

Noname manuscript No.
(will be inserted by the editor)

European Court of Human Rights Open Data project

Alexandre Quemy

Received: date / Accepted: date

Abstract This paper presents thirteen datasets for binary, multiclass and multilabel classification based on the European Court of Human Rights judgments since its creation. The interest of such datasets is explained through the prism of the researcher, the data scientist, the citizen and the legal practitioner. Contrarily to many datasets, the creation process, from the collection of raw data to the feature transformation, is provided under the form of a collection of fully automated and open-source scripts. It ensures reproducibility and a high level of confidence in the processed data, which is some of the most important issues in data governance nowadays. A first experimental campaign is performed to study some predictability properties and to establish baseline results on popular machine learning algorithms. The results are consistently good across the binary datasets with an accuracy comprised between 75.86% and 98.32% for an average accuracy of 96.45%.

Keywords datasets · European Court of Human Rights · open data · classification

1 Introduction

In this paper, we present the European Court of Human Rights Open Data project (ECHR-OD). It aims at providing up-to-date and complete datasets about the European Court of Human Rights decisions since its creation. To be up-to-date and exhaustive, we developed a fully automated process to regenerate the entire datasets from scratch, starting from the collection of raw documents. As a result, datasets are as complete as they can be in terms of

Alexandre Quemy
IBM, Cracow Software Lab, Poland
Faculty of Computing, Poznań University of Technology, Poznań, Poland
E-mail: aquemy@pl.ibm.com

number of cases. The reproducibility makes it easy to add or remove information in future iterations of the datasets. To be able to check for corrupted data or bias, black swans or outliers, the whole datasets generation process is open-source and versioned.

In a second part, we present the results of a large experimental campaign performed on three flavors of the 13 datasets. We compared 13 standard machine learning algorithms for classification with regards to several performance metrics. Those results provide a baseline for future studies and provide some insights about the interest of some types of features to predict justice decisions. Notably, as for previous studies, we found that case textual description contains interesting elements to predict the (binary) outcome. However, for the first time, we show that the judgement is not as good as purely descriptive features to determine what article a given case is about, such that, for real-life predictive systems, the methodology of previous studies might not be suitable by itself.

Before presenting the project and datasets in Section 3, we discuss in Section 2 the importance of data quality and the multiple issues with current datasets in complex fields such as the legal domain. The creation process is presented in detail in Section 4. Section 5 is dedicated to the experiments on the datasets while Section 6 concludes the paper by discussing the remaining challenges and future work. The paper comes with Supplementary Material available on GitHub¹. It contains additional examples about the data format, as well as all secondary results of the experiments that we omitted due to space constraint.

2 Context and related work

It is now well established that the recent and spectacular results of artificial intelligence, notably with deep learning (LeCun et al. (2015)), are partly due to the availability of data, so called “Big Data”, and the exponential growth of computational power. For a very long time, the bias-variance tradeoff seemed to be an unbeatable problem: complex models reduce the bias but hurt the variance, while simple models lead to high variance. In parallel, the regularization effect of additional data for complex models was also well known as illustrated by Figure 1. The advent of representation learning (Bengio et al. (2013)) allowed to efficiently build extremely complex models with moderate variance by letting the algorithms discovering the interesting “patterns” or “representations” to solve a given problem. However, it requires a considerable amount of data to correctly reduce the variance, and there is now a growing consensus on the fact that data are as important as algorithms. In particular, the quality of a model is bounded by the quality of the data it learns from (Valiant (1984); Vapnik (1999)). The availability and quality of data are thus of primary importance for researchers and practitioners.

¹ https://aqueemy.github.io/ECHR-OD_project_supplementary_material/

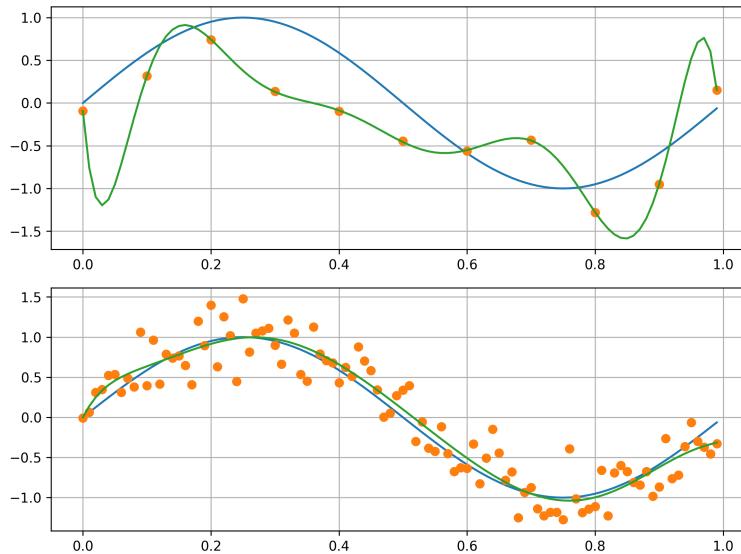


Fig. 1 The regularization effect of data The ground truth (blue) is a $\sin(x)$ modeled by a polynomial of degree 9 (green). On top, with only 11 training points, the model does not approximate correctly the ground truth while at the bottom, with 100 training points, the model error is far lower.

Beyond the scope of pure scientific interest, the data governance, that is to say the lifecycle management of the data, is particularly crucial for our modern societies (Olhede and Wolfe (2018); Tallon (2013); Attard and Brennan (2018)). What data are publicly available? Who produces, manages and manipulates those data? What is the quality of the data? What is the process of collection, curation and transformation? Those are few legitimate questions that a citizen, a company, or an institution may (should?) ask due to the ethical, political, social and legal concerns (Kitchin (2014)). One can mention the recent General Data Protection Regulation² (GDPR), a European Union law with global application, that tries to give a legal framework to address some of the abovementioned questions. Beside privacy and business considerations, the quality of data is at the core of the quality of insights and decisions derived from the data.

2.1 Open Data limits and validity threat

The Open Data movement considers that the data should be freely available and reusable by anyone (Kitchin (2014)). Although it has strong beneficial effects at many levels (for instance, in science (Molloy (2011)), in civic engagement (Kassen (2013)), or in governmental transparency (Janssen et al.

² <https://www.eugdpr.org/>

(2012)), some critics have emerged with regards to social questions (Gurstein (2011); Misuraca and Viscusi (2014)). Aside from those specific considerations, we argue that open data are not enough to insure data quality and to totally handle all the questions mentioned in the previous paragraph. There are many ways data may be unsuitable for making decisions (either solely human based or assisted by any kind of model):

- **Data sparsity and irrelevant information:** a dataset may lack information to correctly make a decision or on the contrary, contain a lot of irrelevant information. Some useful information might not be available at the moment the dataset is constructed. Also, it is hard to know *a priori* what piece of information is useful or not to model or understand a phenomenon. It usually requires specific techniques such as feature selection (Guyon and Elisseeff (2003)) and several studies to obtain a big picture. Having a dataset, even open, without the whole process from the collection to the moment it is publicly available is usually not enough for practical applications.
- **Missing unexpected patterns or learning wrong patterns:** for some reasons, regime change might occur in the data and be learnt or not by a model. How to know if this change is valid or is the result of a problem somewhere between the data collection (e.g. some sensors are not working or being recalibrated) and the data processing (e.g. a bug in the software used to sanitize the data) **without** expert knowledge? Some points may also be outliers for good reasons (e.g. improbable event that *eventually* occurs, often referred as a *black swan*) or bad reasons (e.g. error in processing the data, problem in collecting the data)? In the first case, the models **must** take into account those data, while in the second case, it should be discarded. Open data cannot help, except for obvious cases.
- **Data corruption:** at any stage of the collection, processing and usage, the data may be partly corrupted. It may be hard to determine where and how the data has been corrupted even if the data are open.
- **Biased data:** from the collection process itself to the sanitization choices, bias is introduced. Having access to open data is no help for building better models and algorithms if data are biased from the beginning.

For those reasons, the datasets presented in this paper are accompanied with the full creation process, carefully documented.

Let us illustrate some of those limitations through a concrete case that motivated this project. To validate and compare a new method for classification, we used some datasets provided along with an article published in an open data journal. The reproduction of the experiment was successful but the results with the new methods were inconsistent with our preliminary tests. By digging into the datasets, we found out that many input vectors were empty (up to 70% for some datasets) and the labels were not consistent among those empty vectors. In other words, many situations to classify were described with no information, and this absence of information could not be linked to a specific label. Furthermore, the prevalence in the group of degenerated vectors

was rather high in comparison to the overall prevalence, resulting in relatively good metrics for standard classification algorithms. As the authors did not provide the raw data and the transformation process, there was no way to figure out what exactly caused this problem.

We would like to insist on the last two elements, namely the data corruption and biased data, as they represent a huge validity threat in machine learning-based retrospective studies. The concern is articulated around the following:

1. a typical study focuses on a dataset transformed by a data pipeline. For a new method, a cross-validation is used with the preprocessed data. Results are used to make a comparison with other methods presented in previous studies.
2. in general, comparisons are done solely between methods, without taking into account the data pipeline.
3. however, the data preprocessing operations introduces bias and possibly data corruption, which can drastically affect the final results.

It may be difficult to evaluate how much the effect of data preprocessing affects the final result, especially that the data pipelines are rarely reported. For instance, Crone et al. (2006) notice that algorithm hyperparameter tuning is performed in 16 out of 19 selected publications while only two publications study the impact of data preprocessing.

The data preprocessing impact has been evaluated for multiple algorithms and operators. In Crone et al. (2006), the authors showed that the accuracy obtained by Neural Network, SVM and Decision Trees are significantly impacted by data scaling, sampling and continuous and categorical coding. A correlation link between under and oversampling is also demonstrated. In Nawi et al. (2013), three specific data processing operators has been tested for neural networks. Despite the authors do not provide the results without any data processing, the results show an important accuracy variability between the alternatives, thus implying a data processing impact. Using a representative sample of the available data can also lead to better overall results, as showed by Nalepa et al. (2018). For a more comprehensive view on data processing impact, we refer the reader to Dasu and Johnson (2003). Recently, Quemy (2019) focuses on the data pipeline optimization and found that, between no preprocessing step and a carefully selected data pipeline, the classification error is reduced by 66% in average among four methods (SVM, Decision Tree, Neural Network and Random Forest) and three datasets (Iris, Breast, Wine). More interesting, by changing the data pipeline, it is possible to obtain any possible ranking of the methods with regard to the error rate.

2.2 Related work in legal analytics

The legal environment is a *messy concept* (Rissland (2006)) that intrinsically poses some of the most challenging problems for the artificial intelligence research community: grey areas of interpretation, many exceptions,

non-stationarity, presence of deductive and inductive reasoning, non-classical logic, etc. For some years and in several areas of the law, some "quantitative" approaches have been developed, based on the use of more or less explicit mathematical models. With the availability of massive data, those trends have been accented and brand-new opportunities are emerging at a sustained pace. Among the stakes of those studies, one can mention a better understanding of the legal system and the consequences of some decisions on the society, but also the possibility to decrease the mass of litigations in a context of cost rationalization. For a survey on legal analytics methods, we refer the reader to Quemy (2017). In Quemy (2017), the author also defines some practical problems in the field of legal analytics. First, the prediction problem consists in determining the outcome of a trial given some facts about the specific case and some knowledge about the legal environment. The second problem consists in building a legal justification, knowing the legal environment.

The justification problem should not be misunderstood with the model explainability problem in machine learning. Understanding how a model make a prediction is certainly useful to generate legal justifications, however it is not enough. Many studies modeled justice decisions solely based on the estimation of the judge ideology (Spitzer and Cohen (1994); Segal and Cover (1989); Segal et al. (1995); Quinn and Martin (2002)). The way the model makes decisions is rather clear, however, the model is unable to provide a satisfying legal justification. Those considerations are highly connected to a major debate among the legal practitioners, namely legalism to realism: are the judges objectively applying a method contained in the text (legalism) or do they create their own interpretation (realism). The feasibility of a solution to the justification problem largely depends on the answer to this debate. Some discussions on the topic can be found in Posner (2010); Klerman (2012); Tamanaha (2012); Leiter (2010).

By releasing large datasets, with several flavors based on different types of features, along with the whole tunable preprocessing pipeline, we hope to gain a better understanding on how justice decisions are taken, what elements are useful for a prediction, and to quantify the balance between realism and legalism. We also hope to make a step toward solving the justification problem.

Predicting the outcome of a justice case is challenging, even for the best legal experts: 67.4% and 58% accuracy, respectively for the judges and whole case decision, is found in Ruger et al. (2004) for the Supreme Court of the United States (SCOTUS). Using crowds, the Fantasy Scotus³ project reached respectively 85.20% and 84.85% correct predictions.

The Supreme Court of the United States (SCOTUS) has been widely studied, notably through the SCOTUS database⁴ (Katz et al. (2017); Martin et al. (2004); Guimerà and Sales-Pardo (2011)). This database is composed of structured information about every case since the creation of the court but no textual information from the opinions. The opinions and other related tex-

³ <https://fantasyscotus.lexpredict.com/>

⁴ <http://scdb.wustl.edu/>

tual documents also have been studied separately for SCOTUS (Islam et al. (2016); Lauderdale and Clark (2014); Sim et al. (2014)). Conversely, very little if no similar work has been done in Europe. As far as we know, the only predictive model was using the textual information only (Aletras et al. (2016)), despite more structured information is publicly available on HUDOC⁵. Using NLP techniques, the authors of Aletras et al. (2016) achieve 79% accuracy to predict the decisions of the European Court of Human Rights (ECHR). They make the hypothesis that the textual content of the European Convention of the Human Rights holds hints that will influence the decision of the Judge. They extracted from the judgement documents the top 2000 N-grams, calculated their similarity matrix based on cosine measure and partition this matrix using Spectral Clustering to obtain a set of interpretable topics. The binary prediction was made using an SVM with linear kernel. Contrary to the previous studies on SCOTUS, they found out that the formal facts are the most important predictive factors, which tend to favor legalism.

The data used in Aletras et al. (2016) are far from being exhaustive: 3 articles considered (3, 6 and 8) with respectively 250, 80 and 254 cases per article. For many data-driven algorithms, this might be too little to build a correct model. As far as we know, the project presented in this paper already provides the largest existing legal datasets directly consumable by machine learning algorithms. In particular, it includes several types of features: purely descriptive and textual.

2.3 On the availability of legal data

One of the prominent domains of application for deep learning is computer vision. In this area, it is relatively easy to obtain new data, even if the process of manually labelling training data may be laborious. Techniques like Data Augmentation allow to generate artificially new data by slightly modifying existing examples (Dyk and Meng (2001)). For instance, for hand-written recognition, one may add some small perturbations or apply transformations to an existing example, such as rotation, zoom, gaussian noise, etc. Those techniques are efficient at providing useful new examples but rely on an implicit assumption: the solution's behavior changes continuously with the initial data. In other words, slightly modifying the picture of an 8 by a small rotation or distortion still results in an 8. However, in many other fields, a small change in the data may result in a totally different outcome such that one cannot use Data Augmentation to artificially grow her dataset. In general, the fields where Data Augmentation is not applicable are more complex, require more information to process or require a sophisticated or *conscious* reasoning before being able to give an answer. Anyone can recognize a cat from a horse without processing any additional data than the picture itself, without elaborating a complex and explicit reasoning. Conversely, deciding if someone is guilty w.r.t. some

⁵ <https://hudoc.echr.coe.int/eng>

available information and the current legal environment is not as natural as recognizing a cat, even for the best legal experts.

We would like to draw the attention on the fact that, it is not because those fields are more complex for humans that nowadays artificial intelligence techniques cannot perform better than humans: medical diagnosis is a complex field requiring expertise and explicit reasoning, however humans are regularly beaten by the machine (Tiwari et al. (2016); Yu et al. (2016); Patel et al. (2016)). That said, in many complex fields, data augmentation techniques cannot be used, access to the data may be difficult, the data itself can be limited and, when open, the data are provided already curated without access to the curation process. ECHR-OD project has been created with those considerations in mind and we will see with the experiments that despite being exhaustive w.r.t. number of cases, more data would have helped the models to perform better.

3 ECHR-OD in brief

ECHR-OD project aims at providing exhaustive and high-quality database and datasets for diverse problems, based on the European Court of Human Rights documents available on HUDOC. It appears important to us 1) to draw the attention of researchers on this domain that has important consequences on the society, 2) to provide a similar and more complete database for the European Union as it already exists in the United States, notably because the law systems are different in both sides of the Atlantic. The project is composed of five components:

1. **Main website:** <https://echr-opendata.eu>
2. **Download mirror:** <https://osf.io/52rhg/>
3. **Creation process:** https://github.com/aquemy/ECHR-OD_process
4. **Website sources:** https://github.com/aquemy/ECHR-OD_website
5. **Data loader in python:** https://github.com/aquemy/ECHR-OD_loader

ECHR-OD is guided by three core values: **reusability**, **quality** and **availability**. To reach those objectives,

- each version of the datasets is carefully versioned and publicly available, including the intermediate files,
- the integrality of the process and files produced are careful documented,
- the scripts to retrieve the raw documents and build the datasets from scratch are open-source and carefully versioned to maximize reproducibility and trust,
- no data is manipulated by hand at any stage of the creation process.

At the submission date, the project offers 13 datasets for the classification problem. Datasets for other problems such as structured predictions will be available in the future. The datasets are available under the **Open Database**

Licence (ODbL)⁶⁷ which guarantees the rights to copy, distribute and use the database, to produce work from the database and to modify, transform and build upon the database. The creation scripts and website sources are provided under **MIT Licence**.

3.1 Datasets description

In machine learning, the problem of classification consists in finding a mapping from an input vector space \mathcal{X} to a discrete decision space \mathcal{Y} using a set of examples. The binary classification problem is a special case of the multiclass such that \mathcal{Y} has only two elements, while in multilabel classification, each element of \mathcal{X} can have several labels. It is often viewed as an approximation problem s.t. we want to find an estimator \bar{J} of an unknown mapping J available only through a sample called *training set*. A training set (\mathbf{X}, \mathbf{y}) consists of N input vectors $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ and their associated correct class $\mathbf{y} = \{J(\mathbf{x}_i) + \varepsilon\}_{i=1}^N$, possibly distorted by some noise ε . Let $\mathcal{J}(\mathcal{X}, \mathcal{Y})$ be the class of mappings from \mathcal{X} to \mathcal{Y} . Solving an instance of the classification problem consists in minimizing the classification error:

$$J^* = \arg \min_{\bar{J} \in \mathcal{J}(\mathcal{X}, \mathcal{Y})} \sum_{\mathbf{x} \in \mathcal{X}} \mathbb{I}_{\{J(\mathbf{x}) \neq \bar{J}(\mathbf{x})\}} \quad (1)$$

From the HUDOC database and judgment files, we created several datasets for three variants of the classification problem: binary classification, multiclass classification and multilabel classification. There are 11 datasets for binary, one for multiclass, and one for multilabel classification.

Each dataset comes in different flavors based on descriptive features and Bag-of-Words and TF-IDF representations:

1. **Descriptive features:** structured features retrieved from HUDOC or deduced from the judgment document,
2. **Bag-of-Words representation:** based on the top 5000 tokens (normalized n -grams for $n \in \{1, 2, 3, 4\}$),
3. **TF-IDF representation:** idem but with a TF-IDF transformation to weight the tokens,
4. **Descriptive features + Bag-of-Words:** combination of both sets of features,
5. **Descriptive features + TF-IDF:** combination of both sets of features.

Those different representations exist to study the respective importance of descriptive and textual features in the predictive models build upon the datasets.

For binary classification, the label corresponds to a violation or no violation of a specific article. Each of the 11 datasets corresponds to a specific article. We kept only the articles such that there are at least 100 cases with a clear

⁶ Summary: <https://opendatacommons.org/licenses/odbl/summary/>

⁷ Full-text: <https://opendatacommons.org/licenses/odbl/1.0/>

Table 1 Datasets description for binary classification.

	# cases	min #features	max #features	avg #features	prevalence
Article 1	951	131	2834	1183.47	0.93
Article 2	1124	44	3501	2103.45	0.90
Article 3	2573	160	3871	1490.75	0.89
Article 5	2292	200	3656	1479.60	0.91
Article 6	6891	46	3168	1117.66	0.89
Article 8	1289	179	3685	1466.52	0.73
Article 10	560	49	3440	1657.22	0.75
Article 11	213	293	3758	1607.96	0.85
Article 13	1090	44	2908	1309.33	0.91
Article 34	136	490	3168	1726.78	0.64
Article p1	1301	266	2692	1187.96	0.86

The columns min, max, avg #features indicate the minimal, maximal and average number of features in the dataset cases for the representation "descriptive features + Bag-of-Words representation".

output (see Section 4.2 for additional details) without consideration on the prevalence. Notice that a same case can appear in two datasets if it has in his conclusion two elements about a different article. A basic description of those datasets is given by Table 1.

For multiclass, there is a total of 18 different classes (the number of different articles multiplied by two possible decisions: violation or no violation). To create the multiclass dataset, we aggregate the different binary classification datasets by removing the cases present in several datasets. For this reason, articles 13 and 34 are not included since they had less than 100 cases after this step. We did not simply merge the binary classification datasets for the remaining cases. The processing part consisting in the creation of the BoW and TF-IDF representations is based on the 5000 most frequent n -grams among the corpus of judgments (see Section 4.5 documents for additional details). As the most frequent n -gram changes depending on the corpus, the BoW representation of a given case is different in the binary, multiclass and multilabel dataset. The descriptive features are, however, not modified.

For multilabel classification, there are 22 different labels and the main difference with the multiclass is that there is no need to remove cases that appear in multiple binary classification datasets. The labels are simply stacked. Table 3 summarizes the dataset composition, Figures 2 and 3 shows the labels repartition among the multiclass and multilabel datasets, and Figure 4 provides the histogram of label numbers and cases per label.

The final format to encode the case information is close to the LIBSVM format. Each couple (**variable**, **value**) is encoded by **<variable_id>:<value_id>** with the specificity that the **<value_id>** is not encoded per variable but globally. For instance, **0:7201** corresponds to variable **itemid=001-170361**. The encoding for the variables can be found in **variables_descriptive.json** and the encoding for the couples (**variable**, **value**) in **features_descriptive.json**. The format for the mapping **features_descriptive.json** is "**<variable>=<value>**" : **<id>**

Table 2 Dataset description for the multiclass dataset.

	# cases	violation	no-violation	prevalence
Article 1	310	280 (0.039)	30 (0.004)	0.90
Article 2	267	230 (0.032)	37 (0.005)	0.86
Article 3	775	676 (0.095)	99 (0.014)	0.87
Article 5	791	689 (0.097)	102 (0.014)	0.87
Article 6	3491	3143 (0.441)	348 (0.049)	0.90
Article 8	623	457 (0.064)	166 (0.023)	0.73
Article 10	413	315 (0.044)	98 (0.014)	0.76
Article 11	110	92 (0.013)	18 (0.003)	0.84
Article p1	353	294 (0.041)	59 (0.008)	0.83

For each article is indicated the number of cases, the number of cases labeled as violated and not violated with in parenthesis the prevalence w.r.t. the **whole** dataset.

Table 3 Dataset description for the multilabel dataset.

	# cases	violation	no-violation
Article 1	951	882 (0.082)	69 (0.006)
Article 2	1124	1017 (0.095)	107 (0.010)
Article 3	2573	2295 (0.214)	278 (0.026)
Article 5	2292	2081 (0.194)	211 (0.020)
Article 6	6891	6152 (0.574)	739 (0.069)
Article 8	1289	940 (0.088)	349 (0.033)
Article 10	560	418 (0.039)	142 (0.013)
Article 11	213	180 (0.017)	33 (0.003)
Article 13	1090	997 (0.093)	93 (0.009)
Article 34	136	87 (0.008)	49 (0.005)
Article p1	1301	1120 (0.105)	181 (0.017)

For each article is indicated the number of cases, the number of cases labeled as violated and not violated with in parenthesis the prevalence w.r.t. the **whole** dataset.

Table 4 Files contained in a dataset.

descriptive.txt	Descriptive features only.
BoW.txt	Bag-of-Word representation only.
TF_IDF.txt	TF-IDF representation only.
descriptive+BoW.txt	Descriptive features and Bag-of-Words.
descriptive+TF_IDF.txt	Descriptive features and TF-IDF.
outcomes.txt	Contain the labels of the datasets.
features_descriptive.json	Mapping between feature and numerical id.
features_text.json	Mapping between <i>n</i> -grams and numerical id.
outcomes_variables.json	Mapping between labels and numerical id.
variables_descriptive.json	Mapping between descriptive variable and numerical id.
statistics_datasets.json	Contain some statistics about the dataset.

or "<variable>_has_<value>":<id> if the variable is a set of elements. For instance, the variable **parties** has two elements and is encoded by 19. Having "BASYUK" in the parties of a case is encoded by "**parties_has_BASYUK**": 109712 and thus, the case description contains 19:109712 and 19:X where X

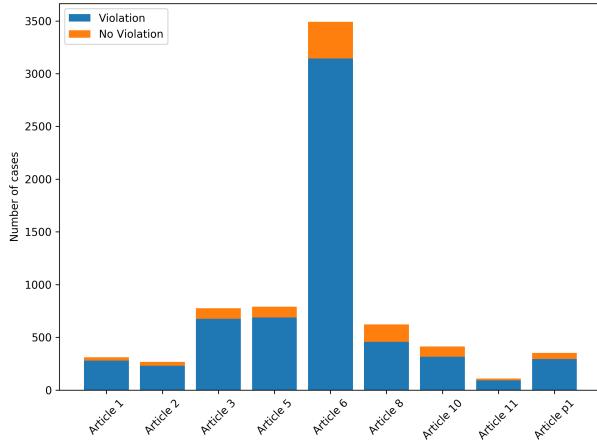


Fig. 2 Number of cases depending on the article and the outcome for the multiclass dataset.

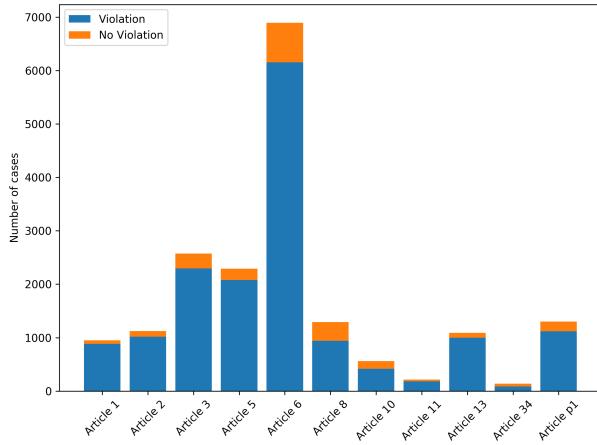


Fig. 3 Number of cases depending on the article and the outcome for the multilabel dataset.

is the id for the second party. As the id is global, having the variable id in prefix is redundant. Notice that it has at least three advantages. First, there is no need to look in the global dictionary and parse the corresponding key to know the encoded variable. Second, some algorithms might want a pair (**variable**, **value**) (e.g. Decision Tree) while others can work with global tokens (e.g. Neural Network). Finally, it makes it easier to re-encode the cases with a specific encoder (e.g. binary, Helmert, Backward Difference, etc.).

Regarding the Bag-of-Words representation, each n -gram is turned into a variable such that when a case judgment contains a specific token, the final representation contains `<token_id>:<occurrences>`. For instance, assuming

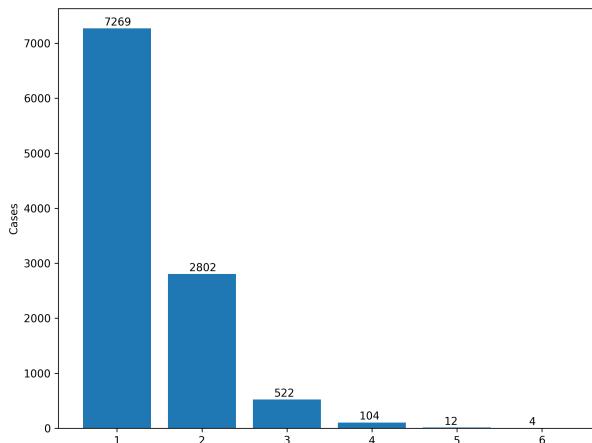


Fig. 4 Number of cases depending on the number of labels for the multilabel dataset.

that the 2-gram "find_guilty" is encoded by 128210 and appears five times in a judgment, the case description will contain 128210:5. For TF-IDF representation, <occurrences> is replaced by the specific weight for this token in the document given the whole dataset.

A Python library to load and manipulate the datasets have been developed and is available at https://github.com/aquemy/ECHR-OD_loader.

4 Creation process

In this section, we describe in detail the dataset generation process from scratch. The datasets are based on the raw documents and information available publicly in HUDOC database. The process is broken down into several steps as illustrated by Figure 5:

1. **get_cases_info.py:** Retrieve the list and basic information about cases from HUDOC,
2. **filter_cases.py:** Remove unwanted, inconsistent, ambiguous or difficult-to-process cases,
3. **get_documents.py:** Download the judgment documents for the filtered list of cases,
4. **preprocess_documents.py:** Analyze the raw judgments to construct a JSON nested structures representing the paragraphs,
5. **process_documents.py:** Normalize the documents and generate a Bag-of-Words and TF-IDF representation,
6. **generate_datasets.py:** Combine all the information to generate several datasets.

The integrality of this process is wrapped into a script **build.py**. This script has some parameters such as the output folder name but also the number

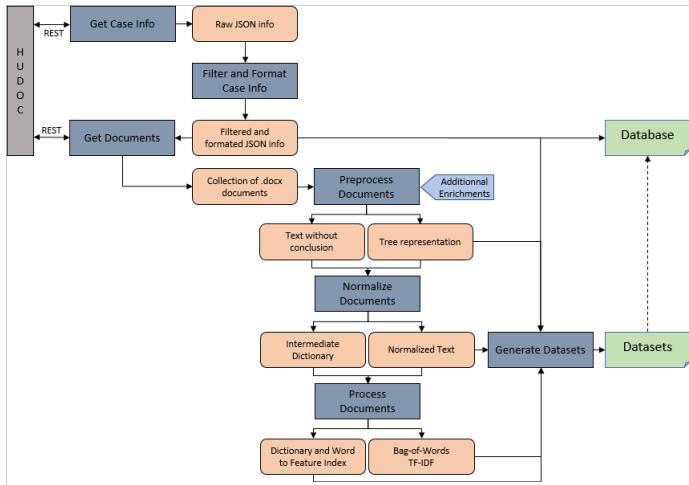


Fig. 5 ECHR-OD datasets creation process.

of tokens to take into consideration during the generation of Bag-of-Words representation. This allows anyone to generate slightly modified versions of the datasets and to experiment with them.

4.1 Retrieving cases

Using HUDOC API, basic information about all entries are retrieved and saved in JSON files. Those entries contain several keys that are listed on top of the script `get_case_info.py`. Among them can be found the case name, the language used, or the conclusion in natural language. See Appendix 6 for an example of a case description.

4.2 Filtering cases

To ensure the quality and usability of the datasets, we filter the cases as follows:

1. We keep only cases in English,
2. We keep only cases with a judgment document,
3. We remove the cases without an attached judgment document,
4. We keep only the cases with a clear conclusion (i.e. containing at least one occurrence of “(no) violation”),
5. We remove a specific list of cases hard to process (three cases for this version of the datasets).

During this step, we also parse and format some raw information: the parties are extracted from the case title and many raw strings are broken down into lists. In particular, the string listing the articles discussed in a case are

turned into a list and the conclusion string into a slightly more complex JSON object. For instance, the string `Violation of Art. 6-1;No violation of P1-1;Pecuniary damage - claim dismissed;Non-pecuniary damage - financial award` becomes the list of elements described in Appendix 6.

In general, each item in the conclusion can have the following elements:

1. **article**: number of the concerned article if applicable,
2. **details**: list of additional information (paragraph or aspect of the article),
3. **element**: part of the raw string describing the item,
4. **mentions**: diverse mentions (quantifier s.a. 'moderate', country...),
5. **type**: violation, no violation or other.

Some representative examples are provided in Appendix 6.

Finally, on top of saving the case information in a JSON file, we output a JSON file for each unique article with at least 100 associated cases⁸.

Additionally, some basic statistics about the attributes are generated, e.g. the cardinality of the domain and the density (i.e. the cardinality over the total number of cases). For instance, the attribute `itemid` is unique and thus, as expected, its density is 1:

```
"itemid": {
  "cardinal": 12075,
  "density": 1.0
}
```

In comparison, the field `article_` (raw string containing a list of articles discussed in a case) and `article` (its parsed and formatted counterpart) have a density of respectively 25% and 1%. This illustrates the interest of our processing method: using the raw string, the article attribute is far more unique than it should be. In reality, there are about 130 different values that are really used across the datasets.

```
"articles_": {
  "cardinal": 3104,
  "density": 0.2570600414078675
}

"article": {
  "cardinal": 131,
  "density": 0.010848861283643893
}
```

4.3 Getting documents

During this phase, we only download the judgment documents in Microsoft Word format using HUDOC API.

⁸ This constant is a parameter of the script and can thus be modified for additional experimentations.

4.4 Preprocessing documents

The preprocessing step consists in parsing the MS Word document to extract additional information and create a tree structure of the judgment file. It outputs two files for each case:

1. **<case_id>.parsed.json**: same JSON document as produced by **filter_cases.py** with additional information.
2. **<case_id>.text_without_conclusion.txt**: full judgment text without the conclusion. It is meant to be used for creating the BoW and TF-IDF representations.

To the previous information, we add the field `decision_body` with the list of persons involved into the decision, including their role. See Appendix 6 for an example.

The most important addition to the case info is the tree representation of the whole judgment document under the field `content`. The content is described in an ordered list where each element has two fields: 1) `content` to describe the element (paragraph text or title) and 2) `elements` that represents a list of sub-elements. For a better understanding, see the example in Appendix 6. This representation eases the identification of some specific sections or paragraphs.

4.5 Normalizing documents

During this step, the documents **<case_id>.text_without_conclusion.txt** are normalized as follows:

- Tokenization,
- Stopwords removal,
- Part-of-Speech tagging followed by a lemmatization,
- n -gram generation for $n \in \{1, 2, 3, 4\}$,

The output files are named **<case_id>.normalized.txt**.

4.6 Processing documents

This step uses Gensim (Řehůřek and Sojka (2010)) to construct a dictionary of the 5000 most common tokens based on the normalized documents (the dictionary is created per dataset) and outputs the Bag-of-Words and TF-IDF representations for each document. The naming convention is **<case_id>.bow.txt** and **<case_id>.tfidf.txt**. Additionally, **feature_to_id.dict** and **dictionary.dict** contain the mapping between tokens and id, respectively in JSON and in a compressed format used by Gensim. The number of tokens to use in the dictionary is a parameter of the script.

4.7 Generating datasets

The final step consists in producing the dataset and related files. See Table 4 for the list of output files. The feature id of the BoW and TF-IDF parts are **not** the same as those obtained during the processing phase. More precisely, they are shifted by the number of descriptive features.

We remove the cases with no clear output. For instance, it is possible to have a violation of a certain aspect of a given article but no violation of another aspect of the same article. In the future, we will consider a lower label level than the article.

5 Experiments

In this section, we perform a first campaign of experiments on each of the produced datasets. The goals are twofold: studying the predictability offered by those datasets and their different flavors, and providing a first baseline by testing the most popular machine learning algorithms for classification. All the experiments are implemented using Scikit-Learn (Pedregosa et al. (2011)). We split the experiments into three categories: binary, multiclass and multilabel classifications, mostly because the evaluation metrics and their interpretation differ. All the experiments and scripts to analyze the results and generate the plots and tables are open-source and available on a separated GitHub repository⁹ for replication.

5.1 Binary classification

We are interested in answering four questions: 1) what is the predictive power of the datasets, 2) are all the articles equal w.r.t. predictability, 3) are some methods performing significantly better than others, and 3) are all dataset flavors equal w.r.t. predictability?

5.1.1 Protocol

We compared 13 standard classification methods: AdaBoost with Decision Tree, Bagging with Decision Tree, Naive Bayes (Bernoulli and Multinomial), Decision Tree, Ensemble Extra Tree, Extra Tree, Gradient Boosting, K-Neighbors, SVM (linear, RBF), Neural Network (Multilayer Perceptron) and Random Forest.

For each article, we used three flavors: descriptive features only, bag-of-words only, and descriptive features combined to bag-of-words. For each method, each article and each flavor, we performed a 10-fold cross-validation with stratified sample, for a total of 429 validation procedures. Due to this important

⁹ https://github.com/aquemy/ECHR-OD_predictions

amount of experimental settings, we discarded the TF-IDF flavors. For the same reason, we did not perform any hyperparameter tuning at this stage.

To evaluate the performances, we reported some standard performance indicators: accuracy, F_1 -score and Matthews correlation coefficient (MCC). Denoting by TP the number of true positives, TN the true negatives, FP the false positives and FN the false negative, those metrics are defined by:

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$F_1 = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}$$

The accuracy, F_1 -score and MCC respectively belongs to $[0, 1]$, $[0, 1]$ and $[-1, 1]$. The closer to 1, the better it is. F_1 -score and MCC take into account false positive and false negatives. Furthermore, MCC has been shown to be more informative than other metrics derived from the confusion matrix Chicco (2017), in particular with imbalanced datasets.

Additionally, we report the learning curves to study the limit of the model space for each method. The learning curves are obtained by plotting the accuracy depending on the training set size, for both the training and the test sets. The learning curves help to understand if a model underfit or overfit and thus, shape future axis of improvements to build better classifiers.

To find out what type of features are the most important w.r.t. predictability, we used a Wilcoxon signed-rank test at 5% to compare the accuracy obtained on Bag-of-Words representation to the one obtained on the Bag-of-Words combined with the descriptive features. Wilcoxon signed-rank test is a non-parametric paired difference test. Given two paired samples, the null hypothesis assumes the difference between the pairs follows a symmetric distribution around zero. The test is used to determine if the changes in the accuracy is significant when the descriptive features are added to the textual features.

5.1.2 Results

Table 5 shows the best accuracy obtained for each article as well as the method and the flavor of the dataset. For all articles, the best accuracy obtained is higher than the prevalence. The method performing the best is linear SVM, obtaining the best results on 4 out of 11 articles. Gradient Boosting accounts for 3 out 11 articles and Ensemble Extra Tree accounts for 2 articles. The standard deviation is rather low with 1% up to 4%, at the exception of article 34 with 9%. The accuracy ranges from 75.86% to 98.32% with an average of 94.43%. The micro-average that ponders each result by the dataset size

Table 5 Best accuracy obtained for each article.

Article	Accuracy	Method	Flavor
Article 1	0.9832 (0.01)	Linear SVC	Descriptive features and Bag-of-Words
Article 2	0.9760 (0.02)	Linear SVC	Descriptive features and Bag-of-Words
Article 3	0.9588 (0.01)	BaggingClassifier	Descriptive features and Bag-of-Words
Article 5	0.9651 (0.01)	Gradient Boosting	Descriptive features and Bag-of-Words
Article 6	0.9721 (0.01)	Linear SVC	Descriptive features and Bag-of-Words
Article 8	0.9542 (0.03)	Gradient Boosting	Descriptive features and Bag-of-Words
Article 10	0.9392 (0.04)	Ensemble Extra Tree	Bag-of-Words only
Article 11	0.9671 (0.03)	Ensemble Extra Tree	Descriptive features and Bag-of-Words
Article 13	0.9450 (0.02)	Linear SVC	Descriptive features only
Article 34	0.7586 (0.09)	AdaBoost	Descriptive features only
Article p1	0.9685 (0.02)	Gradient Boosting	Descriptive features and Bag-of-Words
Average	0.9443		
Micro average	0.9644		

is 96.44%. In general, the datasets with higher accuracy are larger and more imbalanced. The datasets being highly imbalanced, with a prevalence from 0.64 to 0.93, other metrics may be more suitable to appreciate the quality of the results. In particular, the micro-average could simply be higher due to the class imbalance rather than the availability of data.

Regarding the flavor, 8 out 10 best results are obtained on descriptive features combined to bag-of-words. Bag-of-words only is the best flavor for article 10 and descriptive features only for article 13 and article 34. This seems to indicate that combining information from different sources are improving the overall results.

Figure 6 displays the normalized confusion matrix for each line of the Table 5. The normalization is done per line and allows to quickly appreciate how the true predictions are balanced for both classes. As expected by the prevalence, true negatives are extremely high, ranging from 0.82 to 1.00 with an average of 97.18. On the contrary, the true positive rate is lower ranging from 0.47 to 0.91. For most articles, the true positive rate is higher than 80% and is lower than 50% only for article 34.

In addition, we provide the Matthew Correlation Coefficient in Table 6 and the F₁-score in Supplementary Material. The F₁-score is weighted by the class support to account for class imbalance. The Matthew Correlation Coefficient ranges from 0.4918 on article 34 to 0.8829 on article 10. The best score is not obtained by the same article as for accuracy (article 10 achieved 93% accuracy, below the average). Interestingly, the MCC reveals that the performances on article 34 are rather poor in comparison to the other articles and close to those of article 13. Surprisingly, the best method is not linear SVC anymore (best on 3 articles) but Gradient Boosting (best on 4 articles). While the descriptive features were returning the best results for two articles, according the Matthew Correlation Coefficient, it reaches the best score only for article 34.

Once again, the micro-average is higher than the macro-average. As the MCC and the weighted F₁-score take into account class imbalance, it supports

Fig. 6 Normalized Confusion Matrices for the best methods as described by Table 5.

Table 6 Best Matthews Correlation Coefficient and F1 score obtained for each article. The flavor and method achieving the best score for both metrics are similar for every article.

Article	MCC	Method	Flavor
Article 1	0.8654	Linear SVC	Descriptive features and Bag-of-Words
Article 2	0.8609	Linear SVC	Descriptive features and Bag-of-Words
Article 3	0.7714	BaggingClassifier	Descriptive features and Bag-of-Words
Article 5	0.7824	Gradient Boosting	Descriptive features and Bag-of-Words
Article 6	0.8488	Linear SVC	Descriptive features and Bag-of-Words
Article 8	0.8829	Gradient Boosting	Descriptive features and Bag-of-Words
Article 10	0.8411	Gradient Boosting	Bag-of-Words only
Article 11	0.8801	Ensemble Extra Tree	Descriptive features and Bag-of-Words
Article 13	0.5770	Ensemble Extra Tree	Bag-of-Words only
Article 34	0.4918	AdaBoost	Descriptive features only
Article p1	0.8656	Gradient Boosting	Descriptive features and Bag-of-Words
Average	0.7879		
Micro average	0.8163		

Table 7 Overall ranking of methods according to the average accuracy obtained on every article.

Method	Accuracy	Micro Accuracy	Rank
Ensemble Extra Tree	0.9420	0.9627	1
Linear SVC	0.9390	0.9618	2
Random Forest	0.9376	0.9618	3
BaggingClassifier	0.9319	0.9599	4
Gradient Boosting	0.9309	0.9609	5
AdaBoost	0.9284	0.9488	6
Neural Net	0.9273	0.9535	7
Decision Tree	0.9181	0.9419	8
Extra Tree	0.8995	0.9275	9
Multinomial Naive Bayes	0.8743	0.8907	10
Bernoulli Naive Bayes	0.8734	0.8891	11
K-Neighbors	0.8670	0.8997	12
RBF SVC	0.8419	0.8778	13
Average	0.9086	0.9335	

the idea that adding more cases to the training set could still improve the result of those classifiers. This will be confirmed by looking at the learning curves.

Table 7 ranks the methods according to the average accuracy performed on all articles. For each article and method, we kept only the best accuracy among the three dataset flavors. Surprisingly, Linear SVC or Gradient Boosting are not the best method with a respective rank of 2 and 5, but Ensemble Extra Tree. Random Forest and Bagging with Decision Tree are respectively second and third while they never achieved the best result on any article. It simply indicates that those methods are more consistent across the datasets than Linear SVM or Gradient Boosting. This can be confirmed by the detailed results per article provided in the Supplementary Material.

Figure 7 displays the learning curves obtained for the best methods described by Table 5. The training error becomes (near) zero on every instance after only few cases, except for article 13 and 34. The test error converges rather fast and remains relatively far from the training error, synonym of high

bias. Those two elements indicate underfitting. Usually, more training examples would help but as the datasets are exhaustive w.r.t. the European Court of Human Rights cases, this is not possible. As a result, simpler model space has to be investigated as well as, in general, hyperparameter tuning. An exploratory analysis of the datasets may also help in removing some noise and finding the best predictors.

On article 13 and 34, the bias is also high, and variance relatively higher than for the other articles, clearly indicating the worst possible case. Again, adding more examples is not an option. However, if we assume that the process of deciding if there is a violation or not is the same independently of the article, a solution might be transfer learning to leverage what is learnable from the other articles. We let this research axis for future work.

Finally, we used a Wilcoxon signed-rank test at 5% to compare the accuracy obtained on Bag-of-Words representation to the one obtained on the Bag-of-Words combined with the descriptive features. The difference between the sample has been found to be significant only for article 6 and article 8. The best result of column `BoW` is improved in the column `both` for every article. However, statistically, for a given method, adding descriptive features does not improve the result.

Additionally, we performed the test per method. The result is significant for any method.

In conclusion, the datasets demonstrated a strong predictability power. Apart from article 13 and 34, each article seems to provide similar results independently of the relatively different prevalence. If the accuracy is rather high w.r.t. the prevalence, more informative metrics such as MCC and F_1 scores shows that there are still margins of improvements. Hyperparameter tuning is an obvious way to go, and this preliminary work have shown that good candidates for fine tuning are Ensemble Extra Tree, Linear SVM and Gradient Boosting. This experimental campaign has demonstrated that the textual information provides better results than descriptive features alone, but the addition of those descriptive feature improve in general the final **best** result. We emphasize the best (obtained among all methods) because for a given method and any article, adding the descriptive feature are not significantly improving the results. Another way of improving the results is to tune the different phases of the dataset generations. In particular, preliminary work in Quemy (2019) have shown that 5000 tokens and 4-grams might not be enough to take the best out of the documents. It might seem surprising, but the justice language is codified and standardized in a way that n -grams for large n might be good predictors for the outcome.

5.2 Multiclass classification

In this section, we are interested in quantifying the capacity of standard machine learning algorithms to deal with the multiclass dataset. In the previous

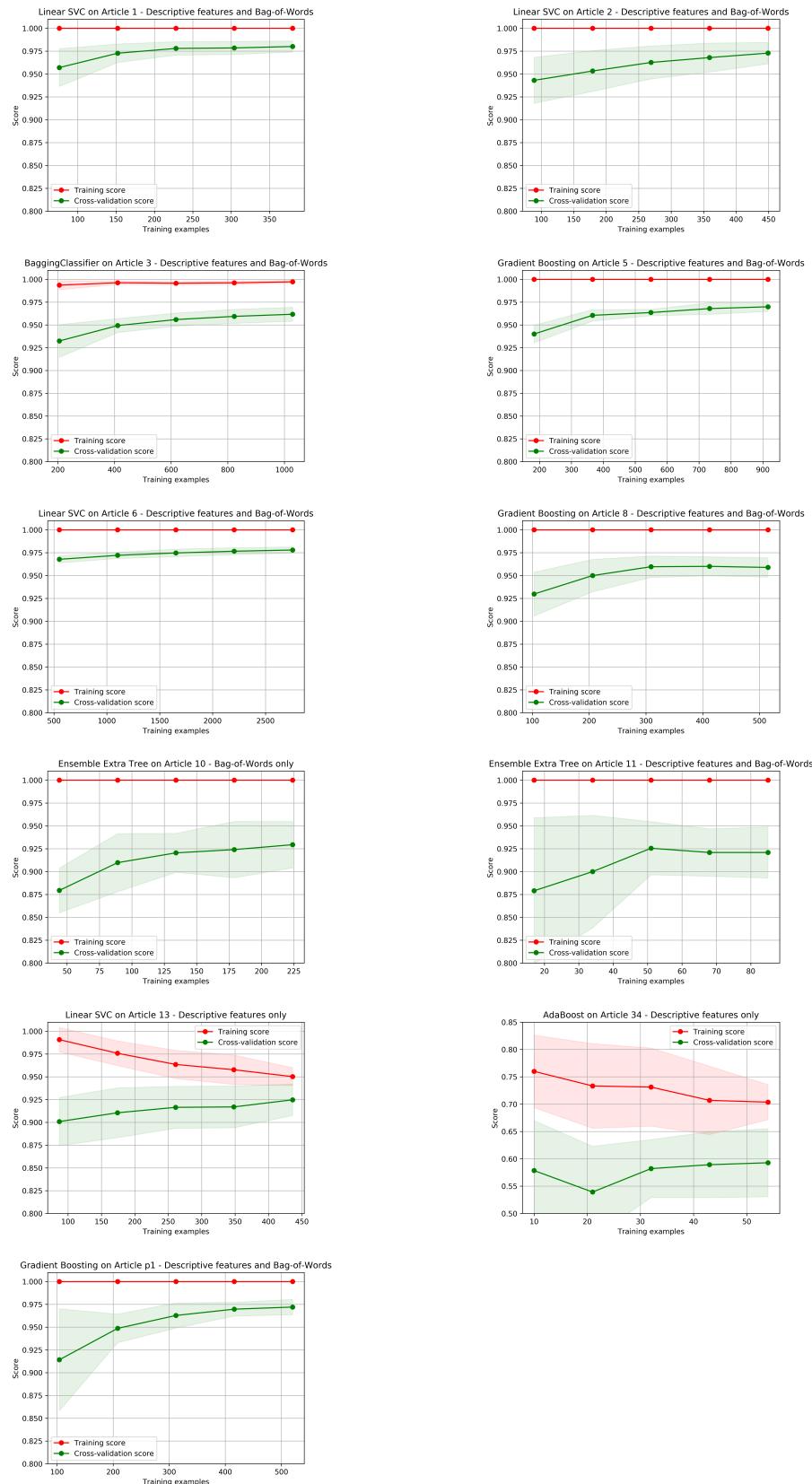
Fig. 7 Learning Curves for the best methods as described by Table 5.

Table 8 Accuracy obtained for each method on the multiclass dataset.

	Accuracy - Multiclass		
	desc	BoW	both
AdaBoost	0.7789 (0.04)	0.5451 (0.15)	0.5720 (0.04)
BaggingClassifier	0.8998 (0.01)	0.8794 (0.01)	0.9499 (0.01)
Bernoulli Naive Bayes	0.5096 (0.01)	0.7516 (0.01)	0.7464 (0.01)
Decision Tree	0.8897 (0.02)	0.8457 (0.01)	0.9434 (0.01)
Ensemble Extra Tree	0.8788 (0.01)	0.8904 (0.01)	0.9195 (0.01)
Extra Tree	0.7776 (0.03)	0.7164 (0.03)	0.7458 (0.02)
Gradient Boosting	0.8911 (0.01)	0.8904 (0.01)	0.9494 (0.01)
Linear SVC	0.9141 (0.01)	0.9136 (0.01)	0.9420 (0.01)
Multinomial Naive Bayes	0.7980 (0.01)	0.7784 (0.01)	0.7829 (0.01)
Neural Net	0.8813 (0.01)	0.9072 (0.01)	0.9231 (0.01)
Random Forest	0.8669 (0.01)	0.8825 (0.01)	0.9125 (0.01)

section, we showed that most methods could obtain an accuracy higher than the dataset prevalence, and more generally, good evaluation metrics. Usually algorithms for binary classification adapt relatively well to multiclass problems, however, in the case of ECHR-OD, the labels come by pair (violation or no-violation of a given article) which may confuse the classifiers. The experimental protocol being similar to the one of the previous section, we describe the results right after. For computational purposes, we dropped the two worst classifiers on the binary datasets, namely RBF SVM and KNN.

Table 8 presents the accuracy obtained on the multiclass dataset. The best accuracy for descriptive features only and Bag-of-Words only is linear SVM with respectively 91.41% and 91.36% correctly labeled cases. This is aligned with the results obtained on binary datasets. However, the top score of 94.99% is obtained by Bagging Classifier that only ranked 4th on binary datasets. In other words, SVM ranked first on two types of features individually, but the improvement of combining the features is lower than the one obtained by Bagging Classifier. The same can be observed with Gradient Boosting that outperforms SVM. Except from Ada Boost, the standard deviation is mostly lower than 1%.

For most methods, the flavor Bag-of-words only scores better than descriptive features. This observation is reversed by looking at the Matthew Correlation Coefficient provided by Table 9. For both indicators however, combining both types of features increases performances at the notable exceptions of Extra Tree, Multinomial Naive Bayes and Ada Boost with Decision Tree.

This highly contrasts with the binary setting for which descriptive features were quantitatively far below textual features, in particular the MCC indicator. For binary datasets, the flavor descriptive features only was mostly scoring below the bag-of-words only, for any article and any method (c.f. Supplementary Material). On top of that, taking only the best result per flavor, the descriptive features score better than purely textual features. The explanation can be found by studying the confusion matrix.

Fig. 8 Normalized Confusion Matrix for multiclass dataset. The normalization is performed per line. A white block indicates that no element has been predicted for the corresponding label. Percentages are reported only if above 1%.

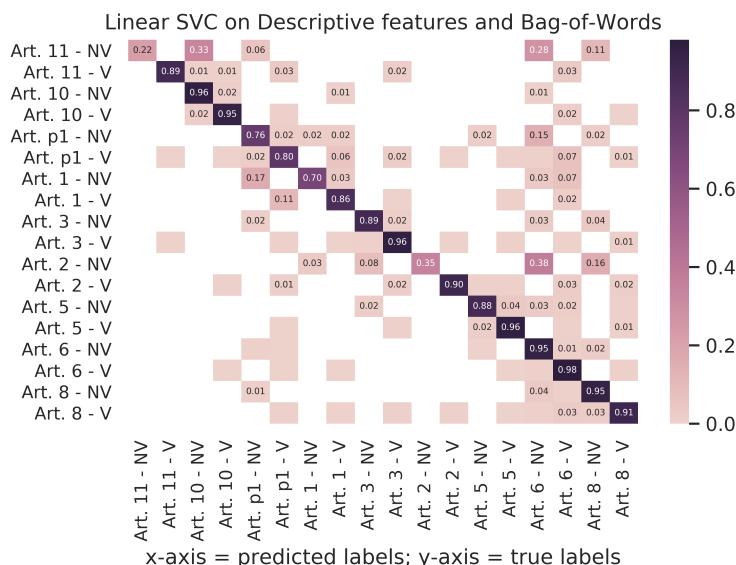
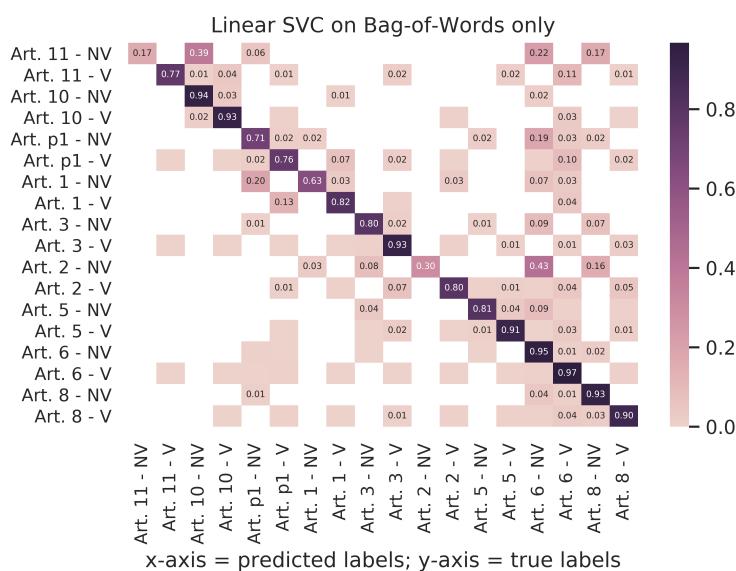
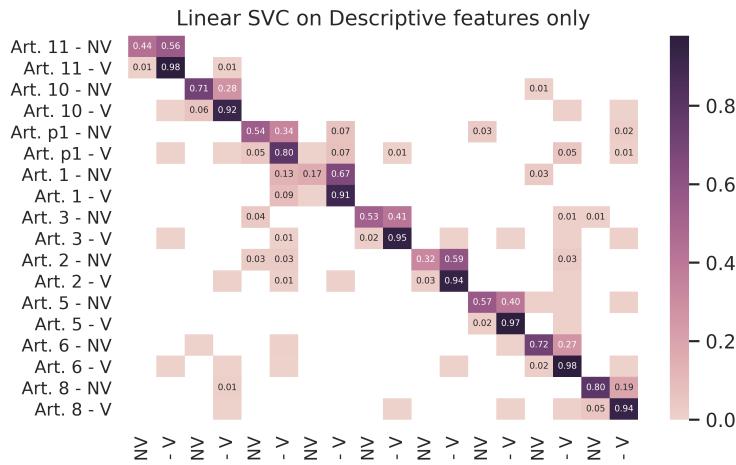


Table 9 Matthew Correlation Coefficient obtained for each method on the multiclass dataset.

	MCC - Multiclass		
	desc	BoW	both
AdaBoost	0.7171 (0.04)	0.4416 (0.14)	0.4580 (0.05)
BaggingClassifier	0.8700 (0.02)	0.8435 (0.02)	0.9353 (0.01)
Bernoulli Naive Bayes	0.2750 (0.02)	0.6845 (0.01)	0.6664 (0.01)
Decision Tree	0.8572 (0.02)	0.8004 (0.02)	0.9268 (0.01)
Ensemble Extra Tree	0.8419 (0.01)	0.8576 (0.01)	0.8956 (0.01)
Extra Tree	0.7103 (0.04)	0.6343 (0.04)	0.6700 (0.02)
Gradient Boosting	0.8580 (0.01)	0.8568 (0.01)	0.9346 (0.01)
Linear SVC	0.8886 (0.01)	0.8883 (0.01)	0.9251 (0.01)
Multinomial Naive Bayes	0.7323 (0.01)	0.7193 (0.01)	0.7190 (0.02)
Neural Net	0.8452 (0.01)	0.8795 (0.01)	0.9001 (0.01)
Random Forest	0.8262 (0.01)	0.8472 (0.01)	0.8864 (0.01)

Figure 8 shows the normalized confusion matrix for Linear SVM. The normalization has been done per line, i.e. each line represents the distribution of cases according to their ground truth. For instance, on descriptive features only, for the class "Article 11, no-violation", 44% only were correctly classified and 56% assigned to a violation of article 11. The perfect classifier should thus have a diagonal of 1. The diagonal is equivalent to the recall for the corresponding class and the average the diagonal terms is the balanced accuracy (Brodersen et al. (2010)).

The flavor "Descriptive features only" have a sparser normalized confusion matrix than the counterpart with Bag-of-Words. The fact that the first flavor returns a lower accuracy is explained by looking at the 2x2 blocks on the diagonal. Those blocks are the normalized confusion matrix of the subproblem restricted to find a specific article. For instance, 100% of non-violation of article 10 has been labeled in one of the two classes related to article 10 (99% for a violation). In general, the classifiers on "Descriptive feature only" are good at identifying the article but generates a lot of false negatives, most likely due to the imbalance between violation and non-violation labels for a given article. Adding the bag-of-words to the case representation slightly lowers the accuracy in average but largely rebalance the 2x2 blocks on the diagonal. On the other hand, it seems that the textual information does not hold enough information to identify the article, which explain why classifiers perform in general lower on this flavor.

We performed two Wilcoxon signed-rank tests: first between the samples of results on BoW and both, then between desc and both. The first result is clearly significant while the second is not significant, comforting us in the idea that for the multiclass domain, Bag-of-Words alone flavor is unlikely to give good results compared to descriptive features.

The main conclusion to draw from the multiclass experiment is that the descriptive features are excellent at identifying the article while the text offers more elements to predict the judgement. Using only the Bag-of-Words leads to the worst possible results, while adding textual information to the descriptive

features slightly increases the accuracy and have a strong beneficial effect on discriminating between violations and non-violations. This is quite in opposition with the conclusion drawn from the experiments on the binary datasets where the textual representation clearly overperformed while the descriptive features had only a marginal effect.

This indicates that it might be more interesting to create a two-stage classifier: a multiclass classifier determines the article based on descriptive features, followed by an article-specific classifier in charge of determining if the article is violated or not. Over-sampling techniques to deal with imbalanced classes constitutes another axis of improvement to explore in future work.

Last but not least, it shows that the benefits of combining sources of information are not monotonic: the best scoring method on individual types of features might not be the best method overall.

5.3 Multilabel Classification

The multilabel dataset generalizes the multiclass one in a way there is not only one article to identify before predicting the outcome, but an unknown number. It is closer to real-life situations in which, when a complaint is filled, the precise articles to be discussed are yet to be determined. On top of analyzing the usual performance metrics, we would like to quantify how good are the methods to identify all the articles in each case. From the multiclass results, it is expected that the textual information alone will provide the lowest results among all flavors.

5.3.1 Protocol

Appreciating the results of a multilabel classifier is not as easy as in the binary or multiclass case. For instance, having wrongly added one label to 100 cases is not exactly the same as adding 100 wrong labels to a single case. Similarly, being able to predict correctly at least one correct label per case is not the same as predicting all good labels for a fraction of the cases, even if the total amount of correct label is the same in both scenarios. The distributions of ground truth and predicted labels among the dataset are important to evaluate the model.

For this reason, we reported the following multilabel-specific metrics: subset accuracy, precision, recall, F_1 -score, Hamming loss and the Jaccard Similarity score. The subset accuracy is strictest metric since it measures the percentage of samples such that all the labels are correctly predicted. It does not account for partly correctly labeled vectors. The Jaccard index measures the number of correctly predicted labels divided by the union of prediction and true labels. The Hamming loss calculates the percentage of wrong labels in the total number of labels. Let T (resp. P) denotes the true (resp. predicted) labels, n the size of the sample, and l the number of possible labels. Then the metrics are defined by:

Table 10 Accuracy obtained for each method on the multilabel dataset.

	Multilabel		
	desc	BoW	both
Decision Tree	0.7837 (0.02)	0.6612 (0.02)	0.7966 (0.02)
Ensemble Extra Tree	0.7252 (0.03)	0.6643 (0.02)	0.6954 (0.02)
Extra Tree	0.5978 (0.03)	0.5410 (0.02)	0.5407 (0.03)
Neural Net	0.6914 (0.03)	0.6745 (0.02)	0.7159 (0.02)
Random Forest	0.7061 (0.03)	0.6438 (0.02)	0.6674 (0.02)

$$\text{ACC} = \frac{1}{n} \sum_{i=1}^n I(Y_i = \bar{Y}_i)$$

$$\text{RECALL} = \frac{T \cap P}{T}$$

$$\text{PRECISION} = \frac{T \cap P}{P}$$

$$F_1 = \frac{\text{RECALL} \times \text{PRECISION}}{\text{RECALL} + \text{PRECISION}}$$

$$\begin{aligned} \text{HAMMING} &= \frac{1}{nl} \sum_{i=1}^n \text{xor}(y_{i,j}, \bar{y}_{i,j}) \\ \text{JACCARD} &= \frac{T \cap P}{T \cup P} \end{aligned}$$

Finally, we are interested in quantifying how much a specific article was properly identified, as well as how much cases with a given number of labels are correctly labeled, taking into account their respective prevalence in the dataset reported by Figure 4. Indeed, about 70% of cases in the multilabel dataset have only one label such that a classifier assigning only one label to each case could reach about 70% of subset accuracy.

Not all binary classifiers can be extended for the multilabel problem. We used the five following algorithms: Extra Tree, Decision Tree, Random Forest, Ensemble Extra Tree and Neural Network. As previously, a 10-fold cross-validation has been performed on each flavor.

5.3.2 Results

The accuracy is reported in Table 10 and shows that Decision Tree outperforms with 79.66% of cases that have been totally correctly labeled. Similarly to the multiclass setting, the descriptive features provide a better result than the

Table 11 Precision obtained for each method on the multilabel dataset.

	Multilabel		
	desc	BoW	both
Decision Tree	0.8470	0.7780	0.8705
Ensemble Extra Tree	0.8808	0.8957	0.9147
Extra Tree	0.7202	0.6786	0.6780
Neural Net	0.8674	0.8719	0.8995
Random Forest	0.8698	0.8929	0.9120

Table 12 Recall obtained for each method on the multilabel dataset.

	Multilabel		
	desc	BoW	both
Decision Tree	0.8482	0.7688	0.8611
Ensemble Extra Tree	0.7635	0.7049	0.7261
Extra Tree	0.6999	0.6575	0.6564
Neural Net	0.7615	0.7671	0.7792
Random Forest	0.7440	0.6821	0.6996

Table 13 F_1 -score obtained for each method on the multilabel dataset.

	Multilabel		
	desc	BoW	both
Decision Tree	0.8446	0.7711	0.8639
Ensemble Extra Tree	0.7917	0.7720	0.7923
Extra Tree	0.7066	0.6651	0.6642
Neural Net	0.7938	0.7991	0.8174
Random Forest	0.7733	0.7533	0.7720

Table 14 Hamming loss obtained for each method on the multilabel dataset.

	Multilabel		
	desc	BoW	both
Decision Tree	0.0188	0.0286	0.0169
Ensemble Extra Tree	0.0195	0.0226	0.0203
Extra Tree	0.0350	0.0412	0.0412
Neural Net	0.0209	0.0210	0.0183
Random Forest	0.0208	0.0239	0.0219

bag-of-words. Decision Tree scores also the best for the F_1 -score (Table 13) and recall (Table 12). However, Ensemble Extra Tree overperforms Decision Tree when it comes to precision (Table 11). Decision Tree provides the best results for the *strict* metrics (highest accuracy and lowest Hamming loss) but also on more permissive metrics (best F_1 -score and Jaccard index). Therefore, all things being equal (in particular default hyperparameters), Decision Tree is clearly the top method for multilabel which is a bit surprising since it ranked 8th over the binary datasets and 3rd on the multiclass one.

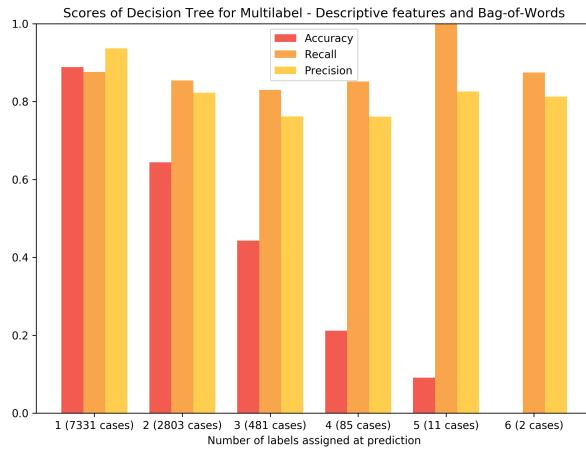
As expected, the bag-of-words flavor provides the worst possible results. Similarly to the experiments on the multiclass dataset, the textual information is inefficient at identifying the article.

The Figure 9 shows the accuracy, recall and precision depending on the number of labels assigned by Decision Tree on the test set. It also indicates

Table 15 Jaccard Similarity Score obtained for each method on the multilabel dataset.

	Multilabel		
	desc	BoW	both
Decision Tree	0.8424	0.7591	0.8672
Ensemble Extra Tree	0.7877	0.7376	0.7652
Extra Tree	0.6911	0.6431	0.6452
Neural Net	0.7673	0.7596	0.7888
Random Forest	0.7700	0.7160	0.7394

the number of cases for each label count. It is striking how the distribution of cases depending on the labels is close to the real distribution described by Figure 4. We can reasonably assume that the model correctly identifies the labels a case is about (or at least the article). The subset accuracy for cases with a single label is consistant with the score on multiclass dataset (which is then virtually similar). The subset accuracy decreases linearly with the number of labels, which is not surprising since the metric become stricter with the number of labels. However, the recall and precision remain stable, above 80% in average indicating that, not only the algorithm carefully identify the labels (recall) but also identify a large portion of labels (precision). Thus, Figure 9 clearly discards the possibility that the algorithm mostly focuses on cases with a single label.

**Fig. 9** Multilabel scores depending on the number of labels assigned.

6 Conclusion

In this paper, we introduced the European Court of Human Rights Open Data project consisting in multiple datasets for several variants of the classification

problems. The datasets come in different flavors (descriptive features, Bag-of-Word representation, TF-IDF) and are based on the real-life data directly retrieved from the HODOC database. In total, 13 datasets are provided for the first release. We argued that providing the final data are not enough to ensure quality and trust. In addition, there are always some opinionated choices in the representation, such as the number of tokens, the value of n for the n -grams calculation or the weighting schema in the TF-IDF transformation. As a remedy, we provide the whole process of dataset construction from scratch. The datasets will be iteratively corrected and updated along with the ECHR new judgments. The datasets are carefully versioned to reach a compromise between the need to keep the data up-to-date (as needed by legal practitioners or algorithms in production) and to have the same version of data to compare results between scientific studies.

In the future, we plan to add additional enrichments (e.g. entity extraction from the judgments), new datasets with fine-grain labels and new datasets for different problems (e.g. structured prediction). We hope to offer a web platform such that anyone can tune the different dataset hyperparameters to generate its own flavor: a sort of *Dataset as a Service*.

A first experimental campaign has been performed to established a baseline for future work. The predictability power of each dataset and flavor has been tested for the most popular machine learning methods. On binary datasets, we achieved an average accuracy of 0.9443, against 0.9499 for multiclass. It demonstrates the interest of treating the problem at a higher level rather than at the article level. In particular, the learning curves have shown that the models are underfitting on binary datasets but, as the datasets are exhaustive, it is not possible to provide more examples. We showed than the descriptive features are excellent at determining the articles related to a case while the textual features helps in determining the binary outcome. Combining both features always help, but the gap between the two type of features is smaller on the multiclass than on the binary datasets. In both cases, descriptive features actually hold reasonable predictive power. Those preliminary experiments certainly do not clearly answer the realism versus legalism debate, but they open new possibilities to understand better our justice system. They also provide several axes of improvements: hyperparameter tuning, multi-stage classifier and transfer learning. From those results, it seems clear that the prediction problem can be handled with the current state of the art in artificial intelligence. We hope that this project will pave the way toward a solution to the justification problem.

Last but not least, we encourage all researchers to explore the data, generate new datasets for various problems and submit their contributions to the project.

References

- Aletras N, Tsarapatsanis D, Preoiuc-Pietro D, Lampis V (2016) Predicting judicial decisions of the european court of human rights: a natural language processing perspective. *PeerJ Computer Science* 2:e93
- Attard J, Brennan R (2018) Challenges in value-driven data governance. In: "OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", Springer, pp 546–554
- Bengio Y, Courville A, Vincent P (2013) Representation learning: A review and new perspectives. *IEEE Trans Pattern Anal Mach Intel* 35(8):1798–1828
- Brodersen KH, Ong CS, Stephan KE, Buhmann JM (2010) The balanced accuracy and its posterior distribution. In: Proc. Int. Conf. Pattern Recognition, IEEE, pp 3121–3124
- Chicco D (2017) Ten quick tips for machine learning in computational biology. *BioData Min* 10(1)
- Crone SF, Lessmann S, Stahlbock R (2006) The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing. *Eur J Oper Res* 173(3):781 – 800
- Dasu T, Johnson T (2003) Exploratory data mining and data cleaning, vol 479. John Wiley & Sons
- Dyk DAV, Meng XL (2001) The art of data augmentation. *Journal of Computational and Graphical Statistics* 10(1):1–50
- Guimerà R, Sales-Pardo M (2011) Justice blocks and predictability of u.s. supreme court votes. *PLoS ONE* 6(11):e27188
- Gurstein MB (2011) Open data: Empowering the empowered or effective data use for everyone? *First Monday* 16(2)
- Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. *J Mach Learn Res* 3(Mar):1157–1182
- Islam MR, Hossain KSMT, Krishnan S, Ramakrishnan N (2016) Inferring multi-dimensional ideal points for us supreme court justices. In: Proc. Int. Conf. Art. Intel., pp 4–12
- Janssen M, Charalabidis Y, Zuiderwijk A (2012) Benefits, adoption barriers and myths of open data and open government. *Inform Syst Manage* 29(4):258–268
- Kassen M (2013) A promising phenomenon of open data: A case study of the chicago open data project. *Government Information Quarterly* 30(4):508–513
- Katz DM, Bommarito MJ, Blackman J (2017) A general approach for predicting the behavior of the supreme court of the united states. *PLOS ONE* 12(4):e0174698
- Kitchin R (2014) The data revolution: Big data, open data, data infrastructures and their consequences. Sage
- Klerman D (2012) The selection of 13th-century disputes for litigation. *J Emp Legal Studies* 9(2):320–346
- Lauderdale BE, Clark TS (2014) Scaling politically meaningful dimensions using texts and votes. *Am J Polit Sci* 58(3):754–771

- LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *nature* 521(7553):436
- Leiter B (2010) Legal formalism and legal realism: What is the issue? *Legal Theory* 16(2):111–133
- Martin AD, Quinn KM, Ruger TW, Kim PT (2004) Competing approaches to predicting supreme court decision making. *Perspectives on Politics* 2(4):761767
- Misuraca G, Viscusi G (2014) Is open data enough?: E-governance challenges for open government. *Int J Elect Gov Res* 10(1):18–34
- Molloy JC (2011) The open knowledge foundation: Open data means better science. *PLoS Biology* 9(12):1–4
- Nalepa J, Myller M, Piechaczek S, Hrynczenko K, Kawulok M (2018) Genetic selection of training sets for (not only) artificial neural networks. In: Proc. Int. Conf. Beyond Databases, Architectures Struct., pp 194–206
- Nawi NM, Atomi WH, Rehman MZ (2013) The effect of data pre-processing on optimized training of artificial neural networks. *Procedia Technology* 11:32 – 39, int. Conf. Elect. Eng. Info.
- Olhede SC, Wolfe PJ (2018) The future of statistics and data science. "Stat & Probab Letters" 136:46 – 50
- Patel TA, Puppala M, Onguni RO, Ensor JE, He T, Shewale JB, Ankerst DP, Kaklamani VG, Rodriguez AA, Wong STC, Chang JC (2016) Correlating mammographic and pathologic findings in clinical decision support using natural language processing and data mining methods. *Cancer* 123(1):114–121
- Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. *J Mach Learn Res* 12:2825–2830
- Posner R (2010) How Judges Think. Harvard University Press
- Quemy A (2017) Data science techniques for law and justice: Current state of research and open problems. In: Proc. Int. Workshop Data Sci.: Methodologies and Use-Cases, pp 302–312
- Quemy A (2019) Data pipeline selection and optimization. In: Proc. Int. Workshop Des. Optim. Lang. Anal. Process. Big Data
- Quinn KM, Martin AD (2002) Dynamic Ideal Point Estimation via Markov Chain Monte Carlo for the U.S. Supreme Court, 1953–1999. *Polit Anal* 10(2):134–153
- Řehůřek R, Sojka P (2010) Software Framework for Topic Modelling with Large Corpora. In: Proc. of the LREC Workshop, ELRA, Valletta, Malta, pp 45–50
- Rissland EL (2006) Ai and similarity. *IEEE Intel Syst* 21(3):39–49
- Ruger TW, Kim PT, Martin AD, Quinn KM (2004) The supreme court forecasting project: Legal and political science approaches to predicting supreme court decisionmaking. *Columbia Law Review* 104(4):11501210
- Segal JA, Cover AD (1989) Ideological values and the votes of u.s. supreme court justices. *Am Pol Sci Review* 83(2):557–565

- Segal JA, Epstein L, Cameron CM, Spaeth HJ (1995) Ideological values and the votes of u.s. supreme court justices revisited. *J Politics* 57(3):812–823
- Sim Y, Routledge BR, Smith NA (2014) The utility of text: The case of amicus briefs and the supreme court. *CoRR* abs/1409.7985
- Spitzer M, Cohen L (1994) Solving the chevron puzzle. *J Law & Contemporary Problems* 57:65
- Tallon PP (2013) Corporate governance of big data: Perspectives on value, risk, and cost. *Computer* 46(6):32–38
- Tamanaha BZ (2012) Balanced realism on judging
- Tiwari P, Prasanna P, Wolansky L, Pinho M, Cohen M, Nayate A, Gupta A, Singh G, Hatanpaa K, Sloan A, Rogers L, Madabhushi A (2016) Computer-extracted texture features to distinguish cerebral radionecrosis from recurrent brain tumors on multiparametric MRI: A feasibility study. *Am J Neurorad* 37(12):2231–2236
- Valiant LG (1984) A theory of the learnable. *Commun ACM* 27(11):1134–1142
- Vapnik VN (1999) An overview of statistical learning theory. *IEEE Trans Neural Net* 10(5):988–999
- Yu KH, Zhang C, Berry GJ, Altman RB, Ré C, Rubin DL, Snyder M (2016) Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features. *Nature Commun* 7:12474

Supplementary Material

https://aqueemy.github.io/ECHR-OD_project_supplementary_material/