portion of the data as a final evaluation set. Without very large data sets, it is possible to overfit parameters to a subsample, thereby sacrificing true predictive capacity for supposed out-of-sample accuracy.
- Finally, consider other models! Unless there is a theoretical reason to prefer ADTs, it is always possible that another model may be superior in any single context. Random forests, support vector machines, and neural networks are generally worth trying for any particular application.

9 Appendix H: Previous Supreme Court Prediction Models

Statistical models occasionally surpass the "petitioner wins" baseline. For example, Martin et al. compared expert predictions to a classification tree using six case-level covariates.[13] That model correctly predicted 75% out of 68 cases. Although the statistical model does beat the "petitioner wins" baseline, its findings are limited by the the small sample size of the study (Martin et al., 2004b, p. 765) and that it examined only one natural Court with highly Justice-specific covariates (Katz et al., 2014).

Following in the steps of Martin et al., recent attempts have shown reliable improvements over the "petitioner wins" baseline. {Marshall}+, which incorporates 95 case-level covariates into a predictive model (Katz et al., 2014), reports a predictive accuracy of 69.7% using a random forest variant called Extremely Random Trees. These split candidate features randomly instead of along optimal thresholds, enjoying a decreased variance in estimates at the cost of increased bias. The second attempt is CourtCast (Roeder, 2015), which uses three features derived from oral arguments transcripts: (1) the number of words uttered by each Justice when talking to the parties, (2) the sentiment of the words used, and (3) the number of

[13]These were circuit of origin, the issue area, the type of petitioner, the type of respondent, the ideological direction of the lower-court ruling, and whether the case raised a constitutional issue. Experts were free to consider any information they wished (Martin et al., 2004b, p. 762).

times each Justice interrupts. CourtCast reports a predictive accuracy of 70%. The CourtCast model is an unweighted ensemble classifier consisting of random forests, support vector machines, and logistic regression. Ensemble methods, which synthesize the results from multiple uncorrelated classifiers into one prediction, mitigate the costs of their constituent methods but often reduce the benefits. Finally, a random forest model by Katz et al. 2017 allows for dynamic, time-varying predictions and reports an accuracy of 70.2%. Despite relatively modest gains in predictive accuracy, they boast the flexibility to predict any case for which covariates exist regardless of court composition or year.[14]

[14] As of writing, we have not had access to the data or replication code.

10 References

Aliotta, J. M. (1987). Combining judges' attributes and case characteristics: An alternative approach to explaining supreme court decisionmaking. *Judicature*, 71:277.

Archer, K. J. and Kimes, R. V. (2008). Empirical Characterization of Random Forest Variable Importance Measures. *Computational Statistics & Data Analysis*, 52(4):2249–2260.

Bergara, M., Richman, B., and Spiller, P. T. (2003). Modeling supreme court strategic decision making: The congressional constraint. *Legislative Studies Quarterly*, 28(2):247–280.

Bergstra, J. S., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, pages 2546–2554.

Black, R. C., Treul, S. A., Johnson, T. R., and Goldman, J. (2011). Emotions, Oral Arguments, and Supreme Court Decision Making. *The Journal of Politics*, 73(2):572–581.

Cameron, C. M. and Park, J.-K. (2009). How will they vote? predicting the future behavior of supreme court nominees, 1937–2006. *Journal of Empirical Legal Studies*, 6(3):485–511.

Casillas, C. J., Enns, P. K., and Wohlfarth, P. C. (2011). How public opinion constrains the us supreme court. *American Journal of Political Science*, 55(1):74–88.

Cherry, M. A. and Rogers, R. L. (2006). Tiresias and the justices: Using information markets to predict supreme court decisions. *Nw. UL Rev.*, 100:1141.

Dietrich, B. J., Enos, R. D., and Sen, M. (2016). Emotional Arousal Predicts Voting on the US Supreme Court. Technical report, Technical Report.

Epstein, L., Knight, J., and Martin, A. D. (2001). The supreme court as a strategic national policymaker. *Emory LJ*, 50:583.

Epstein, L., Landes, W. M., and Posner, R. A. (2010). Inferring the Winning Party in the Supreme Court from the Pattern of Questioning at Oral Argument. *The Journal of Legal Studies*, 39(2):433–467.

Giles, M. W., Blackstone, B., and Vining Jr, R. L. (2008). The supreme court in american democracy: Unraveling the linkages between public opinion and judicial decision making. *The Journal of Politics*, 70(2):293–306.

Ho, T. K. (2002). A Data Complexity Analysis of Comparative Advantages of Decision Forest Constructors. *Pattern Analysis & Applications*, 5(2):102–112.

Johnson, T. R., Wahlbeck, P. J., and Spriggs, J. F. (2006). The Influence of Oral Arguments on the US Supreme Court. *American Political Science Review*, 100(1):99–113.

Katz, D. M., Bommarito, M. J., and Blackman, J. (2014). Predicting the Behavior of the Supreme Court of the United States: A General Approach. *Available at SSRN 2463244*.

Katz, D. M., Bommarito II, M. J., and Blackman, J. (2017). A General Approach for Predicting the Behavior of the Supreme Court of the United States. *PloS one*, 12(4):e0174698.

Kaufman, A., Kraft, P., and Sen, M. (2018). Replication Data for: Improving Supreme Court Forecasting Using Boosted Decision Trees, https://doi.org/10.7910/DVN/JJCXTH, Harvard Dataverse.

Kearney, J. D. and Merrill, T. W. (2000). The influence of amicus curiae briefs on the supreme court. *University of Pennsylvania Law Review*, 148(3):743–855.

Kort, F. (1957). Predicting supreme court decisions mathematically: A quantitative analysis of the "right to counsel" cases. *American Political Science Review*, 51(1):1–12.

Liaw, A. and Wiener, M. (2002). Classification and Regression by randomForest. *R News*, 2(3):18–22.

Martin, A. D. and Quinn, K. M. (2002). Dynamic ideal point estimation via markov chain monte carlo for the us supreme court, 1953–1999. *Political Analysis*, 10(2):134–153.

Martin, A. D., Quinn, K. M., and Epstein, L. (2004a). The median justice on the united states supreme court. *NCL rev.*, 83:1275.

Martin, A. D., Quinn, K. M., Ruger, T. W., and Kim, P. T. (2004b). Competing Approaches to Predicting Supreme Court Decision Making. *Perspectives on Politics*, 2(04):761–767.

Mason, L., Baxter, J., Bartlett, P. L., and Frean, M. R. (2000). Boosting algorithms as gradient descent. In *Advances in neural information processing systems*, pages 512–518.

Murphy, W. F. (1964). *Elements of Judicial Strategy*. University of Chicago Press.

Pritchett, C. H. (1948). *The Roosevelt Court: A study in judicial politics and values, 1937-1947*, volume 21. Quid Pro Books.

Roeder, O. (2015). How to read the mind of a supreme court justice. https://fivethirtyeight.com/features/how-to-read-the-mind-of-a-supreme-court-justice/.

Ruger, T. W., Kim, P. T., Martin, A. D., and Quinn, K. M. (2004). The Supreme Court Forecasting Project: Legal and Political Science Approaches to Predicting Supreme Court Decisionmaking. *Columbia Law Review*, pages 1150–1210.

Segal, J. A. (1984). Predicting supreme court cases probabilistically: The search and seizure cases, 1962-1981. *American Political Science Review*, 78(4):891–900.

Segal, J. A., Epstein, L., Cameron, C. M., and Spaeth, H. J. (1995). Ideological values and the votes of us supreme court justices revisited. *The Journal of Politics*, 57(3):812–823.

Segal, J. A. and Reedy, C. D. (1988). The supreme court and sex discrimination: The role of the solicitor general. *Western Political Quarterly*, 41(3):553–568.

Shullman, S. L. (2004). The illusion of devil's advocacy: How the justices of the supreme court foreshadow their decisions during oral argument. *J. App. Prac. & Process*, 6:271.

Spiller, P. T. and Gely, R. (1992). Congressional control or judicial independence: The determinants of us supreme court labor-relations decisions, 1949-1988. *The RAND journal of Economics*, pages 463–492.

Tate, C. N. and Handberg, R. (1991). Time binding and theory building in personal attribute models of supreme court voting behavior, 1916-88. *American Journal of Political Science*, pages 460–480.

Wu, J., Zhang, W., and Jiang, R. (2010). Comparative study of ensemble learning approaches in the identification of disease mutations. In *Biomedical Engineering and Informatics (BMEI), 2010 3rd International Conference on*, volume 6, pages 2306–2310. IEEE.