Predicting Supreme Court Case Decisions

Maia Ezratty
Princeton University
mezratty@princeton.edu

Will Hinthorn
Princeton University
hinthorn@princeton.edu

Mihika Kapoor
Princeton University
mkapoor@princeton.edu

Alice Zheng
Princeton University
azheng@princeton.edu

Abstract

"The prophesies of what the courts will do, in fact, and nothing more pretentious, are what I mean by the 'law.'" [5]
- Oliver Wendell Holmes

The bias of today's judiciary has become a major point of contention. With Trump as President, many question whether hidden biases are undermining the system of checks and balances and eroding the power of the law. In our project we set out to answer 3 primary questions. First, how well can we predict Supreme Court case decisions in the U.S., based on previous decisions? Secondly, are there shifts in opinions of individual Justices over time? Lastly, how do written opinions reflect Justices' votes? We investigated the factors which influence the highest court in the United States and examine hidden biases in our legal system. After running 7 classification models, 5 regression models and feature selection, we found that categorical data is in fact able to predict case results with approximately 87% accuracy. We were able to use similar methods to model each Justice and obtain an average of 62% accuracy depending on the individual. We ran 5 different classification models on TF-IDF [10] features of written opinions and were able to achieve at best 54% accuracy in predicting a conservative vote based on the written opinion. Our analysis uncovered political leanings and biases in the system: in case documentation and text and in the Justices themselves. Our findings and similar research can be leveraged to potentially neutralize these very biases by informing the appointment of future Justices.

1 Introduction

The Supreme Court is the highest court of the country and determines some of the most contentious cases, but the inner workings of the US Court System are hidden to the public [12]. We would like to trust that decisions are predicated on a just system, yet with so many factors, even ones as important as the appointment of Justices, up to human bias, it is hard to be sure. We therefore set out to examine which factors most heavily influence Supreme Court decisions. We are motivated by the conviction that it is always possible to improve a system with explicit pain points and bottlenecks –identifying these is just the beginning.

2 Related Work

In their paper on judicial forecasting, Martin, et. al seek to distance themselves from legal experts by relying solely on statistical trends in a Court's behavior in order to predict future decisions[8]. They proposed a decision tree algorithm that was accurate yet inflexible to alterations in the court and owed its superhuman accuracy (though it performed worse for individual justice's predictions),

to the vast set of decisions in the previous 8 terms. Humans tend to rely on fewer, more-detailed observations. Ruger, et. al proved the effectiveness of these methods by predicting the outcomes of the 2002 term in real time, comparing the results of statistical method with those of human experts and obtaining accuracy rates which beat those of the panel[11]. These decision tree methods were adapted in [5] to develop a more flexible, online algorithm capable of making accurate predictions when Court composition changes, when a Justice cannot make it to a vote, and in other extraneous circumstances. In our project, we seek to develop accurate classification methods, expand on the data analysis to better understand statistical relationships between justices and cases, and to incorporate new textual data to develop novel ensemble prediction methods.

2.1 Approach

Rather than blindly follow existing literature, we sought to first better understand relationships in the data through clustering methods and visualization techniques. We then took three parallel approaches to the prediction and interpretation task through analysis of the case-centered, justice-centered, and text data.

3 Methods

3.1 Data

Washington University's Supreme Court Database [4] compiles data on Supreme Court cases over the period 1946-2016. A full list of the available features can be found in the codebook [4]. Additional documentation can also be found on our github wiki [3].

3.1.1 Case-Centered Data

The case-centered data consists primarily of categorical features and outcome variables. Each of the 8737 rows represents a case that passed through the Supreme Court. The SCDB Case ID ("caseId") serves as the primary key. This dataset provides 53 variables for each row, some background variables, some outcome, some chronological, and some identifying. We used 28 of these variables as feature variables and chose one outcome variable to predict. This dataset serves as a basis for training predictive models for the Court as a whole based on past cases. We sought to predict "partyWinning", which indicates whether or not the party that filed the case won. Hearings from January 1, 2015 and onwards were withheld as the test set, leaving the period 1946-2014 for training.

3.1.2 Justice-Centered Data

This dataset expands on the case-centered data –here, each of the 78,233 rows represents an individual Justice's vote on a case. The dataset includes additional features identifying the Justice and encoding the Justice's vote. The "voteId" column (combining "caseId" with an identifier for the justice's vote on the case) functions as the primary key. Outcomes are normalized to 8 categorical variables indicating agreement with majority, dissent, and varying levels and types of concurrence and dissent. In addition to this and other agreement variables, the dataset also encodes the vote as being in a liberal, conservative, or neutral "direction".

3.1.3 Text Data

The third dataset is from CourtListener.com bulk data API [2]. This dataset consists of 63,864 JSON objects corresponding to cases, 63,865 opinions that correspond to legal opinions written by people about cases, and 8,521 people. The JSON objects were loaded into a MongoDB database and people were labelled by the corresponding "justiceName" from the SCDB dataset. Opinions were matched to rows in the justice-centered dataset by "caseId" and the name of the justice who authored or co-authored the opinion, producing a dataset of 18,275 rows with columns "voteId" and "opinion". "voteId" is from the SCDB justice-centered dataset, and allows for joining on the SCDB dataset to fetch outcome variables like "vote", "majority", or "direction". The field "opinion" contains a plain text string representing the written legal opinion. After filtering out documents with empty "opinion", we finally have 5362 documents. These are converted into bag-of-words representation with a vocabulary of 1554 using a word count threshold of 1000, down from a raw vocabulary of 85,408 (i.e. we use 1.8% of the word count features). We assign the first 80% of the corpus to training and 20% to test, giving us 4289 rows in our training set (cases from 1946 to 1989) and 1073 rows in our test set (cases from 1989 to 2004).

3.2 Case-Centered Analysis

3.2.1 Classification Models

We compared the following classifiers: Random Forest, Decision Tree, K-Nearest Neighbors, Gaussian Naive Bayes, Bernoulli Naive Bayes, SVM (Gaussian Kernel), Logistic Regression

We chose Decision Tree based on our analysis of related work [8, 11, 5] and Random Forest as an ensemble classifier that improves on a single Decision Tree [7]. The other classification models are standard tools for data scientists, which we treated as a benchmark.

3.2.2 Regression Models

We used cross validation to compare the following models: Random Forest, Ridge Regression, Lasso Regression, Lasso Lars Regression, and Elastic Net.

3.2.3 Feature-Selection

Feature selection was performed to avoid over-fitting the training set and as a way of interpreting the most predictive elements about an individual case. A randomized lasso was used to do so, and features were selected using a modified LARS algorithm. We decided to have subsets of 1/2 of the features be selected at random for each of 400 trials to best mimic a statistical bootstrap. We then compared this to the features selected with scikit-learn's SelectKBest feature selection algorithm using the χ^2 values to determine dependencies[9]. This allowed us to select the most predictive features of the dataset.

3.3 Justice-Centered Analysis

We sought to characterize each justice in the dataset through data visualization, clustering, and supervised prediction algorithms. We studied correlations of the political inclinations of Justices' votes for each case and sought to smooth over unknown values and decision (since each Justice cannot possibly deliberate over every case in history) using K-means clustering, drawing unknown values from the cloud about the centroids of classes defined around various subsets of the feature vector space. The data was imputed similarly to the case-centered set, label-encoding and binarizing (i.e. converting to an indicator-variable vector) categorical variables and normalizing continuous data. We then sought to generate features and additional labels to boost model performance.

3.4 Text Analysis

3.4.1 Data Comprehension: χ^2 Feature-Selection

We performed χ^2 feature-selection to find the 100 features that capture most of the dependence by the outcome variables on word-count features. We then generated word clouds of these top features, in which the size of each word corresponds to its χ^2 score.

3.4.2 Classification

We attempted a naive approach at predicting two different outcome variables based on TF-IDF representations of written opinions. We were interested in answering two questions:

1. How well can we predict the political direction (liberal or conservative) of a justice's vote based on their written opinion?
2. Splitting on whether the justice's vote was liberal or conservative, how well can we predict whether the justice voted with the majority or in dissent based on their written opinion?

We tested the following classifiers on 1,554 TF-IDF features:

- SVC (RBF kernel)
- SVC (linear kernel)
- Gaussian Naive Bayes
- Multinomial Naive Bayes
- Random Forest (max depth of 40, minimum of 12 samples per split, and 10 estimators)

3.5 Evaluation

3.5.1 Accuracy, Precision, Recall, F1

We compared the performance of the classifiers with different groups of features using accuracy, precision, recall, and F1-score metrics. We define precision, recall, and F1 score in terms of false positives (FPs), false negatives (FNs), true positives (TPs), and true negatives (TNs):

$$precision = \frac{TP}{TP + FP}, recall = \frac{TP}{TP + FN}, F1 = 2\frac{precision * recall}{precision + recall}$$

3.5.2 ROC and AUC

ROC curves plot the false positive rate vs the true positive rate of various classifiers. the closer an ROC curve is to the "top-left corner," the greater the performance of that classifier. AUC is the area under an ROC curve, and classifiers with higher AUC values have greater performance.

3.5.3 RMSE, R^2

In order to evaluate the performance of the regression models, we used calculations of the RMSE (root mean squared error) and R^2. The RMSE is a measure of the differences in the predicted values and the training value and R^2 is a measure of how close the regression line is to the data.

4 Spotlight Model: Ridge Regression

Now we will consider the Ridge Regression Model (RR). In our analysis, we found that RR gave the lowest R^2 out of all our regression models.

The RR Model was developed by Andrey Tikhonov. While in statistics the model is referred to as a ridge regression, in machine learning contexts, it is referred to as weight decay. With regard to the development, Hoerl and Kennard (1970) noticed a fix for instability in the LS estimator β: [6]

$$\beta = (X'X)^{-1}X'Y$$

RR is most frequently used for ill-posed problems, where there is no solution or no unique solution for x in the equation $Ax = b$. The fix was adding a small constant value (λ) to the entries along the diagonal of $X'X$ to minimize the penalized sum of squares: $\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$. This gives the RR estimator β_{ridge}:

$$\beta_{ridge} = (X'X + \lambda I_p)^{-1}X'Y$$

Here λ is a preselected constant and λ*the squared norm of β is a penalty term, which is minimized when β_j is small. Something to note is that the residual sum of squares and penalty term are at odds with one another.

RR is primarily motivated by too many predictors (ie. when the number of input variables is too high). Fitting such a model without penalization gives way to large and unhelpful prediction intervals. It is also motivated by an X in $X'X$ that is singular or almost singular. In such cases, the least square estimator may fit the training data, but not the test data, well. Hence, the need for β_{ridge}.

One may test the efficacy of RR using scores of RMSE and R^2 as we have.

5 Results

5.1 Case-Centered Results

5.1.1 Classification and Regression

When using classification models to predict case outcome, we found that Random Forests resulted in the highest scores, with an accuracy of 0.875, precision of 0.876528, recall of 0.876, and f1 score of0.875585. This is visualized in our graph of the ROC curves, where RNB (the Random Forest Classifier) has the highest AUC.

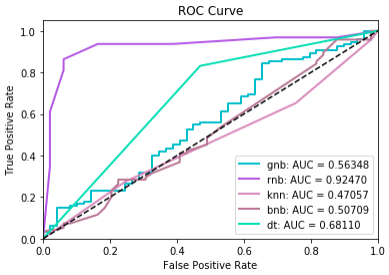

Figure 1: ROC Curve

When using regression models, Lasso, with an alpha of 0.5, performed the best with a root mean squared error of 0.223 and R^2 of 0.007. We utilized parameter tuning via cross-validation to identity the optimal alpha, running scripts with alpha values ranging from .01 to 1.

5.1.2 Feature Selection

Our methods led us to select the following 10 features as being the most predictive and useful for our model. Some features such as issue and issue area indicate that the content of the cases do have significant influence over the case rulings. Further, we noticed that surprisingly, both respondent state and case origin state are more predictive than the petitioner's state.

- issue
- issue area
- supporting law
- respondent state
- administrative action
- case origin
- case source (circuit)
- case source state
- certification reason
- lower court disposition

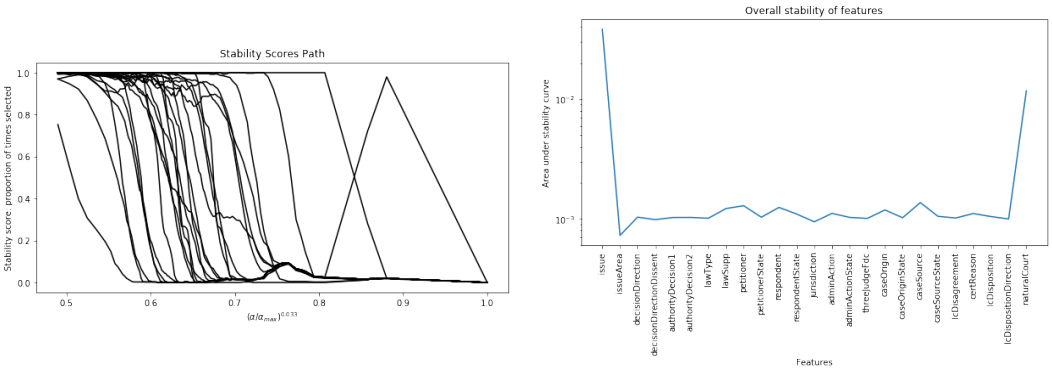

Figure 2: Lasso stability paths (left) were plotted for each feature. The iteration at which a feature is dropped will lower for less "important" features, smoothed by testing over a large number of randomly selected subsets of the feature vector. The relative importance may also be visualized by looking at the area under the curve (right)

5.2 Justice-Centered Results

5.2.1 Data Comprehension: Clustering

In order to lend to the interpretability of later classification tasks and elucidate a manner by which one might make predictions in a low data regime, we sough to characterize the justice's voting habits using unsupervised clustering techniques. We were inspired by the concept of Hebbian learning and sought to to cluster justices based on their agreements and contradictions in votes. We generated a matrix M of the political bent of the vote of each justice for every case in the dataset, imputing values of 0 for the cases which took place while the justices weren't in office. We then generated the covariance matrix (figure 3) to highlight these temporal and sometimes antagonistic relationships.

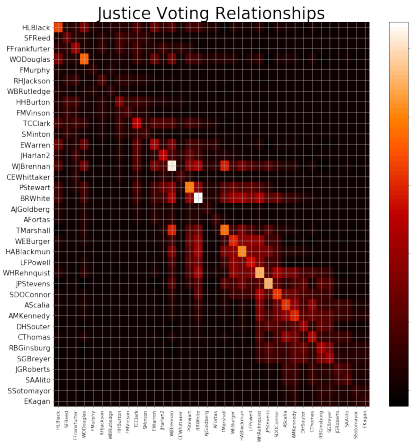

Figure 3: Voting Covariances.

We then sought to further qualify these voting habits by performing K-means clustering on subsets of the feature (or training label) vector of interest to us. The justice-centered dataset identifies 14 different issue areas (as well as our 15th imputed label for nulls), including areas like unions, first amendment, interstate relations, and criminal procedures. We calculated the average political inclination along each dimension of this issue vector to seek to add nuance to our study of bias.

We incremented k in an attempt to adequately capture the variance in the dataset and eventually settled on $k = 4$ clusters. Figure 5.2.1 shows these clusters projected on the first two principal components of the issue area vector. One may then qualitatively describe justices based on the location of the centroid of their corresponding cluster in the original issue vector space. For instance, justices with label 2 tend to be much more conservative than the other three. Label 1 Justices tend to be liberal in most ways except when it comes to cases dealing with Judicial Power. Similarly, Justices with label 3 are liberal except on issues related to privacy, in which they (as a group) tend to take a staunchly conservative stance. Note that similar clustering in theory could be done on the more precise "issue" feature, but the 384 unique values would induce a sparsity that would rend any clustering attempts moot.

A third analytic approach we took was to divide justices based on the frequency with which they dissented from the majority stance. This was also done by using the k-means algorithm with appropriate nationalizations. This variable is of interest since a strong bias one way or the other could

allow us to maximize information gained from other justice's models in predicting on a certain case. The frequencies are plotted in figure 4.

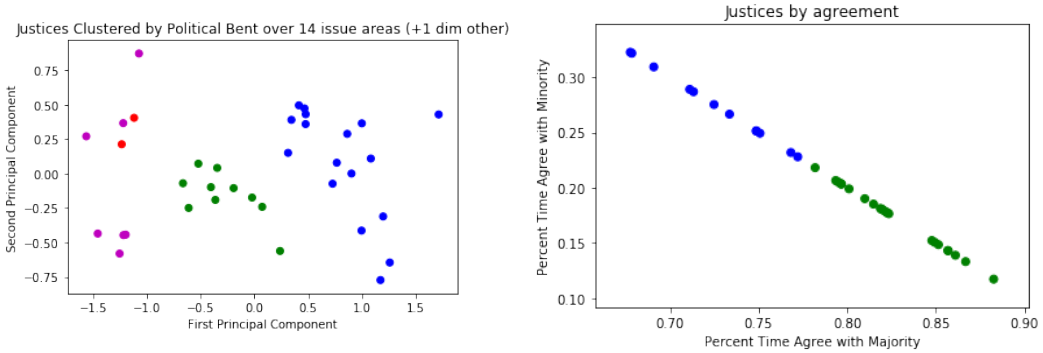

Figure 4: Clusters on average bias over a dim-15 vector issue space are shown projected on their first two principal components (left). Justices were also clustered based on their likelihood to side with the majority.

The frequency of siding with the majority (in the training set) and issue-based bias vectors (smoothed by averaging with the label centroid to infer stance on cases a judge perhaps hadn't seen prior to a test case) were used as generated features for later classification tasks.

5.2.2 Political Drift

Votes were encoded as +1, 0, -1 for conservative stance, no/ambiguous, and liberal stance respectively. The cumulative sum over the justices lifetime amounts to a biased random walk. Figure 5 shows that moderates are the exception, those with strong bias tend to stay in office longer, and bias tends to be homogeneous over the Justice's lifetime. Individual biases generally balance out, and the cumulative bias of the court at any time may be viewed as a Gaussian process (similar to universality). We found the accuracy of such models empirically to be inferior to decision-tree models; the number of interacting "random walks" is small, so the composition of the court strongly influences the group's decisions. The small number of Justices on any given court prompted us to focus on developing Justice-centered models which allow for greater flexibility during times of Court transition, etc.

5.2.3 Classification

When allowed information on the direction of the ruling of the court in general, justice-centered models obtain 85%+ accuracy for most of the justices (boosted by our generated majMin feature to encode the probability that a Justice will side with the court) . However the more interesting case involves using only features that would be available in real-time. Eliminating all variables relating to the Court's deliberations, we predict individual Justice's decisions based on regional, temporal, and case's origins. The accuracies varied greatly for each Justice but were much better

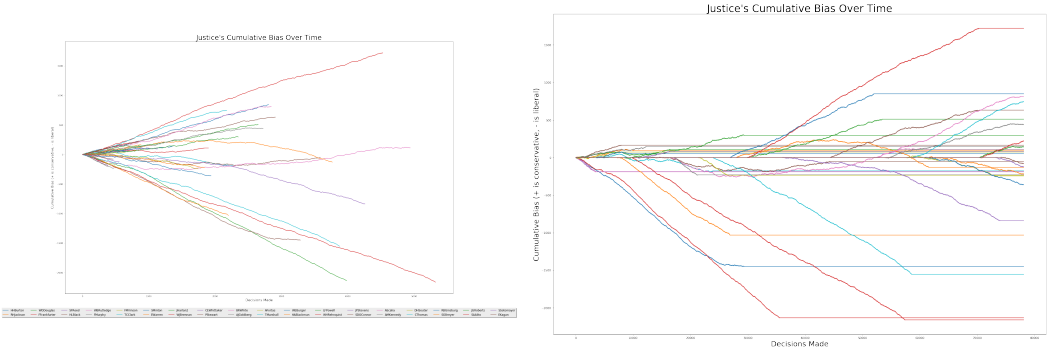

Figure 5: As a whole, Justices tend to maintain a similar level of political bias throughout the extent of their career. A notable exception is Blackmun, who begins his career as a staunch conservative, but reversing paths mid-career, around the time of the New Deal. This is shown over the life of the court (in the dataset) at right.

than a naïve accuracy of 0.33 (allowing for absences), and 0.5 (requiring the justice to take a side). Of all methods attempted, a random forests classifier (with 100 trees) demonstrated the highest classification accuracy on the test set. Below is a summary of results for three Justices. See tables 4 and 5 in the appendix for more information.

Table 1: Summary statistics for the accuracy (left) and f1 scores (right) for Justice models.

	Features	Features+	Only Gen
Average	0.6229219656	0.6402002137	0.5820925116
Variance	0.0079013471	0.0066996689	0.008525549
Stdev	0.0888895221	0.0818515054	0.0923338998

	Features	Features+	Only Gen
Average	0.6454189358	0.6502942025	0.6879103688
Variance	0.0089229093	0.0073264649	0.013311615
Stdev	0.0944611524	0.0855947714	0.1153759722

5.3 Text Analysis Results

5.3.1 Word Clouds

Figure 6: On the left: words that best predict whether they voted with the majority or in dissent. On the right: words that best predict whether the justices leaned conservative or liberal on their vote.

The word cloud on the left in 6 shows us the words that best predict whether justices voted with the majority or in dissent. The most prominent stems include ”racial”, ”minor”, and ”miner”, suggesting that these were issues with a clear majority.

The word cloud on the right in 6 shows us the words that best predict whether the justices leaned conservative or liberal. Most prominent are stems like ”pecial”, ”boundari”, ”execut”, ”olorado”, ”resid”, and ”divorc”, suggesting that issues regarding special boundaries, execution, divorce, and Colorado may be divisive.

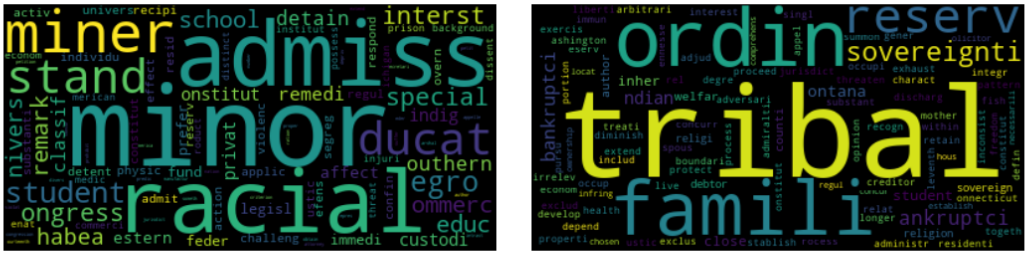

Figure 7: On the left: words that best predicted whether justices voted with the majority when they leaned liberal. On the right: words that best predicted whether justices voted with the majority when they leaned conservative.

In 7, we drill down to see which stems were most predictive of whether justices voted with the majority or in dissent when they leaned conservative or liberal. The most prominent stems when justices leaned liberal include ”minor”, ”racial”, ”admiss”, ”miner”, ”ducat”, ”stand”, and ”egro”. This suggests that educational access for minorities was of special interest to justices when they voted in a liberal direction.

The most prominent stems when justices voted in a conservative direction include ”tribal”, ”famili”, ”ordin”, ”reserv”, and ”sovereignti”, suggesting that perhaps Native American issues were of special interest to justices when they voted conservatively.

Raw rankings by χ^2 score are available in the appendix in table 3.

5.3.2 Classification

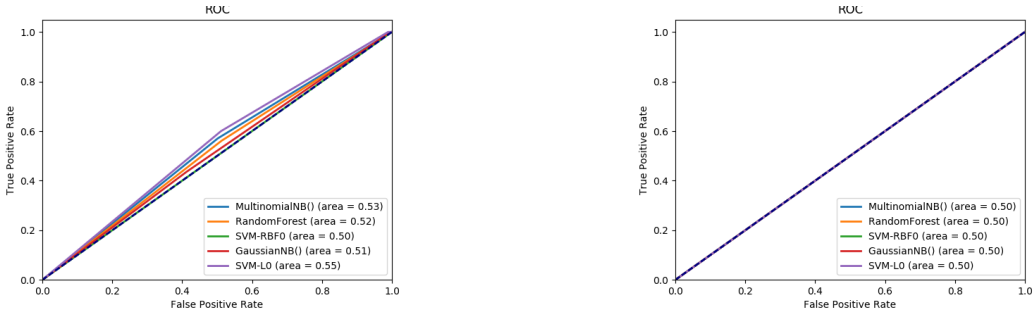

Figure 8: ROC curves for several classifiers on TF-IDF features. Left: predicting whether justices voted in a liberal direction. Right: predicting whether justices voted with the majority

Classification on the full set of TF-IDF features was unsuccessful, as evidenced by ROC curves that show performance on par with a coin toss. It is interesting to note that even with naive models with default parameters, we can predict political direction better than we can the justice's vote. In 9 (in appendix) we even see Gaussian Naive Bayes under-performing a coin toss. This is a sign of over-fitting, revealing the importance of randomizing the selection of samples for the train and test sets. This likely happened due to a natural shift in issues of importance to liberal justices over time.

6 Discussion and Conclusion

We performed exploratory data analysis on justices' written opinions, justices' votes on specific court cases, and on the court cases themselves. We found that we were able to predict the outcomes of cases quite accurately, and that the most predictive features of each case were the issue, the laws used to support the argument, and the case's originating circuit and state. We found that justices tend to maintain their political directions over time and visualized the differences in words they used in their written decisions when defending liberal or conservative votes.

These findings are quite unnerving and suggest that there may be more at play than a fair system of checks and balances. Given that our findings showed human bias influencing the judicial system, we propose that the insights be further investigated and incorporated to maintain a balanced selection of Justices moving forward. Such a move would potentially prevent presidents and others from heavily influencing serious situations at hand.

In the future, we hope to improve the flexibility of a combined model of case, justice, and text data by streamlining the data collection process and using meta-classifiers learn structure in using the individually trained components. Moving forward, we would like to generate features through LDA topic modeling [1] on the text data and incorporate these into the court-centered and justice-centered datasets to see how they fare in predicting the direction and vote of justices. We could also use these topic features to visualize the shift in "liberal" and "conservative" topics over time.

On a Justice level, we would like to aggregate biographical information about each individual to better plot how demographic, geographic, and educational characteristics introduce additional bias. This could even expand to incorporate text data from Justice's previous works or court proceedings, creating semantic "snapshots" of the Justice throughout his or her career (thereby preserving key court cases and movements in the process). Collecting such data would effectively enable us to identify where the largest biases stem from, and ideally allow us to propose tactics for improving the system to be more fair.

References

[1] David Blei, Andrew Ng, and Michael Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:4–5, 2003.

[2] Brian W. Carver and Michael Lissner. Court listener. https://www.courtlistener.com/. Accessed: 2017-05-12.

[3] Maia Ezratty, William Hinthorn, Mihika Kapoor, and Alice Zheng. Supreme predictions. https://github.com/hinthornw/supreme_predictions/, 2017.

[4] Andrew D. Martin Jeffrey A. Segal Theodore J. Ruger Harold J. Spaeth, Lee Epstein and Sara C. Benesh. 2016 supreme court database, version 2016 release 01. http://Supremecourtdatabase.org/. Accessed: 2017-04-22.

[5] Oliver Wendell Holmes Jr. *The path of the law*. The Floating Press, 2009.

[6] Jia Li. Ridge regression. https://onlinecourses.science.psu.edu/stat857/node/155. Accessed: 2017-05-15.

[7] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[8] Andrew D Martin, Kevin M Quinn, Theodore W Ruger, and Pauline T Kim. Competing approaches to predicting supreme court decision making. *Perspectives on Politics*, 2(04):761–767, 2004.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, 2003.

[11] Theodore W Ruger, Pauline T Kim, Andrew D Martin, and Kevin M Quinn. The supreme court forecasting project: Legal and political science approaches to predicting supreme court decisionmaking. *Columbia Law Review*, pages 1150–1210, 2004.

[12] Harold J. Spaeth, Lee Epstein, et al. Supreme court of the united states. https://www.supremecourt.gov/. Accessed: 2017-05-1.

7 Appendix

	Classifier Evaluation Without Feature Selection			
Classifier	Accuracy	Prec	Recall	F1
RF	0.819444444444	0.826884008626	0.819444444444	0.821708937198
KNN	0.513888888889	0.503900112233	0.513888888889	0.508557554045
GNB	0.625	0.635478670635	0.625	0.608994276988
BNB	0.638888888889	0.433531746032	0.638888888889	0.516548463357
DT	0.743055555556	0.73500267094	0.743055555556	0.735874284029

Table 2: **Results for Different Classifiers on Test Data**: Accuracy, Precision, Recall, and F1 scores for the different classifiers. Results with only the first four features on the left and with adding KMeans features on the right

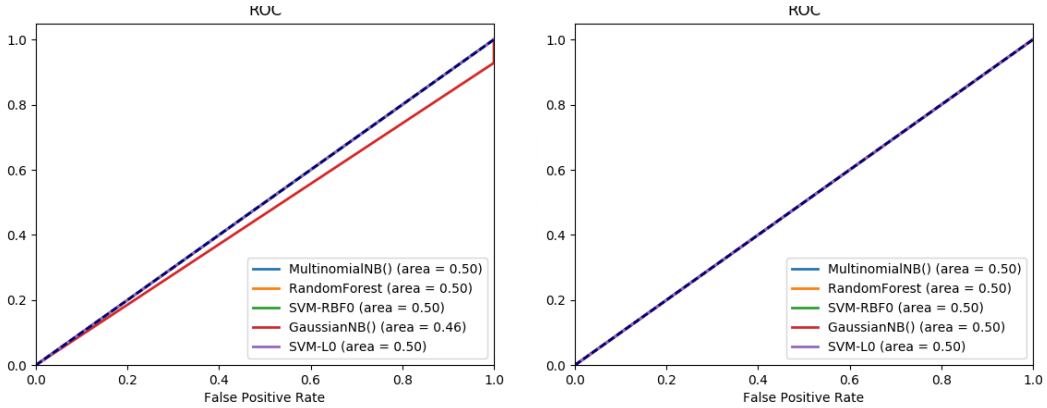

Figure 9: ROC Curves for several classifiers on TF-IDF features. Predicting whether justices voted with the majority when they voted in a liberal (left) or conservative (right) direction.

Table 3: Top Chi2 scores for bag-of-word features

majority		direction		liberal		conservative	
minor	3326	pecial	24795	minor	4601	tribal	7000
racial	2766	boundari	12739	admiss	3642	ordin	6423
miner	2665	olorado	10498	racial	3428	famili	5639
tribal	2350	rticl	5067	miner	2821	reserv	3952
student	2326	execut	4642	stand	1707	sovereignti	3745
admiss	2325	xecut	3651	ducat	1623	ankruptci	1746
reserv	2199	ebraska	3629	student	1445	bankruptci	1554
famili	1830	resid	3621	egro	1280	ndian	1370
ducat	1397	divorc	3587	ongress	1059	ontana	1124
ordin	1273	decre	3362	special	1022	student	990

Table 4: Model accuracies (left three) and precisions (right three) for each justice using a random forest model with 100 trees using feature vectors that were provided (base), the generated features (OnlyGen) and both (+Gen).

	Base	+Gen	OnlyGen	Base	+Gen	OnlyGen
HHBurton	0.3982	0.4867	0.5133	0.4653	0.5298	0.7252
RHJackson	0.4839	0.5161	0.5161	0.6028	0.6379	0.9389
WODouglas	0.7363	0.7582	0.7308	0.8715	0.8508	1.0000
FFrankfurter	0.5886	0.6266	0.4241	0.6296	0.6590	1.0000
SFReed	0.5326	0.5435	0.5109	0.5603	0.5766	0.5601
HLBlack	0.7109	0.7517	0.6769	0.8307	0.8330	1.0000
WBRutledge	0.7407	0.7407	0.7407	0.8222	0.8222	1.0000
FMurphy	0.7407	0.7407	0.7407	0.8920	0.8068	1.0000
FMVinson	0.4545	0.6364	0.4545	0.4705	0.6533	0.6013
TCClark	0.6296	0.6250	0.5046	0.6379	0.6272	0.5673
SMinton	0.5806	0.5484	0.5484	0.5806	0.5729	0.6640
EWarren	0.7373	0.7604	0.6959	0.8513	0.8435	1.0000
JHarlan2	0.5804	0.5804	0.5045	0.6687	0.6679	1.0000
WJBrennan	0.7110	0.7319	0.6559	0.8279	0.8130	1.0000
CEWhittaker	0.5741	0.5741	0.5370	0.7042	0.7042	0.7968
PStewart	0.6264	0.6207	0.5172	0.6313	0.6284	0.5428
BRWhite	0.6320	0.6720	0.4600	0.6396	0.6733	0.5102
AJGoldberg	0.7600	0.8000	0.7200	0.7742	0.8627	0.8100
AFortas	0.7167	0.7500	0.6667	0.8399	0.8072	1.0000
TMarshall	0.7109	0.7214	0.6458	0.8399	0.8279	1.0000
WEBurger	0.7038	0.7213	0.6341	0.7703	0.7598	1.0000
HABlackmun	0.6203	0.6230	0.4706	0.6369	0.6420	0.7125
LFPowell	0.5616	0.6087	0.5254	0.6215	0.6741	1.0000
WHRehnquist	0.7052	0.6984	0.6848	0.8235	0.7779	1.0000
JPStevens	0.5783	0.5904	0.5687	0.6394	0.6456	1.0000
SDOConnor	0.6893	0.6464	0.6357	0.7140	0.6651	1.0000
AScalia	0.6274	0.6426	0.6540	0.6546	0.6651	1.0000
AMKennedy	0.5681	0.5953	0.6031	0.5964	0.6137	1.0000
DHSouter	0.5806	0.6323	0.5355	0.6194	0.6706	0.9002
CThomas	0.6634	0.6535	0.6782	0.7179	0.7431	1.0000
RBGinsburg	0.5792	0.5792	0.5464	0.6483	0.6542	1.0000
SGBreyer	0.6047	0.5814	0.5058	0.6240	0.6289	1.0000
JGRoberts	0.6413	0.6304	0.5326	0.6413	0.6286	0.6977
SAAlito	0.6364	0.6250	0.6364	0.6720	0.6552	0.8486
SSotomayor	0.5000	0.5345	0.5000	0.6395	0.7111	0.7377
EKagan	0.5200	0.5000	0.4800	0.7604	0.6921	1.0000

Table 5: The recall and f1 scores for each model are displayed below.

	Base	+Gen	OnlyGen	Base	+Gen	OnlyGen
HHBurton	0.3982	0.4867	0.5133	0.4202	0.4988	0.5739
RHJackson	0.4839	0.5161	0.5161	0.5227	0.5510	0.6548
WODouglas	0.7363	0.7582	0.7308	0.7874	0.7917	0.8444
FFrankfurter	0.5886	0.6266	0.4241	0.5942	0.6294	0.5956
SFReed	0.5326	0.5435	0.5109	0.5389	0.5509	0.5244
HLBlack	0.7109	0.7517	0.6769	0.7524	0.7781	0.8073
WBRutledge	0.7407	0.7407	0.7407	0.7749	0.7749	0.8511
FMurphy	0.7407	0.7407	0.7407	0.8004	0.7634	0.8511
FMVinson	0.4545	0.6364	0.4545	0.4563	0.6388	0.4973
TCClark	0.6296	0.6250	0.5046	0.6310	0.6252	0.5000
SMinton	0.5806	0.5484	0.5484	0.5806	0.5556	0.5807
EWarren	0.7373	0.7604	0.6959	0.7760	0.7877	0.8207
JHarlan2	0.5804	0.5804	0.5045	0.6017	0.6016	0.6706
WJBrennan	0.7110	0.7319	0.6559	0.7482	0.7568	0.7922
CEWhittaker	0.5741	0.5741	0.5370	0.6104	0.6104	0.5992
PStewart	0.6264	0.6207	0.5172	0.6263	0.6210	0.5287
BRWhite	0.6320	0.6720	0.4600	0.6332	0.0000	0.4600
AJGoldberg	0.7600	0.8000	0.7200	0.7651	0.8000	0.7200
AFortas	0.7167	0.7500	0.6667	0.7589	0.7687	0.8000
TMarshall	0.7109	0.7214	0.6458	0.7518	0.7554	0.7848
WEBurger	0.7038	0.7213	0.6341	0.7243	0.7333	0.7761
HABlackmun	0.6203	0.6230	0.4706	0.6219	0.6254	0.5353
LFPowell	0.5616	0.6087	0.5254	0.5793	0.6289	0.6888
WHRehnquist	0.7052	0.6984	0.6848	0.7460	0.7266	0.8129
JPStevens	0.5783	0.5904	0.5687	0.5971	0.6070	0.7250
SDOConnor	0.6893	0.6464	0.6357	0.6971	0.6524	0.7773
AScalia	0.6274	0.6426	0.6540	0.6382	0.6515	0.7908
AMKennedy	0.5681	0.5953	0.6031	0.5788	0.6018	0.7524
DHSouter	0.5806	0.6323	0.5355	0.5912	0.6418	0.6519
CThomas	0.6634	0.6535	0.6782	0.6845	0.6877	0.8083
RBGinsburg	0.5792	0.5792	0.5464	0.5989	0.6010	0.7067
SGBreyer	0.6047	0.5814	0.5058	0.6101	0.5932	0.6718
JGRoberts	0.6413	0.6304	0.5326	0.6413	0.6288	0.5894
SAAlito	0.6364	0.6250	0.6364	0.6522	0.6388	0.7190
SSotomayor	0.5000	0.5345	0.5000	0.5538	0.5996	0.5000
EKagan	0.5200	0.5000	0.4800	0.5898	0.5000	0.4800