Feature Importance for the Prediction of Cases in the European Convention on Human Rights

JACKSON GAZIN and NIKOLAS LINDAUER*, Wake Forest University, USA

In this paper, we present an evaluation of feature importance for a neural network operating on a large corpus of legal documents. Examined are issues of feature selection, computational efficiency, and network architecture. We attempt several models to predict violations of human rights given a series of facts from the court transcript. We compare a random forest to a neural net and evaluate the performance of both. Given the high cost of calculating Shapley values on large, multi-label models, we propose an approximation technique and compare it to random forest methods of determining feature importance.

CCS Concepts: • **Applied computing** → **Law**; • **Computing methodologies** → **Feature selection**; • **Networks** → **Network performance analysis**.

Additional Key Words and Phrases: Human Rights, Legal Data Science

ACM Reference Format:
Jackson Gazin and Nikolas Lindauer. 2018. Feature Importance for the Prediction of Cases in the European Convention on Human Rights. *ACM Trans. Graph.* 37, 4, Article 111 (August 2018), 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Legal text has proven to be an abundant source of data for natural language processing tasks, with research focused on predicting topics for future legislation, the likelihood of success in various lawsuits, and the generation of legal documents [Katz et al. 2023]. Progress in these fields could make the chaotic legal world more predictable, encouraging confidence among stakeholders, while expanding access and awareness for those often excluded from the legal domain. One factor limiting the widespread use of more powerful neural networks is their uninterpretable 'black box' nature, which inhibits the trust of the legal community in decision-making.

Our dataset represents a collection of over 11,000 cases that were presented to the European Convention on Human Rights (ECHR), which adjudicates alleged violations of human rights provisions. We have a list of facts about each case, which was cultivated by Chalkidis et al. (2019) using regular expression methods developed by Aletras et al. (2016), and a list of the violations the defendant was found guilty of [Chalkidis et al. 2019] [Aletras et al. 2016]. An example of such an input is provided in Figure 6. For cases in which the defendant was acquitted, the list is empty. Our goal was to predict which, if any, provisions were violated. The lists of facts were tokenized by SPACY for model consumption. The data is divided into training and development sets with cases from 1959 to 2013, and our test set contains cases from 2014-2018. There are 66 provisions protected under the ECHR, but only 18 different violations occur in our training data, and only 9 occur in more than 50 cases. We utilized Python 3.8 modules for our entire Python code.

We decided to focus on classifying these 9 cases. Note that the most common violations involved the right to a fair trial (6), the right to liberty and security (5), and the prohibition of torture (3) [ECH 1950]. Many violations never occurred, but some of the rarer ones we attempted to predict included the proper execution of judicial rulings (46), plenary court (25), and the right to marry (12). We can see in Table 1 the distribution of articles in our training set.

Code	Count	Percentage (%)
2	215	4.50
6	1966	41.12
13	519	10.86
5	599	12.53
14	104	2.18
8	442	9.24
3	526	11.00
7	18	0.38
34	36	0.75
11	82	1.72
38	24	0.50
9	34	0.71
10	194	4.06
18	4	0.08
4	4	0.08
12	8	0.17
25	5	0.10
46	1	0.02

Table 1. Distribution of Article Codes among our Training Set

The Cross Industry Standard Process for Data Mining (CRISP-DM) is the process model used for data science projects. The sequential and iterative phases in CRISP-DM are: business understanding, data understanding, data preparation, modeling, evaluation, and deployment (Figure 1).

Classification task

We built two classification models: one that performs multi-label classification on the nine most commonly occurring violations of human rights, and one that performs multi-label classification on all 18 different violations that occur in our training dataset.

We found that by using TF-IDF matrices with only 100 words as our input, we were able to achieve a Hamming loss of 0.0899 on

Authors' address: Jackson Gazin, gazij22@wfu.edu; Nikolas Lindauer, lindn23@wfu.edu, Wake Forest University, 127 Manchester Hall, Winston-Salem, North Carolina, USA, 27109.

ACM 0730-0301/2018/8-ART111
https://doi.org/XXXXXXX.XXXXXXX

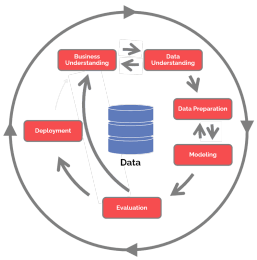

Fig. 1. **CRISP-DM diagram**. Diagram of the six different phases in CRISP-DM and how they relate to each other.

our test set for our reduced model using a Neural Network with one layer. For our low-shot model, we used a TF-IDF matrix with 1000 words, and we were able to achieve a Hamming loss of 0.0460. However, this model failed to overcome the severe class imbalance for the rarer violation codes.

Note, we used the Scikit-Learn Python library to build our Machine Learning models [Pedregosa et al. 2011] and the Sci-Keras library to build our neural network model [Badaracco 2020]. Our batch job to peform Phase 2, Phase 3, Phase 4, and Phase 5 of the CRISP-DM process took 2.18 GB of memory to run.

Feature Importance Task

Our secondary goal was to gain some clarity with regards to the impact of individual tokens on our model's predictions. We used two methods, permutation importance and Shapley values, to estimate each token's importance [Molnar 2022].

We compared their outputs, along with the feature importance measured by our random forest, to attempt to better justify our model's decisions. The importance of a feature in a bad model might differ significantly from the importance of the same feature in a good model, so the relevance of such analysis is subject to the accuracy and sensitivity of our best model.

2 PRIOR WORKS

2.1 Multi-Label Legal Classification

Chalkidis tested several models known to produce strong results in multi-label classification with sparse data, including a Bidirectional GRU with attention (BiGRU-Att), Hierarchical Attention Network (HAN), Label-Wise Attention Network (LWAN), and Bidirectional Encoder Representations from Transformers (BERT). In this paper, their team extended BERT's word limit (512 tokens) to accommodate more complex legal documents. This model, dubbed Hierarchical BERT (HIER-BERT), takes each fact, creates an embedding of each fact with basic BERT, and combines them to create an embedding for the list of facts. Many of these models utilize complex self-attention scores to augment their embeddings, allowing them to better understand the context of each fact and intricate references, which are commonplace in legal proceedings.

In general, each neural network found similar success across F1 scores, ranging from 50 to 60. They found their HIER-BERT model to be the most successful, producing an F1 score of 60, though HAN was very close with a score of 59.9. In detecting low-frequency classes (fewer than 50 cases), the LWAN model proved to be the most capable, achieving an F1 score of 10.6, with HIER-BERT in second place at 8.9. HAN was the least sensitive to low-shot classes, reaching an F1 of 2.8.

2.2 Feature Importance

Yang et al. (2020) discuss the importance of transparency for neural network approaches to gain industry adoption [Yang et al. 2022]. Specifically in financial technology, black-box models struggle more than interpretable machine learning models to persuade stakeholders who are wary of risking significant amounts of money. This paper uses neural network architectures similar to those used by Chalkidis and implements an algorithm that generates texts modified to be "counterfactual" to true test inputs by replacing high-importance words with their opposites. By studying these counterfactuals, they worked to identify the smallest possible differences between inputs that would trigger a different classification from the original. This approach allowed them to create highly interpretable and quantifiable measurements for the importance of given words in a text.

3 METHODS

The dataset we used is available for download at ECHR Dataset. It contains JSON files for each case, which include the text of the case, whether the text was in English, and which, if any, human rights articles were violated. Other information was present in each JSON file, but we primarily used these three points to load our data.

As the ECHR attempts to maintain neutrality with regards to nationality, Chalkidis thought it important to minimize the impact of nation-specific features. Accordingly, all our inputs had been anonymized using SPACY's named-entity recognizer [Virtanen et al. 2020]. This replaced all recognized entities, specifically people and locations, with placeholders <location> and <person>. SPACY failed to recognize some misspelled or unknown entities, such as lowercase "iraq" or "Armeni", but none of these features were among the most influential.

Each anonymized file was already categorized into train, dev, and test sets. We initially loaded all the files into three data frames, selecting only those that were anonymized and written in English. However, upon further exploration, we noticed that some of these files still contained a significant amount of non-English words. We initially found the presence of words in Arabic or Georgian to be strong indicators of human rights abuse, but to preserve the impartiality of our model, we removed all cases such that less than half of their "facts" text was in other languages. These changes did not significantly reduce the size of our dataset or the accuracy of our model. Our final test set contained 7096 rows, and our training set contained 2997. Finally, the dataset was balanced by Chalkidis for binary prediction (abuse vs no abuse), but not for individual labels. Thus, we had the opportunity to attempt prediction on low-occurrence features.

3.1 Phase 2: Row Selection

For Phase 2, our first task was to determine how to tokenize the list of facts. We removed punctuation from our text and lemmatized it by reducing each word to its base form. We considered making a

negation adjustment to our text, meaning replacing each token in a sentence after a no, none, not, never, or "n't" with <NOT_token>. We thought this would be especially valuable for legal text, but discovered our accuracy decreased significantly with this modification. Perhaps a more robust negation approach could improve our results, but we have left this to future researchers.

After tokenizing our text, we created a TF-IDF matrix using scikit-learn's vectorizer. Note, this process required creating a vectorizer that tokenized our text and generated a TF-IDF matrix from it. It took approximately 82 minutes to fit the vectorizer to our training text and create the resulting training TF-IDF matrix, and an additional 52 minutes to utilize the vectorizer to create the initial development and test TF-IDF matrices.

TF-IDF matrices are sparse matrices that contain the count of each token for each document, or text facts in our case. It was crucial that our models work well on very sparse data, given the computational demands and complications with distance metrics. We selected logistic regression instead of negative binomial regression as our baseline classifier due to slightly improved performance on training accuracy. We then tested different means of encoding our data, including negation and the minimum prevalence for a human rights abuse article code to be encoded in our response vector. We tested thresholds of 1%, 2%, and 3% and determined, as seen in 2, that a threshold of 1 percent with no negation produced the optimal results on the development set: a mean hamming loss of .12 across all folds. This means our error rate across all articles was 12 percent. Note, it took 36 minutes to conduct this grid search and then create the corresponding response vectors.

Negation	Threshold (%)	Mean Hamming Loss
No	1	0.12
No	2	0.15
No	3	0.14
Yes	1	0.22
Yes	2	0.17
Yes	3	0.19

Table 2. Combined Evaluation Metrics for Phase 2 with Logistic Regression Models across different threshold

To match our model's output to the list of violations, we used SKLEARN's `OneVsRestClassifier` as our baseline model to predict a vector of ones and zeros. This approach breaks down the multi-label classification into a series of binary classifications (for our baseline, using logistic regression), then concatenates them into a vector [Harris et al. 2020].

```
{"2": 0, "3": 1, "5": 2, "6": 3, "8": 4,
"10": 5, "11": 6, "13": 7, "14": 8}
```

Note, for example, that the output vector `[0,0,0,1,1,0,1,0,0]` has ones in the 4th, 5th, and 7th positions, which would correspond to violations of the 6th, 8th, and 11th provisions. We then mapped the true output of each case onto a similar vector and computed the Hamming loss between our predicted and true outputs to assess the accuracy of each prediction. Later, we considered the expanded mapping below to predict low-shot labels.

```
{2: 0, 3: 1, 4: 2, 5: 3, 6: 4, 7: 5, 8: 6,
9: 7, 10: 8, 11: 9, 12: 10, 13: 11, 14: 12,
18: 13, 25: 14, 34: 15, 38: 16, 46: 17}
```

3.2 Phase 3: Feature Selection

In Phase 3, to enhance computational efficiency, we explored feature selection techniques for our TF-IDF matrix. We utilized two methods from scikit-learn:

(1) **SelectKBest:** This method selects the top K features based on a chosen scoring function. We considered the following K values $[100, 200, 300, 400, 500]$.
(2) **Variance Threshold:** This method removes all features whose variance does not meet a certain threshold. We applied thresholds of 0.05, 0.10, and 0.20.

We scored each method with the following scoring function:

$$\text{Score} = .6 * \text{Mean Hamming Loss} + .001 * \text{STD Hamming Loss} + .3 * \text{Num of Features} \quad (1)$$

The mean and standard deviation of Hamming Loss were calculated via a 3-Fold Cross-Validation (CV). The output is reflected in Table 3, and we elected to pursue `SelectKBest` where $K = 100$ for its impressive results on our score and its comparative simplicity with regards to feature importance. Note, it took 48 minutes to complete this grid search and choose the best method of feature reduction.

Type of Feature Reduction	Mean Hamming Loss	Score
Variance Threshold (Top 5%)	0.18726	16.107
Variance Threshold (Top 10%)	0.13277	15.235
Variance Threshold (Top 20%)	0.20596	13.609
Variance Threshold (Top 30%)	0.16338	11.891
SelectKBest (k=100)	0.16802	0.134
SelectKBest (k=200)	0.22875	0.205
SelectKBest (k=300)	0.21398	0.222
SelectKBest (k=400)	0.16788	0.225
SelectKBest (k=500)	0.18007	0.262

Table 3. Comparison of Feature Reduction Techniques

We discuss the tokens selected features and their importance in section 4.

3.3 Phase 4: Model selection

We performed a grid search across various classifiers known to work well on sparse data, including Ridge, Elastic Net, Stochastic Gradient Descent Classifier (SGD), and Random Forest. The SGD Classifier, a linear classification method,seeks the ideal hyperplane to distinguish between various class memberships. Ridge and Elastic Net Classification are basically iterations of logistic regression except they include regularization terms. Random Forest outperformed all others on our scoring function, which was a weighted sum of the mean and standard deviation of the Hamming loss. Random Forest also offers the added benefit of providing a robust explanation of feature importance, which we will discuss later. We trained these

models on our development set to avoid data leakage. In Table 4, we show the results of each machine learning classifier. Note that to choose the best model, we used Scikit-learn's functions for both our models and our grid search. It only took around 57 seconds to run this grid search and approximately one second to fit the best model [Van Rossum 2020].

Table 4. Best Performing Models - Scores

Model	Score	Mean Hamming	Std Hamming
Logistic Regression	0.2947	0.4138	0.0169
Ridge Classifier	0.29198	0.4101	0.01627
SGD Classifier	0.29212	0.4101	0.01675
Random Forest	0.2729	0.3862	0.0085

We then evaluated the utility of using a Neural Network. Due to the infrequent occurrence of many classes, we optimized our Neural Network based on the F1 score rather than accuracy. The F1 score is a harmonic mean of precision and recall, providing a single metric that balances the trade-off between predicting positive (violation) and negative (no violation) cases for each article.

We constructed a simple Neural Network with one hidden layer, which we compared to the Random Forest. It slightly outperformed the Random Forest in sensitivity across some classes. Given these promising preliminary results, we constructed an 8-layer NN utilizing dropout and normalization to help prevent overfitting and improve accuracy for the low-count article codes.

3.4 Phase 5: Model evaluation

Our low-shot neural network model performed poorly on articles with low counts, predicting only two positive cases in the training set. We returned our focus to the more sensitive, single layer NN. Our simpler Neural Network outperformed our Random Forest in mean Hamming loss (0.0727 vs 0.0856) across articles. However, we can see in Tables 5 and 6 that the NN fell short in terms of sensitivity (0.168 vs 0.207). The Neural Network model demonstrated greater sensitivity in articles with higher counts, suggesting that re-balancing our dataset might improve our model's performance. We leave the task of re-balancing a multi-label classification problem to future researchers.

Interestingly, the Random Forest had much more success at detecting violations 6 (right to a fair trial) and 10 (freedom of expression) than the Neural Network, while the Neural Network was more successful in identifying violations 3 (prohibition of torture) and 11 (freedom of assembly and association). This disparity suggests that perhaps these two models could be used in tandem to identify low-frequency cases with greater success.

4 FEATURE IMPORTANCE

4.1 Methods

The first metric of feature importance we explore is permutation importance. This method is based on shuffling the values of each column in our TF-IDF matrix, effectively turning each token into noise, and measuring the drop in model accuracy. The most influential features identified using this method are displayed in Table 7 for the RF and Table 8 for the NN.

Article Code	Acc	Sens	Spec	# Positive	# Negative
2	0.9713	0.2966	0.9990	118	2879
3	0.8849	0.5019	0.9660	524	2473
5	0.9039	0.3394	0.9866	383	2614
6	0.7898	0.1738	0.9864	725	2272
8	0.9296	0.0372	0.9986	215	2782
10	0.9653	0.0095	1.0000	105	2892
11	0.9837	0.1481	0.9990	54	2943
13	0.8926	0.0062	0.9993	322	2675
14	0.9867	0.0000	1.0000	40	2957

Table 5. Performance Metrics by Article Code (One Hidden Layer Neural Network

Table 6. Performance Metrics for Various Article Codes (Random Forest Model)

Article Code	Acc	Sens	Spec	# Positive	# Negative
2	0.967	0.195	0.998	118	2879
3	0.860	0.265	0.985	524	2473
5	0.895	0.308	0.980	383	2614
6	0.760	0.345	0.892	725	2272
8	0.928	0.051	0.996	215	2782
10	0.959	0.219	0.986	105	2892
11	0.982	0.000	1.000	54	2943
13	0.894	0.025	0.998	322	2675
14	0.987	0.000	1.000	40	2957

Table 7. Permutation Feature Importances RF

feature	weight	std
hearing	0.027160	0.002502
detention	0.018085	0.003373
judgment	0.014815	0.004052
investigation	0.007674	0.001593
abscond	0.006874	0.000687
child	0.006874	0.002169
arrest	0.006874	0.001088
inmate	0.004805	0.000163
officer	0.003871	0.001186
son	0.003537	0.001311

Our second metric of feature importance is the Shapley value, which measures the contribution of each feature in relation to subsets of the dataset that exclude that feature [Gillies et al. 07]. For each subset S excluding the feature under assessment, the following formula is used: S represents the subset, N represents the set containing all features, i represents the i-th feature, and v represents the model's predictions given the input. It is important to note that the actual values of the test set are unnecessary for this calculation, as we are only measuring the change in prediction, not accuracy.

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} (v(S \cup \{i\}) - v(S))$$

Table 8. Permutation Feature Importances NN

index	Permutation Importance
detention	0.013866
judgment	0.008008
hearing	0.006266
cell	0.004560
medical	0.003411
officer	0.002855
treatment	0.002632
investigator	0.002373
release	0.002373
cardinal	0.002262

Shapley values, originally designed for binary classification, become extremely computationally expensive when applied to models with many features. For instance, using a random forest with 9 outputs and 100 features, calculating the Shapley values for each feature would require predicting the output for the test set 100! times. Instead of using every possible combination of features, I considered sampling from a subset of these combinations to approximate the Shapley values. However, even this proved too expensive. By sampling only 10 out of the 100! combinations of features (which would lead to an extremely unsatisfactory estimate), we would still need to make 21 million predictions with our model.

Ultimately, we opted to create an algorithm that combines the simplicity of permutation importance with the focus on individual violation outputs characteristic of the Shapley values. For each feature, we shuffle the values and generate a prediction, which we then compare to the original output. We calculated the change in prediction for each violation code and were able to derive different Shapley Value approximations for each violation code, providing us with very granular feature importance. The pseudo-code for this method is provided below:

Algorithm 1 Shapley Approximations

```
procedure CALCULATE_MULTIOUTPUT_SHAPLEY_PERMUTATION
    base_predictions ← predictions from complete test set
    dict_of_frames ← dictionary with outputs : data frames with feature names as rows
loop:
    for each feature i do
        modified_copy ← test set with shuffled feature i
        modified_predictions ← predictions with shuffled feature i
        for each output j do
            impact ← mean of differences between base and modified predictions
            dict_of_frames[output[j]] ← impact
    dict_of_frames ← sorted version
return dict_of_frames[output[j]]
```

The output, being the Shapley approximations for each feature for each violation code, is presented on Table 9. For our test set, this code consumed 710 MB and ran in 10.5 seconds.

When finding the Shapley approximation for feature k, the best model would typically use every feature excluding k, thus still preserving the most successful model in every combination from the true Shapley calculation. In this way, we hoped to reach an optimal compromise between efficiency and accuracy with our approximation. It would likely produce biased results, specifically values lower than the true Shapley values, since it is not averaged with many lower-performance models, but we hope this overestimation is relatively similar across features. The key downside here is that Shapley accounts for feature interactions by removing other features via combinations that might inform us similarly to feature k. Neural Networks and Random Forests are particularly suited to identify such connections and use them efficiently. When feature k has many important interactions, then iterating through all feature combinations allows us to best understand the role k plays across these models. However, this approximation might underestimate the ability of other features to compensate for the removal of k. It would be interesting to compare this approximation with true Shapley values in simulation studies to better understand this impact.

The Random Forest Classifier also brings insight into what features might be most useful, independent of the neural network. Random Forest importance is determined by how much the use of each feature increases the Gini purity score, averaged across all trees. The feature importance for the Random Forest is also available on the Table 9.

4.2 Analysis

In terms of the features themselves, it is worth noting that of the most important 15 features for RF, nine are related to court proceedings. Cardinal, the most influential token for the RF, ironically is the spacy-anonymized version of a number. We can also see how many of the features have no influence on many of the classes. For class 10 and 14, our neural network produced no positive classifications regardless of any feature permutation. This calls into question our method of feature selection, given it was based only on the k-best from logistic regression. Perhaps with an extended TF-IDF, we might improve sensitivity on low-shot cases.

We found correlations between feature importance for each code to see when our model used similar features to predict different violations. These are shown in Table 10. Note that since we were unable to detect violation 14, it has been removed.

It is interesting to note that the Random Forest feature importance scores were only weakly correlated with the Shapley estimates for each violation. Perhaps if we assessed the importance of each feature for each output on the Random Forest, the connections between the associated features for the Neural Net would be more pronounced.

Also noteworthy are the only two strong correlations (greater than 0.5) formed between Violations 3 (prohibition of torture) and 13 (right to an effective remedy), and between Violations 3 and 5 (right to liberty and security). Torture charges are often brought against government agents who are usually required to provide some kind of remedy, so some association between these crimes is to be expected. If there was torture, there is likely some need to provide a remedy, and if one right was violated, likely the other quickly followed. Liberty and security focuses primarily on the rights of those who have been detained by authorities, namely access to information regarding their charges, access to a lawyer, and a reasonably quick trial. This too seems like it would affect similar populations as victims of torture, and thus their transcripts might involve similar vocabulary.

5 DISCUSSION

The nature of our highly imbalanced dataset made it challenging for simple neural networks to effectively recognize the less common

labels. This issue was somewhat mitigated with a more complex network; notably, in Chalkidis's work, his team achieved the best results with an extended BERT model. Additionally, the Random Forest has a balancing parameter that likely contributed to its accuracy.

Many of the most influential features appear to involve the legal system itself, rather than the details of the case. This is likely because more serious matters demand greater attention from existing legal frameworks. Notably, our model had the greatest success in predicting violations of codes 6, 3, and 5, which are the right to a fair trial (6), the right to liberty and security (5), and the prohibition of torture (3). Perhaps the very explicit, well-documented, and bureaucratic nature of these crimes allows them to be more easily detected not just by our model, but by the courts themselves. In contrast, rights such as freedom of expression and discrimination can be more easily obscured.

It might be interesting to further study the similarity in feature importance across different crimes. This investigation could provide valuable insights into the structure of rights, as perhaps securing rights to attorneys, trials, and information about one's detainment might itself secure one against torture and any necessary remedy associated. Such knowledge could then guide the correct allocation of resources when one better understands the connections between laws.

6 CONCLUSION

One concerning limitation that emerged only after completion of our study was that our feature-selecting logistic regression operated on a OneVsRest classifier. This approach, which treats each class output as a separate binary classification problem, likely selected features crucial for detecting the most prevalent abuses while potentially overlooking those critical for low-shot cases. Although necessary to manage our dataset size, reducing feature selection could potentially enhance our performance, especially in low-shot scenarios.

Optimizing our NN model based on the F1 score resulted in very low test sensitivity scores. In future attempts, it might be beneficial to give additional weight sensitivity in optimization. Additionally, we theorize that using vector embeddings as input could boost performance for the low-shot model.

An intriguing avenue for future research is the actual content of each human rights provision. While the vague themes of each code can be discerned from the features used to detect them, we are not fully leveraging the language within the laws themselves. Perhaps by developing a pooled embedding of the human rights codes and measuring similarity with our encoded set of facts—or better yet, an attention-based memory that new inputs could build upon—we might significantly improve results beyond focusing solely on the language of the text. A sequence-to-sequence model might enhance our sensitivity in low-shot prediction, as greater contextual understanding could enable a Neural Network to better grasp the relationships between inputs and outputs.

Reflecting on this work, it is crucial to distinguish between explanation and justification in the context of legal NLP. While machine learning and neural networks can provide explanations for their outputs, a vital step before any widespread adoption of legal NLP is justification. Unlike mere explanation, justification must also offer reasons why the outputs of NLP algorithms should be considered when determining legal strategy or predicting future legislation. If our models disproportionately emphasize nationality or other features that legal authorities are explicitly forbidden to consider, then it seems imperative that these bodies be precluded from using such models until we can not only explain but also justify why our feature selection is pertinent.

DATA AND CODE AVAILABILITY

All data used in this paper and the code for the analysis can be accessed through Github.

ACKNOWLEDGMENTS

We thank Dr. Natalia Khuri for her invaluable feedback and comments. We used Chat GPT to check the spelling of our paragraphs.

REFERENCES

1950. European Convention on Human Rights. Council of Europe. https://www.echr.coe.int/Documents/Convention_ENG.pdf As amended by Protocols Nos. 11 and 14, supplemented by Protocols Nos. 1, 4, 6, 7, 12, 13, and 16.

Nikolaos Aletras et al. 2016. Predicting Judicial Decisions of the European Court of Human Rights: A Natural Language Processing Perspective. *PeerJ Computer Science* 2 (2016), e93. https://doi.org/10.7717/peerj-cs.93

Adrian Garcia Badaracco. 2020. Scikit-Keras Documentation. https://adriangb.com/scikeras/stable/ Accessed: 2024-05-06.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural Legal Judgment Prediction in English. arXiv:1906.02059 [cs.CL]

Sean Gillies et al. 2007–. Shapely: manipulation and analysis of geometric objects. https://github.com/Toblerity/Shapely

Charles R. Harris, K. Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature* 585 (2020), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Daniel Martin Katz, Dirk Hartung, Lauritz Gerlach, Abhik Jana, and Michael J. Bommarito II au2. 2023. Natural Language Processing in the Legal Domain. arXiv:2302.12039 [cs.CL]

Christoph Molnar. 2022. *Interpretable Machine Learning* (2 ed.). https://christophm.github.io/interpretable-ml-book

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. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

Guido Van Rossum. 2020. *The Python Library Reference, release 3.8.2.* Python Software Foundation.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17 (2020), 261–272. https://doi.org/10.1038/s41592-019-0686-2

Linyi Yang, Jiazheng Li, Pádraig Cunningham, Yue Zhang, Barry Smyth, and Ruihai Dong. 2022. Exploring the Efficacy of Automatically Generated Counterfactuals for Sentiment Analysis. arXiv:2106.15231 [cs.CL]

```
{
  "ITEMID": "001-152311",
  "LANGUAGEISOCODE": "ENG",
  "RESPONDENT": "BGR",
  "BRANCH": "ADMISSIBILITY",
  "DATE": 2015,
  "DOCNAME": "BULGARTSVET-VELINGRAD OOD AND KOPPE v.
      ↪ BULGARIA",
  "IMPORTANCE": "4",
  "CONCLUSION": "Inadmissible",
  "JUDGES": "George Nicolaou; Guido Raimondi; Krzysztof
      ↪ Wojtyczek; Nona Tsotsoria; P ivi Hirvel ;
      ↪ Zdravka Kalaydjieva",
  "TEXT": [
    "CARDINAL . NORP The applicant company, ORG, is a
        ↪ limited liability company created in DATE. It
        ↪  is registered in GPE,
    GPE.
    The applicant, Mr PERSON, is a NORP national who was
        ↪ born in DATE and lives in GPE. The applicants
        ↪  were represented
    before the
    ORG by Mr PERSON, a lawyer practising in GPE. The
        ↪ respondent Government were represented by
        ↪ their Agents, PERSON,
    PERSON
    and PERSON, of ORG.",
    "CARDINAL . ORG, informed of their right to intervene
        ↪  in the proceedings in accordance with
    LAW CARDINAL    CARDINAL of ORG, have not availed
        ↪ themselves of this opportunity.",
    "CARDINAL . NORP The applicant held a MONEY stake in
        ↪ the applicant company. The other MONEY
    was held by Fabrika CARDINAL ORG, a NORP limited
        ↪ liability company wholly owned by the
        ↪ applicant.",
    "CARDINAL . On DATE another company, PERSON ORG (
        ↪    OP   ), a state - owned limited liability
    company,
    was declared insolvent and insolvency proceedings
        ↪ were opened. Following a public auction of
        ↪ part of its property,
    on
    DATE the applicant company acquired several
        ↪ greenhouses. However, in a final decision of
        ↪ CARDINAL DATE the auction
    was cancelled.",
    "CARDINAL . Pending a new auction, on DATE ORG,
        ↪ represented by its trustee in insolvency, and
        ↪  the applicant company signed
    a
    contract pursuant to which the applicant company was
        ↪ appointed guardian of the greenhouses until
        ↪ they were sold.
    The applicant company undertook to maintain the
        ↪ greenhouses  equipment and to carry out
        ↪ urgent repair works. In the
    meantime
    the applicant company was entitled to use the
        ↪ property and in exchange undertook to pay
        ↪ DATE rent.",
    "CARDINAL . In addition to the repairs indicated in
        ↪ the contract of DATE, the applicant company
        ↪ carried out substantial
    works on the greenhouses, which it alleged were
        ↪ necessary for the latters proper
        ↪ functioning. On an unspecified date it
    requested  the trustee in insolvency of ORG to
        ↪ reimburse the costs of those works. Initially
        ↪  the ORG included those costs in
    the list of accepted claims against ORG. Later,
        ↪ however, following an objection by CARDINAL
        ↪ of the other creditors of ORG,
    which was based on the fact that the costs were
        ↪ substantial, had not been approved by the
        ↪ creditors and had been incurred
    after the beginning of the insolvency proceedings,
        ↪ in a decision of DATE the ORG removed this
        ↪ claim from the list.",
    "CARDINAL . In a final decision of DATE, ORG upheld
        ↪ the removal, as it was not satisfied that the
        ↪  additional works had been
    urgent and necessary. Moreover, the works had not
        ↪ been authorised by the ORG and the remaining
        ↪ creditors. ORG considered
    that a creditor such as the applicant company, with a
        ↪  claim which had not been agreed in the
        ↪ insolvency proceedings, could make
    that claim in separate civil proceedings.",
    "CARDINAL . On DATE the applicant company brought a
        ↪ civil claim against ORG and also against ORG,
        ↪  as a representative of the ORG,
    which owned  O R G s  capital. Initially it sought
        ↪ CARDINAL NORP levs (ORG) (the equivalent of
        ↪ MONEY (ORG)), representing part of
    the repair costs. In the course of the proceedings it
        ↪  increased its claim to ORG CARDINAL (
        ↪ approximately EUR CARDINAL),
    claiming that ORG had unjustly
    enriched itself by that amount.
    The applicant company did not seek to have its claim
        ↪ included in the list of accepted claims
        ↪ prepared by the ORG,
    but argued that it was entitled to receive payment
        ↪ prior to the distribution of any amounts
        ↪ obtained through the sale of
    the property of ORG among the remaining creditors.",
    ..... (rest of text not included for space)
  ],
  "VIOLATED_ARTICLES": [],
  "VIOLATED_PARAGRAPHS": [],
  "VIOLATED_BULLETPOINTS": [],
  "NON_VIOLATED_ARTICLES": [],
  "NON_VIOLATED_PARAGRAPHS": [],
  "NON_VIOLATED_BULLETPOINTS": []
}
```

Table 9. Feature Importances

Feature	RF Score	V2 Score	V3 Score	V5 Score	V6 Score	V8 Score	V10 Score	V11 Score	V13 Score	V14 Score
cardinal	0.330456	0.001668	0.006006	0.004671	0.006006	0.000667	0.0	0.000667	0.001001	0.0
hearing	0.080559	0.001001	0.007007	0.002002	0.048048	0.000667	0.0	0.000334	0.000334	0.0
detention	0.067446	0.000000	0.040374	0.078745	0.014348	0.000667	0.0	0.000000	0.001668	0.0
judgment	0.064962	0.000334	0.013347	0.004671	0.048382	0.001001	0.0	0.000334	0.000334	0.0
child	0.039847	0.000000	0.001001	0.002336	0.006673	0.001668	0.0	0.000000	0.000000	0.0
publication	0.026394	0.000000	0.000000	0.000667	0.000000	0.000000	0.0	0.000000	0.000000	0.0
expression	0.023573	0.000000	0.000334	0.000334	0.002336	0.000000	0.0	0.000000	0.000000	0.0
officer	0.022701	0.001668	0.017017	0.001668	0.003337	0.000000	0.0	0.000334	0.000000	0.0
investigation	0.022647	0.003003	0.002336	0.002669	0.006006	0.000000	0.0	0.000000	0.000000	0.0
prosecutor	0.018219	0.000000	0.002336	0.000334	0.004004	0.000000	0.0	0.000000	0.000667	0.0
publish	0.017337	0.000000	0.000667	0.000000	0.001335	0.000000	0.0	0.000000	0.000000	0.0
arrest	0.016848	0.000667	0.001668	0.015682	0.000667	0.000000	0.0	0.000667	0.000667	0.0
son	0.015868	0.001001	0.001335	0.002002	0.002669	0.000334	0.0	0.000000	0.000000	0.0
release	0.014899	0.000334	0.001335	0.019353	0.002002	0.000000	0.0	0.000000	0.000000	0.0
police	0.014748	0.000667	0.013347	0.010677	0.000667	0.000000	0.0	0.001001	0.000000	0.0
office	0.014277	0.000334	0.003337	0.000334	0.003003	0.000000	0.0	0.000000	0.000334	0.0
medical	0.014058	0.000667	0.024358	0.001668	0.004338	0.000000	0.0	0.000000	0.000000	0.0
treatment	0.012002	0.000334	0.014681	0.002002	0.005672	0.000000	0.0	0.000000	0.000334	0.0
body	0.011545	0.001001	0.001335	0.000667	0.001335	0.000000	0.0	0.000000	0.000334	0.0
detain	0.011257	0.000000	0.006340	0.002669	0.004004	0.000000	0.0	0.000000	0.000334	0.0
prison	0.010471	0.000000	0.007007	0.004338	0.004671	0.000334	0.0	0.000000	0.001001	0.0
reputation	0.008903	0.000000	0.000667	0.000334	0.001335	0.000000	0.0	0.000000	0.000334	0.0
beat	0.008876	0.000000	0.006006	0.000000	0.000667	0.000000	0.0	0.000000	0.000000	0.0
investigator	0.008319	0.003670	0.007674	0.008342	0.000667	0.000334	0.0	0.000000	0.000334	0.0
visit	0.007738	0.000000	0.000000	0.000334	0.001668	0.000667	0.0	0.000000	0.000000	0.0
death	0.007546	0.002002	0.001668	0.001335	0.002669	0.000000	0.0	0.000000	0.000334	0.0
inmate	0.007462	0.000000	0.006006	0.000667	0.001335	0.000000	0.0	0.000334	0.000667	0.0
remand	0.006975	0.000000	0.000667	0.008008	0.002002	0.000000	0.0	0.000000	0.000000	0.0
newspaper	0.006939	0.000000	0.000667	0.000334	0.000667	0.000334	0.0	0.000000	0.000000	0.0
abscond	0.006917	0.000000	0.001335	0.006340	0.000334	0.000334	0.0	0.000000	0.000000	0.0
cell	0.006688	0.000000	0.037704	0.001668	0.003337	0.000000	0.0	0.000000	0.002336	0.0
man	0.005961	0.001001	0.002669	0.000000	0.000667	0.000000	0.0	0.000000	0.000000	0.0
shoot	0.005512	0.000334	0.000667	0.000334	0.000334	0.000000	0.0	0.000000	0.000000	0.0
preventive	0.005465	0.000000	0.002002	0.002336	0.000000	0.000000	0.0	0.000667	0.000000	0.0
injury	0.005065	0.000667	0.014014	0.000667	0.002669	0.000000	0.0	0.000000	0.000334	0.0
editor	0.004311	0.000000	0.000334	0.000000	0.000334	0.000000	0.0	0.000000	0.000000	0.0
district	0.003483	0.000000	0.001335	0.000000	0.002002	0.000000	0.0	0.000000	0.000000	0.0
facility	0.002925	0.000000	0.009343	0.000667	0.001335	0.000000	0.0	0.000000	0.000334	0.0
correspondence	0.002874	0.000000	0.000334	0.001001	0.001001	0.000000	0.0	0.000000	0.000000	0.0
search	0.002754	0.000000	0.000334	0.000334	0.001335	0.000000	0.0	0.000000	0.000000	0.0
detainee	0.002739	0.000000	0.008342	0.000667	0.001335	0.000334	0.0	0.000000	0.001668	0.0
bed	0.002739	0.000000	0.006006	0.000000	0.001001	0.000000	0.0	0.000000	0.000334	0.0
political	0.002526	0.000334	0.000667	0.000000	0.002002	0.000000	0.0	0.000000	0.000000	0.0
religious	0.002289	0.000000	0.000000	0.000000	0.000334	0.000000	0.0	0.000000	0.000000	0.0
association	0.002113	0.000334	0.000334	0.000000	0.001335	0.000000	0.0	0.000000	0.000000	0.0
bruise	0.001883	0.000667	0.005672	0.000000	0.000334	0.000000	0.0	0.000000	0.000000	0.0
vehicle	0.001734	0.000334	0.000667	0.000334	0.000334	0.000000	0.0	0.000000	0.000000	0.0
defamation	0.001574	0.000000	0.000334	0.000000	0.001001	0.000000	0.0	0.000000	0.000334	0.0
torture	0.001358	0.000000	0.001668	0.000000	0.001001	0.000000	0.0	0.000000	0.000000	0.0
serviceman	0.001280	0.001335	0.000667	0.000334	0.000334	0.000000	0.0	0.000000	0.000000	0.0
fire	0.001214	0.002002	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.0
unit	0.001192	0.000334	0.002669	0.000667	0.002002	0.000000	0.0	0.000000	0.000000	0.0
military	0.001189	0.002002	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.0

Table 10. Feature Importance Correlation Matrix

Received 3 May 2024

RF Score	V2 Score	V3 Score	V5 Score	V6 Score	V8 Score	V10 Score	V11 Score	V13 Score
1.000000	0.193258	0.236428	0.234349	0.310836	0.341841	0.021916	-0.033511	0.201196
0.193258	1.000000	0.102832	0.005732	-0.032407	-0.005540	-0.101283	-0.036506	-0.034341
0.236428	0.102832	1.000000	0.609695	0.345368	0.082656	-0.088096	-0.028793	0.709854
0.234349	0.005732	0.609695	1.000000	0.257241	0.097298	-0.047236	0.024775	0.468446
0.310836	-0.032407	0.345368	0.257241	1.000000	0.135808	-0.042859	0.031363	0.189178
0.341841	-0.005540	0.082656	0.097298	0.135808	1.000000	-0.044079	0.020974	0.088776
0.021916	-0.101283	-0.088096	-0.047236	-0.042859	-0.044079	1.000000	-0.030901	-0.058819
-0.033511	-0.036506	-0.028793	0.024775	0.031363	0.020974	-0.030901	1.000000	-0.043621
0.201196	-0.034341	0.709854	0.468446	0.189178	0.088776	-0.058819	-0.043621	1.000000