#### Dataset

I chose a dataset provided by Washington University Law, available at <http://scdb.wustl.edu/index.php>. It contained over 120,000 observations, broken down by each individual Justice’s vote on a case since 1946.

#### Features & Targets Selected

My original area of interest was to determine if Justices are truly neutral and unpartisan in their voting patterns. A cursory look at just the ‘directions’ of the total number of votes per Justice reveals that with some variation, most Justices do indeed vote in line with a liberal or conservative bias. Since the dataset did not contain any information with regard to the political leanings of the Justices and the public data on the topic is also scarce, I chose to create and use three new objective features as clues to what their political leanings might be. However, when considered as a team, Courts do not vote with a general tendency toward one direction, and the total directionality of the votes is close to being split down the middle.

As such, I changed my target from looking at individual voting directions to the outcome of the cases overall as defined by the winning party. After making feature importance and permutation importance charts, I had no choice but to drop the features I originally created. In order to avoid data leakage and be able to generalize to future predictions, I settled for the ten most relevant features that would be commonly presented to the Court by the time they consider a case, and eliminated any identifying features for the specific case or the natural Court (the specific 9 Justices sitting on the bench at any given time).

#### Baseline

Since my target is binary, I chose to use classification methods for my predictive model. The majority class was the Petitioner with a rate of 61.8%, indicating that the petitioners who are able to bring their cases in front of the Supreme Court can be predicted to win 61.8% of the time, with no additional information or variables considered. The baseline for ROC AUC is 50%.

#### Training Models & Model Performance

After splitting my main dataset into training, validation, and test sets, I trained four different models on my training data.

As expected, all classification models outperformed the logistic regression model I fit to the training data. I made basic pipelines with a few hyperparameters using the different classifiers. Surprisingly, Random Forest Classification (98%) performed better than XGBoost (85%), and ROC AUC yielded the lowest accuracy (80%) of the classification models. This indicates that all else being equal, the features I chose to create my training, validation, and testing sets are capable of performing much better than random guessing.

#### Further Considerations

In no way, shape, or form, have I found the magic formula for how to prevail at the Supreme Court. The high Random Forest Classification model performed suspiciously well. Even though I ensured, through the features I chose, to avoid introducing any outcome variables into the datasets, there could have been some interactions between individual features that is causing the questionable accuracy score. Even though it performed well across the board, from training to test data, I would have played around with more hyperparameters on my pipeline to make sure that it is not overfit. If, even with more tuning, the evaluation metrics continue to yield stellar results, I may consider opening a consultancy for lawyers hoping to win in our country’s highest court of authority.