What Matters: Agreement between U.S. Courts of Appeals Judges

Daniel L. Chen
Toulouse School of Economics
daniel.chen@iast.fr

Xing Cui
Center for Data Science
New York University
xing.cui@nyu.edu

Lanyu Shang
Center for Data Science
New York University
lanyu.shang@nyu.edu

Junchao Zheng
Center for Data Science
New York University
junchao.zheng@nyu.edu

Abstract

Federal courts are a mainstay of the justice system in the United States. In this study, we analyze 387,898 cases from U.S. Courts of Appeals, where judges are randomly assigned to panels of three. We predict which judge dissents against co-panelists and analyze the dominant features that predict such dissent with a particular attention to the biographical features that judges share. Random forest, a method developed in Breiman (2001), achieves the best classification. Dissent is roughly half-driven by case features and half-driven by personal features.

1 Introduction

Using the universe of U.S. Courts of Appeals cases since 1880, we build models predicting agreement among judges where we include a random component—the composition of the panel of judges assigned to the cases—among the feature set. The random composition implies a causal interpretation and the feature weights offer a natural metric to evaluate the importance of the causal effects relative to other factors exogenous and endogenous to the final decision. We focus particularly on the biographical features that judges share, as these shared biographies may proxy for a shared perspective, life experience or empathy, or network that drives the decision to agree.

1.1 U.S. Courts of Appeals

There are three levels of federal courts: U.S. Supreme Court (1 Court); U.S. Courts of Appeals (13 Circuits including 12 Regional and one for the Federal Circuit); and U.S. District Courts. The middle level is also called Federal Circuit Courts. They are not trial courts and do not hear cases first. They hear cases that have been appealed from federal district courts, as well as appeals of decisions of federal agencies. Federal Circuit judges are nominated by the President and confirmed by the Senate and they are appointed for life. Each Circuit has between eight to 40 judges available to be assigned out of a pool. Typically three are randomly selected to hear each case. Each judge has a different background such as their birth state, education, party, war experience, and other personal information. Currently, there are 179 judges.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

1.2 What We Analyze

Federal Circuit courts are the intermediate appellate courts of the federal court system. It is not like the Supreme Court, which hears less than one hundred cases a year, and is also not like District Courts, which hear more than three hundred thousand cases a year. In addition, since cases heard in Circuit Courts are appealed from lower courts, there are no witnesses and evidence presented in court. The Circuit judges review the records from the original trial, accept written arguments, and sometimes hear oral arguments from the lawyers for each side. When judges write an opinion, they justify their ruling in a case. Because one of the three judges ruling on a case can dissent, our target variable is whether a judge has joined the opinion or disagreed, which results in a separate opinion. Therefore, the Circuit Courts are affirming or reversing the lower court decisions. As such, the issues and problems addressed are extremely serious. We want to investigate if any bias exists in some features of judges (who are randomly assigned) that affect the production of justice.

2 Related Work

2.1 Machine Learning And Legal Study

Several analyses of judicial behavior have been proposed to investigate the effect of ideological diversity on judicial decision making. It has been argued that, in the federal courts of appeals and the U.S. Supreme Court, the dissent rate is positively related to ideological differences [1]. In addition, a realistic conception of judges' incentives predicts "dissent aversion" in the circumstances prevailing in the courts of appeals, for example, the costs of writing a separate dissent in terms of time and collegiality [2] [3]. Prior work has identified that factors such as judical ideology and group polarization predict agreement or disagreement [4] [5].

Machine learning is one of many statistical techniques already widely used within empirical studies in law [6]. In general, predictive analytics approaches use advanced computer algorithms to scan large amounts of data to detect patterns, which can be used to make predictions about never-before-seen future data. For example, one might be able to predict reversal to aid lawyers in the decision to file an appeal (an expensive ordeal) or to aid judges in writing decisions unlikely to be reversed.

2.2 Acknowledgement of Previous Work

Machine learning, one of the most popular and powerful data science techniques, has been applied to predict the behavior of Supreme Court of the United States [7]. The reason data scientists have applied machine learning is because of the perception that machine learning can provide a generic, robust, and fully predictive method. The approach commonly used for binary classification problems is the classification and regression trees (CART) methods first offered in the work of Breiman [8]. Katz, et al. (2014) used court and justice level information, case information, and historical justice and court information as the input data to predict behavior such as affirm or reverse [7]. However, as panels of judges are not randomly assigned in the Supreme Court, Katz, et al. (2014) did not explicitly consider judge similarity.

3 Data Sets Description

The original data was collected by one of the authors and has been used in [9], [10], [11], [12], and [13]. It contains two files: one is on the case level and the other is on the vote level. The case level data set contains 387,898 case records since 1880 and on and describes general information of each case with 134 features. The vote level data set contains 1,163,694 vote records and 414 features. Every three vote records describe the same case but from different judges on the panel. As the vote level data set already contains information in the case level data set, we focused mainly on the vote level data set.

4 Data Processing

4.1 Data Cleaning

Continuous-valued features with missing values were filled in with the mean value. Categorical features are transformed to a set of binary data (0 and 1), and missing values have been filled with a unique value, like -1. Features that contain mostly missing values are dropped. Moreover, we use Min-Max[1], a normalization that transforms a value to avoid having features that would be over-weighted or underweighted in the model. As a diagnostic baseline, we select 51 features to prepare the training data set.

4.2 Feature Categorization

We begin by allocating all features into two categories: general information about the case and information about the judge.

- Case information: e.g., whether the case is criminal, year of the case, etc.
- Judge information: e.g., political party of appointment, education, etc.

4.3 Target Preparation

We take whether two judges in one court agree or disagree with each other as the target: 1 indicates agreement and 0 indicates disagreement. On any case, judges have a few possible actions if they disagree with the author of the verdict. They can dissent, which is a disagreement as to the verdict, and they can concur, which is a disagreement as to the reasoning behind the verdict. Both require the writing of a separate minority opinion.

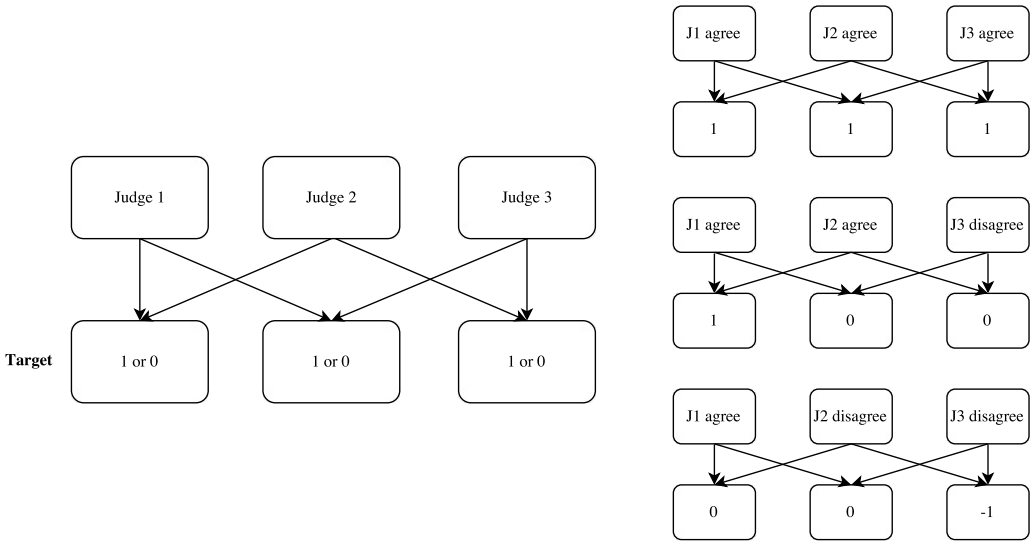


Figure 1: Example of Target Construction

To construct the target, we create a pairwise measure of agreement between every pair of judges on the three-judge panel. There are three main scenarios. In the first scenario, if all three judges agree, the target variable is [1,1,1]. In the second scenario, if one judge dissents or concurs, which is typically judge3, the target variable is [1,0,0]. In this instance, judge1 agrees with judge2 but disagrees with judge3. In the third scenario, judge2 concurs and judge3 dissents. In this instance, our target variable is [0,0,-1], where -1 represents a lack of information as to whether judge2 and judge3 agree or disagree. We selected pairwise judge records with non-negative target to train binary classification models. In other words, we reshaped the data to analyze pairwise judge agreement, and we dropped pairs that involved one judge concurring and the other judge dissenting.

[1] Min-Max: http:scikit-learn.orgstablemodulesgeneratedsklearn.preprocessing.MinMaxScaler.html

4.4 Feature Construction

We construct features involving the judges' social backgrounds. For instance, according to two judges' educational backgrounds, we generate a dummy indicator for whether they attended the same school. We include separate dummy indicators for shared characteristics including age group, political party of appointment, gender, race or ethnicity, state of residence, and school. We also included features such as whether they sat together during the last 3 months, the last 6 months, the last year, and the period since appointment. We also included a feature that is the historical rate in which a judge disagreed with the other judge. Our final step merged the original vote level data to analyze the agreement between any two pairs of judges, thus there were 868,962 entries and 98 features.

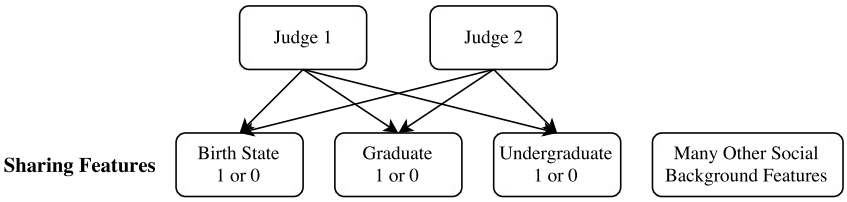


Figure 2: Example of Social Background between Two Judges

5 Predictive Models

5.1 Random Forest

Our first set of models employed random forest, which consists of a collection or ensemble of simple tree predictors, each of which are capable of producing a response when presented with a set of predictor values. Random forests can be thought of as an example of model averaging. The prediction is an aggregation of hundreds or thousands or distinct trees. Bagged (bootstrap aggregated[2]) decision trees, modified to reduce the correlation between trees, reduce the variance of the prediction. Random forests can be analogized to k-nearest-neighbor algorithms[3] or kernel regression[4], where the prediction for each point is a weighted average of nearby points, since the underlying trees are making predictions based on the simple average of nearby points equally weighted. We consider three sets of models: random forest with all features (RF), random forest with only case features (RF case), and random forest using only judge features (RF judge). We run these models to assess the degree to which case and judge information are significant to the prediction of agreement.

5.2 Logistic Regression

Since RF can only give importance for each feature, it cannot provide any information whether the feature is positively or negatively correlated to the final decision making. Therefore, we also introduce Logistic Regression. Logistic Regression(LR L1 and LR L2) is a useful classification algorithm for binary classification problems with high dimensional data. LR scales the output of linear regression to the range [0,1] and it indicates the confidence of our prediction. We implemented two penalty functions, the $l1\text{-}norm$ and the $l2\text{-}norm$. We also chose a threshold S for classification. Any input cases with a regression result larger than S, we categorize as class 1, otherwise as class 0. With the probability and the threshold, we can deal with unbalanced data by

[2]Bootstrap aggregating, also called bagging, is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid overfitting.

[3]The k-Nearest Neighbors algorithm (or k-NN for short) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space.

[4]Kernel regression is a non-parametric technique in statistics to estimate the conditional expectation of a random variable. The objective is to find a non-linear relation between a pair of random variables X and Y.

fitting a ROC[5] curve and adjusting the threshold[6] to get the optimal false positive rate (FPR) and true positive rate (TPR).

F-test, a statistical test in which the test statistic has an F-distribution under the null hypothesis, is widely used in the test for significance of variables in multiple linear regression. In our approach, however, we leverage the F-test to Logistic Regression, in a mere end-to-end fashion, to obtain the appropriate intermediate representations from features to judges' final decisions.

The Logistic Regression maps a value in the space of real number to an interval [0,1]. The architecture of F-test in Logistic Regression is similar to F-test in linear regression while we only use it to reveal rankings of features based on its significance: (i) Fitting a Logistic Regression on all features and calculating the standard deviation between the true value and the predicted result(std_{all}). (ii) For each feature, fitting a Logistic Regression on all other features and recalculating the standard deviation between the real-value and the new predicted result (std_1). Then we can calculate the F-value as:

$$f(1, n-m-1) = \frac{\frac{std_{all}-std_1}{std_{all}}}{n-m-1},$$

where n is the number of observations and m is the number of features (degrees of freedom). Features with large F-value are interpreted as significant for the target prediction, while features with F-value below some threshold are insignificant. We only focus on the importance ranking instead of significant threshold.

5.3 Confusion Matrix and Model Selection

The confusion matrix, also known as error matrix, is a table layout that allows visualization of the performance of a chosen algorithm. In a confusion matrix, there are four values: true positive (TP), false positive (FP), false negative (FN), true negative (TN). From the confusion matrix, we can calculate the True Positive Rate (TPR) and False Positive Rate (FPR) as:

$$TPR = \frac{TP}{TP+FN},$$

$$FPR = \frac{FP}{FP+TN}.$$

We construct five confusion matrices, one for each model described above. A model is evaluated as performing well when it has a high TPR and low FPR. We find that random forest with all features performs best.

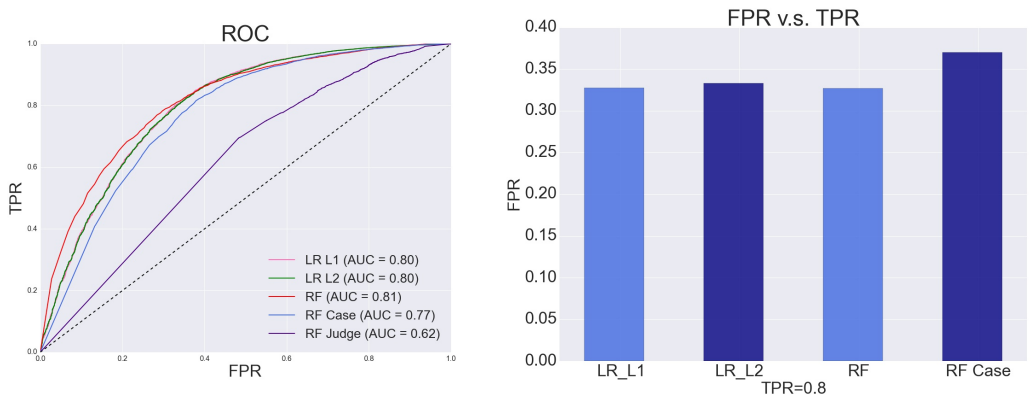
6 Results

6.1 Result of Models

Figure 3a shows the ROC curves of different models, and RF performs the best. Note that we care more about disagreement. The reason is that the majority of cases are decided unanimously, so we want to find out particular factors that drive judges dissenting or concurring. However, we can always adjust the threshold according to the practical use case. In our model selection, when we fix TPR to 80%, we have a lowest FPR = 32.81% with RF, as shown in Figure 3b and Table 1.

[5]In statistics, a receiver operating characteristic (ROC), or ROC curve, is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity, recall or probability of detection in machine learning. The false-positive rate is also known as the fall-out or probability of false alarm and can be calculated as (1 - specificity). The ROC curve is thus the sensitivity as a function of fall-out.

[6]The Threshold or Cut-off represents in a binary classification the probability that the prediction is true. It represents the tradeoff between false positives and false negatives.



(a) ROC Curve

(b) False Positive Rate(TPR = 0.8)

Figure 3: Results

Next, we analyze which features are important. We have also done supplementary analysis to determine to what extent both judge and case level features are necessary and possibly interactive in predicting the outcomes.

Table 1: **Prediction of Each Model**

Model Name	TN	FP	FN	TP
Logistic Regression(L1)	1558	759	8222	32910
Logistic Regression(L2)	1545	772	8217	32915
Random Forest(all)	1559	758	8006	33126
Random Forest(case)	1459	858	8112	33020

6.2 Result of Features Importance

Figure 4 shows the importance of case and judge features from RF model. We find that 18 of top 20 features are case characteristics. The remaining two are prior rates at which the two judges disagree with one another and whether they have sat together in the last three months. Note that regardless of continuous-valued features such as self-certainty word count, receiving disproportionate weight, the binary indicators still appear with some importance relative to the continuous ones. Table 2 reports the total feature importance of each category. Case related features occupy about 49.5% of total feature importance while judge related features occupy about 37.2% of total feature importance. The rest weights are for sharing features.

We have applied F-test to Logistic Regression, and the result in Table 3 shows the ranking of significant features. Empirically, the ranking of significance is similar to what is observed in Figure 4 (i.e. case features are still more predictive than judges' biographical features), which further suggests that the results from the F-test are reasonable.

Table 2: **Sum of Feature Importance from Random Forest** *

Feature Name	Weight
Case Information	0.495146
Judge Information	0.372468
Sharing Information	0.132385
Total	1.0

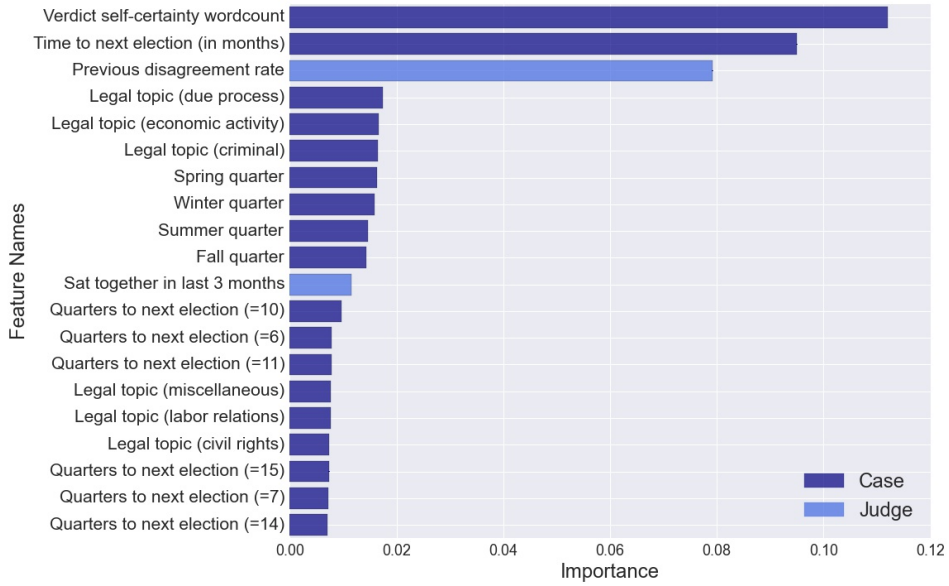


Figure 4: Important Features from Random Forest

Table 3: **Significant Features from F-Test** *

Case Infomation Features	**Significance**
NOT ASCERTAINED CASE	290.7743
DUE PROCESS CASE	20.7456
LABOR RELATIONS CASE	15.5400
ECONOMIC ACTIVITY CASE	5.6671
FIRST AMENDMENT CASE	3.2652
Judge Infomation Features	
JUDGE1 HDEM	58.2198
JUDGE2 HDEM	55.4523
JUDGE2 DISTRICT	15.0088
JUDGE1 HREP	14.1001
JUDGE2 SDEM	13.1935
Sharing Infomation Features	
JUDGES NOT SHARING PARTY	30.4551
JUDGES SHARING GENDER	21.4298
JUDGES SHARING PARTY OF PRESIDENT	17.8126
JUDGES SHARING RACE	8.3888
JUDGES SHARING BIRTH YEAR	6.2020
Other Feature	
PREVIOUS DISAGREEMENT RATE	216.7245

*Features are unabbreviated in the appendix.

7 Conclusions and Further Works

Using new data on the U.S. Courts of Appeals from 1880 to the present, we constructed a random forest model to predict judicial agreement in a setting where judges are randomly assigned. We observe that judges' decisions to agree or disagree with one another are most predicted by case features. Personal features like whether two judges have sat together in the last three months and whether they are of a similar age are among the top twenty predictive features. This suggests that randomly assigned extrajudicial factors play a causal on the outcomes of cases and these factors play a material role relative to other case-level features typically viewed as appropriately determinative of case outcomes. The importance of extraneous features may be understated since some of the case level features like the number of self-certainty words in the verdict may be endogenous to the

assignment of judges. Further analysis of judge opinions using modern neural network methods, such as sentiment analysis, may also help to predict agreement between judges.

References

[1] Posner, R.A. & Landes, W.M. & Epstein, L., *Why (and When) Judges Dissent: A theoretical and Empirical Analysis*, University of Chicago Law School, 2010.

[2] Epstein, L. & Knight J. *The Choices Justices Make*, 1998. Washington, DC: CQ Press.

[3] Fischman, J.B. *Interpreting circuit court voting patterns: A social interactions framework*, 2015. Journal of Law, Economics, and Organization.

[4] Fischman, J.B. *Decision-Making Under a Norm of Consensus: A Structural Analysis of Three-Judge Panels*, 2008. 1st Annual Conference on Empirical Legal Studies Paper. Available at SSRN: https://ssrn.com/abstract=912299

[5] Glaeser, E.L. & Sunstein C.R. *Extremism and Social Learning*, 2009. Journal of Legal Analysis 1.1: 263-324.

[6] Schwartz, D.L., *Practice Makes Perfect? An Empirical Study of Claim Construction Reversal Rates in Patent Cases*, 107 MICH. L. REV. 223, 2008.

[7] Katz, D.M. & Bommarito, M.J. & Blackman, J., *Predicting the Behavior of the Supreme Court of the United States: A General Approach*, 2014. Available from: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2463244

[8] Breiman, L. *Random Forests. Machine Learning*, 45(1):5-32, 2001.

[9] Berdejo, C & Chen, D.L., *Electoral Cycles Among U.S. Courts of Appeals Judges*, 2013. TSE Working Paper No. 16-704.

[10] Chen, D.L., *Priming Ideology: Why Presidential Elections Affect U.S. Judges*, 2016. TSE Working Paper No. 16-681.

[11] Chen, D.L., Michaeli, M., & Spiro, D. *Ideological Perfectionism*, 2016. TSE Working Paper No. 16-694.

[12] Ash, E,, Chen, D.L., & Naidu, S. *The Effect of Conservative Legal Theories on Economic Jurisprudence*, 2016.

[13] Chen, D.L., Parthasarathy, A. & Verma, S. *The Genealogy of Ideology: Identifying Persuasive Memes and Predicting Agreement in the U.S. Courts of Appeals*, 2016.

[14] *The Judicial Research Initiative (JuRI)* at the University of South Carolina. Available from: http://artsandsciences.sc.edu/poli/juri/sct.html

[15] Breiman, L. & Friedman, J. & Stone, C.J. & Olshen, R.A. *Classification and regression trees*, CRC press, 1984.

[16] Geurts, P., Ernst, D., & Wehenkel, L., *Extremely randomized trees. Machine learning*, 63(1): 3-42, 2006.

[17] Surden, H., *Machine Learning and Law*, University of Colorado Law School.

[18] Perrot, M., & Duchesnay, E., *Scikit-learn: Machine learning in Python*, Journal of Machine Learning Research, 12:2825-2830.

[19] Flach, P. *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*, 2012.

[20] Sorkin, D.E., *Technical and Legal Approaches to Unsolicited Electronic Mail*, 35 U.S.F. L. REV. 325, 326, 2001.

[21] Domingos, P., *A Few Useful Things to Know About Machine Learning*, COMM. ACM at 80, 2012.

Appendix

'case_circuit': Circuit of Court, Categorical.
'case_MajSelfCertainWords': The number of words in the verdict that indicate self-certainty, Numerical. (Self-certainty words are "absolute", "apparent", "commit", etc. See [10] for detail.)
'case_geniss': Eight summary issue categories, Categorical.
1. criminal 2. civil rights 3. First Amendment 4. due process 5. privacy
6. labor relations 7. economic activity and regulation 9. miscellaneous 0. not ascertained
'case_distance': The time length to next election in month, Numerical.
'case_quartertoelect': The time length to next election in quarter, Numerical.
'case_quarter': The quarter (season) of the case, Categorical.
'case_lastquarter': Whether the case is in the last quarter before the next election or not, Binary.
'case_circuitjudge': The judge from which circuit, Categorical.
'case_district': District of origin of case, Categorical.
'J1_ageon': The age group of Judge 1 when he or she was appointed to Circuit judge, Categorical
'J1_birthyear': The birthyear of Judge 1, Numerical.
'J1_hdem': Number of Democrats in the House when Judge 1 was appointed, Numerical.
'J1_hrep': Number of Republicans in the House when Judge 1 was appointed, Numerical.
'J1_sdem': Number of Democrats in the Senate when Judge 1 was appointed, Numerical.
'J1_srep': Number of Republicans in the Senate when Judge 1 was appointed, Numerical.
'J1_hother': Number of members of other political parties in the House when Judge 1 was appointed, Numerical.
'J1_sother': Number of members of other political parties in the Senate when Judge 1 was appointed, Numerical.
'J1_state': State of judge's duty station, Categorical.
'J1_presidentname': The president's name when Judge 1 was appointed to Circuit Judge, Categorical.
'J1_degree1': The first degree of Judge 1, Categorical.
'J1_degree2': The second degree of Judge 1, Categorical.
'J1_degree3': The third degree of Judge 1, Categorical.
'J1_placeofbirthstate': The state where Judge 1 was born, Categorical.
'J1_gender': The gender of Judge 1, Binary.
'J1_raceorethnicity': The race or ethnicity of Judge 1, Categorical.
'J1_partyaffiliationofpresident': The party affiliation of the president when Judge 1 was appointed to Circuit Judge, Categorical.
'J1_district_Circuit', The district of circuit Judge 1 belongs to, Categorical.
'J1_left': Means of exiting (death, retirement, etc.), Categorical.
'J1_StateOfResidence': The state of residence of Judge 1, Categorical.
'J1_DesidenceCity': The city of residence of Judge 1, Categorical.
'J1_appres': Party of appointing president, Categorical.
'J1_aba': American Bar Association rating, Numerical.
'J1_congresi': Congress (#) in which appointment occurred, Categorical.
'J1_unityi': Whether government (Congress and president) was unified or divided when appointed, Binary.
'Inter_age': Whether the two judges are in the same age group, Binary.
'Inter_presidentname': Whether the two judges share the same president when they were appointed to Circuit Judge, Binary.
'Inter_predecessor': Whether the two judges share the same predecessor, Binary.
'Inter_party': Whether the two judges are in the same party, Binary.
'Inter_gender': Whether the two judges have same gender, Binary.
'Inter_raceorethnicity': Whether the two judges share same race, Binary.
'Inter_state': Sharing the same state of residence, Binary.
'Inter_school': Whether the two judges went to same Law schools before, Binary.
'Inter_partyaffiliationofpresident: Whether the two judges shared the same political party of appointing president, Binary.
'sit_3mo': Having sat in the same panel in past 3 month, Binary.
'sit_6mo': Having sat in the same panel in past 6 month, Binary.
'sit_1yr': Having sat in the same panel in past year, Binary.

‘sit_before’: Having sat in the same panel before, Binary.
‘previous_dissent_rate’: Previous rate of disagreement between the two judges, Numerical.
Note: Judge 2(J2) and Judge 3(J3) variables are similarly defined.