Predicting Supreme Court Decisions: Analyzing the Impact of Metadata and Chief Justices

Introduction

This report is concerned with applications of machine learning models to predictions made by the Supreme Court based on a dataset related to legal cases provided by Fang et al. (2023). The main goal of this report is to outline different model performances tested on the classification task given both the meta-data inclusion - including attributes of the Chief Justices and case difficulty - and without additional metadata. In the broader sense, this report will address three critical research questions:

- **RQ1**: How do baseline features perform in predicting Supreme Court decisions using machine learning models?
- **RQ2**: Do case difficulty features (e.g., hearing length, vote unanimity) improve prediction accuracy?
- **RQ3**: How does the presence of the Chief Justice influence predictions and potential biases?

This research contributes to the understanding of how judicial metadata and case complexity influence legal decision-making predictions.

Literature Review

Machine learning techniques have, for example, been increasingly applied to the legal domain, such as in studies by Katz et al. (2017) and Yoo and Kock (2017), which successfully predicted Supreme Court outcomes by using NLP techniques. Most of these are blind to one crucial factor that affects predictive accuracy: judicial composition and, particularly, the role of Chief Justices. The present report contributes to these foundational works by incorporating the attributes of Chief Justices in order to see their contribution to model biases and performance.

Method

Feature Engineering

After considering these carefully, we filtered these features further to obtain the best model performance with generalizability. The features used in our study can be listed below:

Category	Feature Name	Description	Feature Type
Baseline Features (RQ1: x1-x13)	title (x1)	Represents the title of the case	Textual
	petitioner_state (x4)	State of the petitioner	Categorical
	respondent_state (x5)	State of the respondent	Categorical
	petitioner_category (x6)	Category of the petitioner	Categorical
	respondent_category (x7)	Category of the respondent	Categorical
	issue_area (x8)	Area of the legal issue	Categorical

	year (x9)	Year of the case	Numerical
	argument_day	Days since 1900-01-01, derived from argument_date (x10)	Numerical
	court_hearing_length (x11)	Length of the court hearing	Numerical
	utterances_number (x12)	Number of utterances in the hearing	Numerical
	court_hearing (x13)	Pre-trained embeddings of court hearing transcripts	Embedding
Additional Features (RQ2: x14-x15)	decision_days	Interval between decision_date (x14) and argument_date (x10)	Numerical
	majority_ratio (x15)	Indicating vote unanimity	Numerical
Court Composition Feature (RQ3: x16)	chief_justice (x16)	Chief justice presiding over the case	Categorical

Table 1- Feature Engineering Result

We have carefully assessed whether "petitioner" (x3) and "respondent" (x4) were categorical features that would be useful in the feature engineering process. To combat the dimensionality problem, we applied target encoding to these features; however, it also meant that the features suffered from overfitting issues. Quite notably, this can be seen from the disproportionately high ANOVA F-values and the huge performance difference between development and training sets, as shown below:

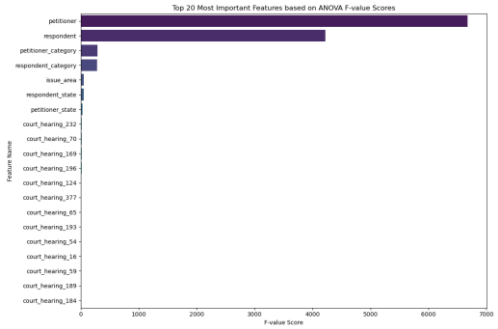

Figure 1- Top 20 most importance features based on ANOVA F-value with Features x3 and x4

Dataset	**Train**	**Dev**
Accuracy	0.9376	0.6447
Precision	0.9143	0.6517
Recall	0.9949	0.9482
F1 Score	0.9529	0.7725
AUC-ROC	0.9591	0.5832

Table 2- Logistic Regression Performance with Features x3 and x4

Data Preprocessing

During preprocessing, several encoding strategies were used for different categorical features with model performance and explainability in mind. It uses Target Encoding for encoding categorical features. One-Hot encoding was used for the 'chief_justice' feature. The "title" feature was vectorized using TfidfVectorizer. Pre-trained embeddings were used for the "court_hearing" feature. Finally, the features were combined and the top 64 selected based on the ANOVA F-test.

Target Encoding is a good strategy for high cardinality features like "petitioner_state" and "respondent_category" because it will prevent the dimension disaster as well as keeping the nuanced relationship between each category and the target variable.

As we would like to gain detailed knowledge of the impact of 'chief_justice' feature, one-Hot Encoding was chosen to create binary columns which helps the model learn specific effects associated with each chief justice (Hancock & Khoshgoftaar, 2020). It also avoids possible data leakage comparing to Target Encoding.

Models and Evaluation

We implemented three models: a Majority Class Baseline, Logistic Regression, and a Neural Network (multi-layer perceptron). Models were evaluated using Accuracy, Precision, Recall, F1 Score, and AUC-ROC, as well as the roc curves and learning curves.

Results

RQ1: Baseline vs. Machine Learning Models

Data Distribution

Our analysis revealed an imbalance in the dataset, with successful appeals accounting for approximately 63.6% of the cases:

Figure 2- Class Distribution of the Original Dataset

Hyperparameter Search

We performed extensive hyperparameter tuning for both Logistic Regression (LR) and Neural Network (NN) models using random search with cross-validation.

For LR, we focused on:

- Regularization (L1 and L2 penalties)
- Inverse regularization strength (C)
- Solver ('liblinear')
- Maximum iterations

For NN, we explored:

- Network architecture (hidden layers' number and width)
- Dropout rates
- Activation functions (ReLU, Tanh, LeakyReLU)
- Learning rate
- Optimizer (Adam)
- Maximum epochs and batch size

The use of skorch allowed us to seamlessly integrate PyTorch models with scikit-learn's hyperparameter tuning tools, ensuring a consistent and efficient optimization process across both model types.

Model Performance and Overfitting Analysis

We evaluated three models: a Majority Baseline, Logistic Regression, and Neural Network, using features x1-x13. Their performance metrics are summarized in Table 3.

Model	Dataset	Accuracy	Precision	Recall	F1 Score	AUC-ROC
Majority Baseline	Train	0.636	0.636	1	0.7775	0.5
	Dev	0.636	0.636	1	0.7775	0.5
Logistic Regression	Train	0.6824	0.6999	0.875	0.7777	0.7037
	Dev	0.6534	0.6843	0.8447	0.7561	0.6117
Neural Network	Train	0.6745	0.6897	0.8863	0.7757	0.6993
	Dev	0.6603	0.6855	0.861	0.7633	0.6349

Table 3- Model Performance Comparison

Although the Recall rate of baseline reaches 1, it is because the dataset has an unbalanced class distribution, which also leads to even slightly higher F1 score for baseline in most cases. For other metrics, Logistic Regression and Neural Network significantly outperform the majority baseline. The Neural Network has a slightly better performance on both accuracy and AUC-ROC between them, which suggests that this Neural Network has a better discrimination capability.

These small differences between training and development set performance for both models suggest that they are doing a great job on generalizing to unseen data. It would appear the Neural Network generalizes slightly better, as this gap is smaller.

We also analyzed the learning curves, as plotted in the following figures:

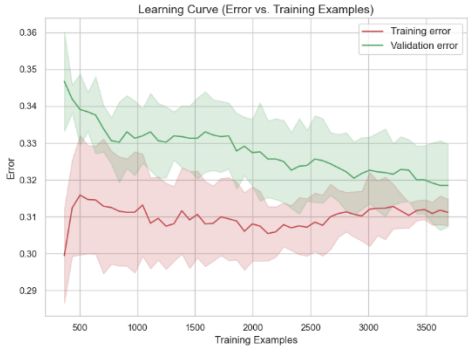

Figure 3- Learning Curve for Logistic Regression

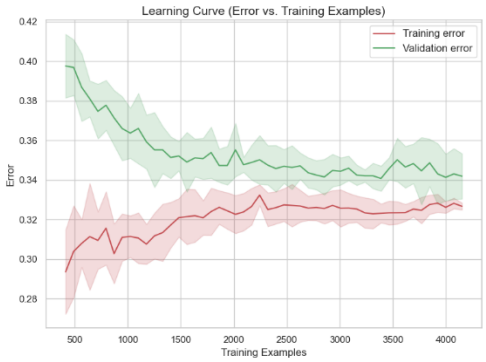

Figure 4- Learning Curve for Neural Network

These give some indication that NN converges sooner than LR due to its ability to capture nonlinear patterns. It also generalized better, as can be seen from the slightly smaller gap between the training and validation errors. Both models finish similarly in performance, which also suggests that LR does not miss the key information. Closeness between training and validation errors for both models indicates that they generalize well, with limited overfitting. This reflects some trade-offs between model complexity and performances for predicting Supreme Court decisions.

ROC Curve Analysis

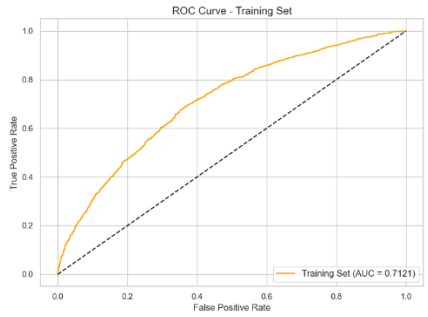

Figure 5- Training Set ROC Curve for Logistic Regression

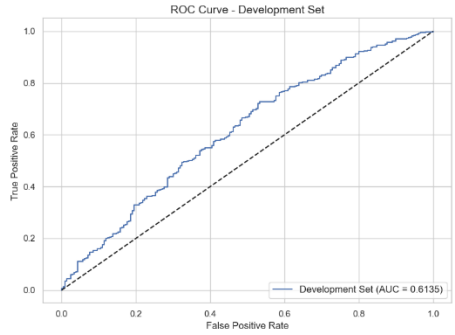

Figure 6- Development Set ROC Curve for Logistic Regression

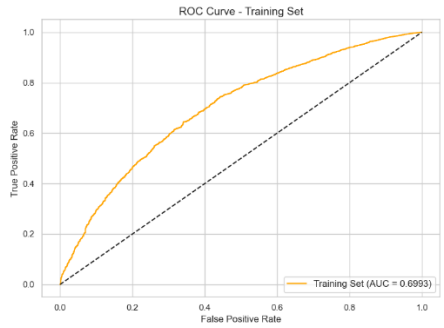

Figure 7- Training Set ROC Curve for Neural Network

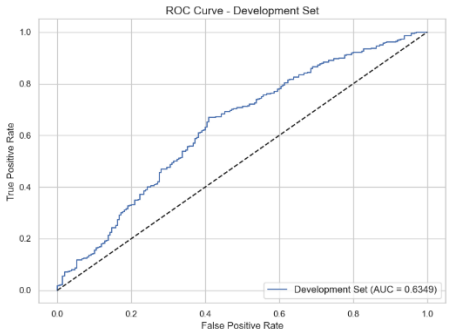

Figure 8- Development Set ROC Curve for Neural Network

The ROC curves demonstrate a clear separation between training and development sets for both machine learning models. The Neural Network shows a more robust curve, with a higher AUC-ROC value, indicating superior discriminative power compared to the Logistic Regression model.

RQ2: Impact of Case Difficulty Features

We added hearing length (x14) and vote unanimity (x15) to assess their impact on model performance. Table 4 presents the performance difference:

	Model	Dataset	Accuracy	Precision	Recall	F1 Score	AUC-ROC
RQ1	Logistic Regression	Train	0.6824	0.6999	0.8750	0.7777	0.7037
		Dev	0.6534	0.6843	0.8447	0.7561	0.6117
	Neural Network	Train	0.6745	0.6897	0.8863	0.7757	0.6993
		Dev	0.6603	0.6855	0.8610	0.7633	0.6349

RQ2	Logistic Regression	Train	0.6876	0.7025	0.8812	0.7818	0.7132
		Dev	0.6534	0.6819	0.8529	0.7579	0.6454
	Neural Network	Train	0.6776	0.6896	0.8952	0.7791	0.7083
		Dev	0.6568	0.6794	0.8719	0.7637	0.6542

Table 4- Performance Difference with additional difficulty features

RQ3: Influence of Chief Justice

To investigate potential biases in Supreme Court decisions, we examined the impact of features related to the Court's composition, focusing particularly on the influence of different chief justices (x16) on ruling patterns.

Chi-Square Test and Success Rates

Analysis

To assess the relationship between the 'chief_justice' feature and case outcomes, we conducted a chi-square test of independence on training set. The results revealed a statistically significant association between the presiding chief justice and the success of appeals ($\chi^2 = 15.7420$, $p = 0.0013 < 0.005$). This finding suggests that the identity of the chief justice may be a relevant factor in predicting Supreme Court decisions, warranting further investigation into its potential influence on judicial outcomes.

Additionally, the success rates varied notably across different chief justices:

Chief Justice	Success rates
Warren	68.07%
Roberts	65.23%
Burger	63.96%
Rehnquist	59.85%

Table 5- Success Rate by Chief Justice

These results suggest that the identity of the chief justice may influence the Court's decisions, raising concerns about potential bias in the judicial process.

Feature Importance Analysis

The logistic regression model achieved an accuracy of 63.26% and an AUC-ROC of 0.6140 on the development set. We use coefficient magnitude as the metrics of feature importance. It revealed that 'chief_justice_Warren' ranked 34th out of all features with an importance score of 0.366033, while 'chief_justice_Rehnquist' ranked 45th with a score of 0.183134. This suggests that while these features are not among the most influential, they do have a noticeable impact on the model's predictions.

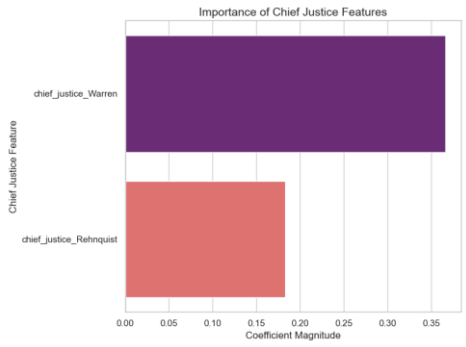

Figure 9- Chief Justice Feature Importance for Logistic Regression Model

The neural network model showed slightly better performance, with an accuracy of 65.68% and an AUC-ROC of 0.6414 on the development set. The mean permutation importance is used to evaluate the importance of chief justice feature. Notably, 'chief_justice_Rehnquist' ranked 2nd among

all features, with an importance score of 0.004333. 'chief_justice_Warren' ranked 9th overall, with an importance score of 0.001733.

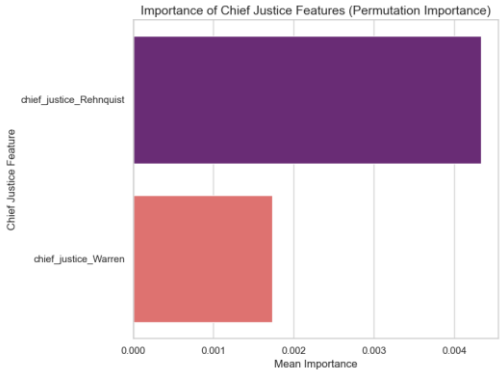

Figure 10- Top 20 Most Important Features of neural network Model with feature x16

Impact of Chief Justice Features

Impact of Chief Justice Features

To further assess the impact of chief justice features, we compared both models' performance with and without these features:

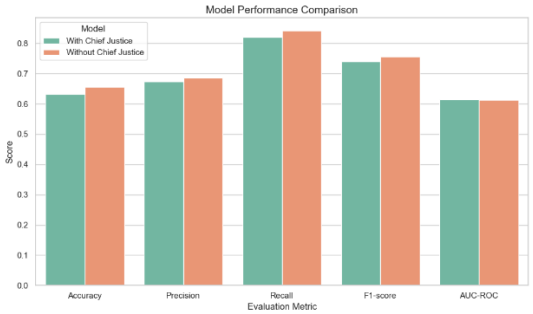

Figure 11- Logistic Regression Model Performance with/without Feature x16

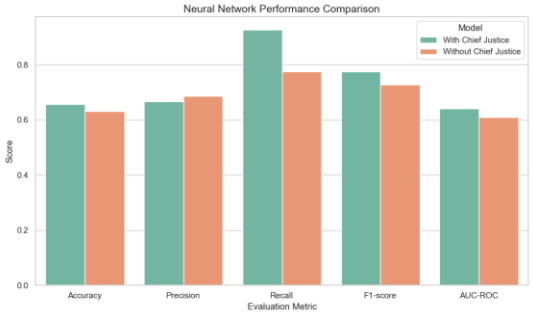

Figure 12- Neural Network Model Performance with/without Feature x16

Interestingly, the two models show different patterns when chief justice features are removed. For the logistic regression model, removing these features led to an increase in accuracy but a slight decrease in AUC-ROC. This suggests that while the chief justice features might introduce some noise that affects overall accuracy, they do contribute to the model's ability to rank predictions correctly.

In contrast, the neural network model shows a decrease in both accuracy and AUC-ROC when chief justice features are removed. This indicates that these features play a more integral role in the neural network's predictive capabilities, aligning with our earlier observation of their high importance rankings in this model.

Discussion / Critical Analysis

Model Performance and Characteristics

Our study showed that various predictive techniques had different strengths and weaknesses:

- **Majority Baseline**: Easy to be implemented but not practical; it cannot consider any feature and has poor performance for classes that are less frequent.
- **Logistic Regression**: Good performance but less interpretable, further understanding of the factors driving those decisions becomes important. Linear by nature, so has limitations because it is hard to capture complex relationships using them.
- **Neural Network**: The nonlinear pattern modeling enabled reaching the highest

accuracy, but at higher computational resources and it is not interpretable.

These machine-learning baselines outperformed the naive baselines; this again was well in line with Katz et al. (2017). performance increases were modest at the best cases, which indicated an inherent limitation in predictability, probably because of the complex nature of deliberations made in the Supreme Court.

Impact of Case Difficulty Features

Including the case difficulty features, RQ2 leads to systematic gains in model performance with the most pronounced improvements in AUC-ROC scores. This underlines the relevance of the case complexity feature-a factor often ignored in the realm of judicial decision prediction as mentioned by Yoo and Kock (2017).

Influence of Chief Justices

Our most striking finding is about the contribution of chief justices to case outcomes (RQ3). The strong correlation of chief justices with successful appeals certainly stands in tension with conceptions of absolute judicial neutrality discussed by Baum (2009). Once more, though, here the results show correlation, not causation.

Contrasting influences of the chief justice characteristics in logistic regression and neural network models underlines complexity in these relationships and performance vs explainability trade-off.

Feature Engineering and Model

Generalization

In fact, the process of feature engineering we adopted-especially in terms of the decision to use higher-order categorical features, without going into any particular party identities-was instrumental in the resolution of the overfitting problem. Hence, by taking a lead from the useful insights provided by Katz et al. (2017) on the "problems" of features of high cardinality in legal datasets, this allowed our models to generalize better for unseen cases.

Limitations

There are several limitations that our study has and which should be addressed in the future. As our research relies on only one dataset, generalizing the findings of our study may be limited. In this way, future studies could thus consider including more than one dataset or even an extended temporal range of data. Furthermore, other justices' influence or a broader court composition might be explored in order to gain a fuller understanding of how judicial decision-making unfolds.

Conclusion

This work demonstrates the promise of machine learning in predicting Supreme Court decisions while discovering intricate relationships between judicial attributes and case outcomes. Although our models outperform simple baselines, there are limits to predictability inherent to the domain.

This was further improved by incorporating case difficulty features and chief justice attributes. But this also raised very fundamental questions with regard to judicial impartiality and what drives decisions in the Supreme Court. These results really undercut any simplistic notions of judicial decision-making and, at the same time, underscore the need for nuanced understanding of Supreme Court functioning.

Future studies should investigate the interactions between judicial attributes and other case characteristics. It shall study temporal trends in the decision-making

pattern. More interpretable, complex models could provide a bridge to connect the gap in predictive power with explainability, an essential consideration in legal applications.

Ultimately, it shows both the strengths of machine learning in legal prediction and the complexity of decisionmaking in the Supreme Court. As these predictive tools continue to be refined, accuracy must be tempered with legal, ethical, and social consideration.

References

Baum, L. (2009). The puzzle of judicial behavior. University of Michigan Press.

Fang, B., Cohn, T., Baldwin, T., & Frermann, L. (2023). Super-SCOTUS: A multi-sourced dataset for the Supreme Court of the US. In Proceedings of the Natural Legal Language Processing Workshop 2023 (pp. 202–214). Association for Computational Linguistics.

Hancock, J. T., & Khoshgoftaar, T. M. (2020). Survey on categorical data for neural networks. Journal of Big Data, 7(1), 1-41.

Katz, D. M., Bommarito II, M. J., & Blackman, J. (2017). A general approach for predicting the behavior of the Supreme Court of the United States. PloS one, 12(4), e0174698.

Yoo, J., & Kock, N. (2017). Predicting Supreme Court decisions using neural networks. Proceedings of ICESD 2017.