Artificial Intelligence and Law (2024) 32:201–230
https://doi.org/10.1007/s10506-023-09348-9

ORIGINAL RESEARCH

Joining metadata and textual features to advise administrative courts decisions: a cascading classifier approach

Hugo Mentzingen[1] · Nuno Antonio[1] · Victor Lobo[1,2]

Accepted: 27 January 2023 / Published online: 18 February 2023

Abstract

Decisions of regulatory government bodies and courts affect many aspects of citizens' lives. These organizations and courts are expected to provide timely and coherent decisions, although they struggle to keep up with the increasing demand. The ability of machine learning (ML) models to predict such decisions based on past cases under similar circumstances was assessed in some recent works. The dominant conclusion is that the prediction goal is achievable with high accuracy. Nevertheless, most of those works do not consider important aspects for ML models that can impact performance and affect real-world usefulness, such as consistency, out-of-sample applicability, generality, and explainability preservation. To our knowledge, none considered all those aspects, and no previous study addressed the joint use of metadata and text-extracted variables to predict administrative decisions. We propose a predictive model that addresses the abovementioned concerns based on a two-stage cascade classifier. The model employs a first-stage prediction based on textual features extracted from the original documents and a second-stage classifier that includes proceedings' metadata. The study was conducted using time-based cross-validation, built on data available before the predicted judgment. It provides predictions as soon as the decision date is scheduled and only considers the first document in each proceeding, along with the metadata recorded when the infringement is first registered. Finally, the proposed model provides local explainability by preserving visibility on the textual features and employing the SHapley Additive exPlanations (SHAP). Our findings suggest that this cascade approach surpasses the standalone stages and achieves relatively high Precision and Recall when both text and metadata are available while preserving real-world usefulness. With a weighted F1 score of 0.900, the results outperform the text-only baseline by 1.24% and the metadata-only baseline by 5.63%, with better discriminative properties evaluated by the receiver operating characteristic and precision-recall curves.

Keywords Administrative decision prediction · Cascade generalization · Legal assistance · Machine learning · Natural language processing

Extended author information available on the last page of the article

1 Introduction

Public services such as licensing, social benefits granting, or economic agents supervision are carried out by government bodies whose decisions affect various aspects of citizens' lives. Often, these institutions are responsible for solving conflicts, enforcing rules, and shaping administrative justice. This justice system deals, in many countries, with more cases than criminal or private civil justice (Nason 2018).

Within this scope, the corrective measures taken by financial system supervisors, namely in the banking and insurance sectors, are a good example of enforcing administrative law. The worldwide insurance industry's gross premiums in 2018 reached USD 5.3 trillion, with an average penetration[1] of 7.23% in 2019 (Statista 2020), supporting the need for regulation and supervision. The supervisory authorities' duties include bankruptcy avoidance, contract fairness evaluation, and consumer protection.

One of the insurance supervision principles asserts that the jurisdictions must guarantee that the supervisor "enforces corrective action and, where needed, imposes sanctions based on clear and objective criteria that are publicly disclosed" (IAIS 2017). This principle directly gives rise to a demand for administrative bodies that timely deliver justified and coherent proceedings when infringements are identified.

However, to produce coherent and timely decisions, the decision-maker would benefit from swiftly knowing the past decisions in similar cases. In addition, inspectors responsible for supervising companies gain from knowing in advance whether potential problems are usually considered infringements by administrative courts. Machine learning (ML) may provide valuable tools to help administrative courts meet this demand. This study proposes a novel ML method for combining variables extracted from the infringement text and proceedings metadata to predict proceedings outcomes. Its three main goals are to ensure good generalization, so that other courts can adopt it, out-of-sample applicability (Katz et al. 2017), and preserve some degree of explainability.

We used data from the Superintendency of Private Insurance (SUSEP), an independent agency under Brazil's Ministry of Finance with authority to license and supervise insurance brokers and companies. In 2019, SUSEP oversaw 286 insurers and reinsurers, with technical provisions of nearly one trillion BRL (two hundred fifty billion USD) (SUSEP 2020a) and 99.836 insurance brokers (SUSEP 2020b). The agency's duties include monitoring and prosecuting infringements to the rules in force, both for prudential and conduct infractions. As a conduct directive example, insurers are required to pay claims within 30 days, a deadline that only may be suspended if the claim processing requires additional information. SUSEP may start a sanctioning proceeding after an inspection or a complaint and enforce penalties if an infringement has existed. Similarly, as an example of a prudential rule,

[1] The ratio of total insurance premiums to gross domestic product.

companies must regularly provide financial information and maintain investments under specific guidelines, whose infringement may result in penalties.

SUSEP initiated, on average, 822 infringement proceedings per year between 2016 and 2019. When data were extracted and considering cases initiated in this period, 2471 cases remained undecided, as well as, on average, it took 1113 days to decide. In this context, there is an immense opportunity to take advantage of ML on supervision efficiency. Decidedly SUSEP and other governmental entities that assume administrative judges' roles can use this study's outcome to speed up the legal analysis in their jurisdictions while still observing jurisprudence.

This study is dedicated to predicting the infringement proceedings' decisions, *i.e.*, whether an infringement has existed or not, which is a binary classification task. It provides an advisory tool so inspectors and decision-makers can take past decisions under similar circumstances when dealing with a new case. The proposed model can provide uniformity, legal certainty, and better use of jurisprudence without neglecting the need for human reasoning in each specific case.

In practice, the model provides the government stakeholders (inspectors, legal analysts, and judges) with a presumable outcome for each case based on past decisions under similar circumstances. Additionally, other stakeholders such as lawyers, companies, citizens, and administrative staff of the courts can benefit from knowing the trend of a judgment of infringements under similar circumstances, saving time and resources.

2 Literature review

Recent work established practices for predicting judicial decisions, either strictly using metadata or text-based approaches. In a seminal paper, Ruger et al. (2004) proposed a statistical method (in practice, classification trees) for predicting the Supreme Court of the United States (SCOTUS) decisions, using general case characteristics, and compared its results with legal experts' predictions. The result was a much better performance achieved by the classification trees. The authors explain it was due to the model's ability to predict the more ideologically central votes, suggesting that it was more accurate than the gold-standard legal analysts in separating the classes in a binary decision task. Despite choosing variables particular to that court and not generalizing for SCOTUS terms different from those used in 2002, this work brought an optimistic scenario in which metadata-based decision prediction could outperform human-based judgment.

In another experiment with classification trees, Chen and Eagel (2017) applied the Random Forests algorithm (an ensemble of randomized decision trees) to a much larger dataset related to asylum adjudications. The authors used metadata internal to the cases and extraneous factors such as news and local weather to predict outcomes accurately. On the other hand, the number of variables involved hampered the model's interpretation. Also, incorporating external variables demanded analysis of whether they represented causality or mere correlation. The proposal had a complex replication from this perspective, mainly because it considered the subject's many internal and external variables.

Random Forests' success for metadata-based prediction tasks was confirmed by Katz et al. (2017) in a paper that described the design of a new model for predicting SCOTUS decisions. The authors established consistency, out-of-sample applicability, and generality as principles to make predictive ML models useful for real-world applications.

Consistency may be described as the quality of having the lowest possible performance variance across time, case issues, and Justices. Out-of-sample applicability, in its turn, is directly related to ensuring that all information required for the model is known before the decision's date. Katz et al. (2017) also described generality as a model's property of achieving good performance on different court compositions. The authors compared their assembly with the study from Ruger et al. (2004) that, even though representing an enormous contribution to the field, did not apply to other terms or different compositions of the SCOTUS. In the authors' description, a general model is a model that can learn across time with new samples. Finally, to ensure out-of-sample applicability, Katz et al. (2017) divided the dataset into yearly terms and used data before each test period for training.

To the authors' knowledge, Aletras et al. (2016) produced the first study based solely on textual content for predicting the outcome of cases tried by the European Court of Human Rights (ECHR). With a bag-of-words embedding model, the authors extracted *n*-gram features and topics based on *n*-gram similarity. Their study provided strong evidence that the factual background described in the textual data may be a good predictor for each trial's result, *i.e.*, the usefulness of the natural language processing (NLP) approach.

Medvedeva et al. (2020), in their turn, extended these findings by testing how well support vector machines (SVM) were able to predict future cases by dividing the samples used for training and testing based on the year of the case. They concluded that forecasting decisions for future lawsuits, *i.e.*, considering the time dimension, poses a much more challenging task, and predictions based on the distant past negatively impact performance.

Following the evolution of neural network-based classifiers, Pillai and Chandran (2020) applied convolutional neural networks (CNN) to bag-of-words derived from Indian Courts data. Sivaranjani et al. (2021) applied hierarchical convolutional neural networks (HCNN) on a feature set extracted from the Indian Supreme Court to predict the outcomes of appeal cases. The feasibility and results of these deep-learning studies for text-based prediction are questionable as they lack generality and out-of-sample applicability for not considering the time dimension when testing the findings. As in some previous works, the mentioned studies also present the results based on the implementations' Accuracy, which can skew an imbalanced datasets' results. Accuracy mixes true positives and true negatives, being prevalence-dependent, which enforces the importance of choosing an adequate score-evaluation metric.

Moreover, the increasingly frequent application of deep learning techniques and black-box models in the legal context motivated the study of Bibal et al. (2021). In the authors' words, "despite their high performance, they may not be accepted ethically or legally because of their lack of explainability." This review paper clarified the concept of explainability in law and how the different levels of legal requirements could be interpreted and translated into ML models' explainability. In their

Table 1 Roles and their responsibilities in the infringement examination process

Role	Responsibilities
Inspector	Assess potential violations during their routine activities and file an infraction notice as appropriate, check whether a complaint corresponds to a possible infringement, and register the infraction notice or complaint in the penalties system
Defendant	Present defense
Legal analyst	Conduct a formal review of cases, notify Defendant to present a defense, prepare the case for trial
Administrative Judge	Issuing the final decision in cases whose infraction is within their competence
Board of directors	Issuing the final decision in cases whose infraction is within its competence

study, sets of requirements that can be required from ML models were derived from the need for decision motivation.

To the authors' knowledge, no previous works investigated the joint use of metadata and textual information through a cascading classifier model for legal decision prediction.

3 Methodology

In the current infringement examination process, represented in Appendix 1, the proceedings are initiated by inspectors and distributed among legal analysts. The latter produces an evaluation of the case, subsiding the decision-making. Their role is crucial to the trial process since legal analysts are the ones who conduct the formal review and prepare the case for trial, acting as assistants to the judge. The normative competence for judging each case depends on the infringed rule: it can be from an administrative judge or the agency's board of directors, composed of five members. The responsibilities in the infringement examination process are summarized in Table 1.

The number of participants in this business process makes it even more challenging to consider the existing jurisprudence. Furthermore, each analyst can evaluate similar cases differently over time, mainly depending on the final-decision makers to maintain coherence. A tool based on this study can help stakeholders obtain visibility on the precedents of court judgments. In practice, inspectors, analysts, and judges may receive advice on the decision outcome (a binary classification) by processing the proceeding's initial document.

It is also common to find different administrative judges and distinct compositions of the directors' board, *e.g.*, in the analyzed data, there were two different judges and two different board compositions. Keeping the model's performance across time and under varying circumstances creates a challenge.

On the other hand, the differences between judicial and administrative proceedings are advantageous for text feature extraction. In this sort of administrative justice, although complaints may initiate the proceedings, most of the potential infringements are identified by inspectors during their routine activities, leading to infraction notices of similar wording. They commonly refer exclusively to infringed legislation and do not contain a significant number of references to decisions taken in similar cases. In cases where the proceeding begins with a complaint, the inspectors assess whether there is any potential infringement. Still, an infraction notice is not generated, and the complaint continues in its original state.

3.1 Data extraction

Data from two of SUSEP's internal systems were used: the penalties system and the administrative process system. The former keeps track of the infringement analysis lifecycle, storing metadata as the infringement date, regulations involved, whether the infringement was committed by a person or a company, decision date, and the existence of mitigating or aggravating circumstances. The later system stores the text documents of the proceedings in a timely order.

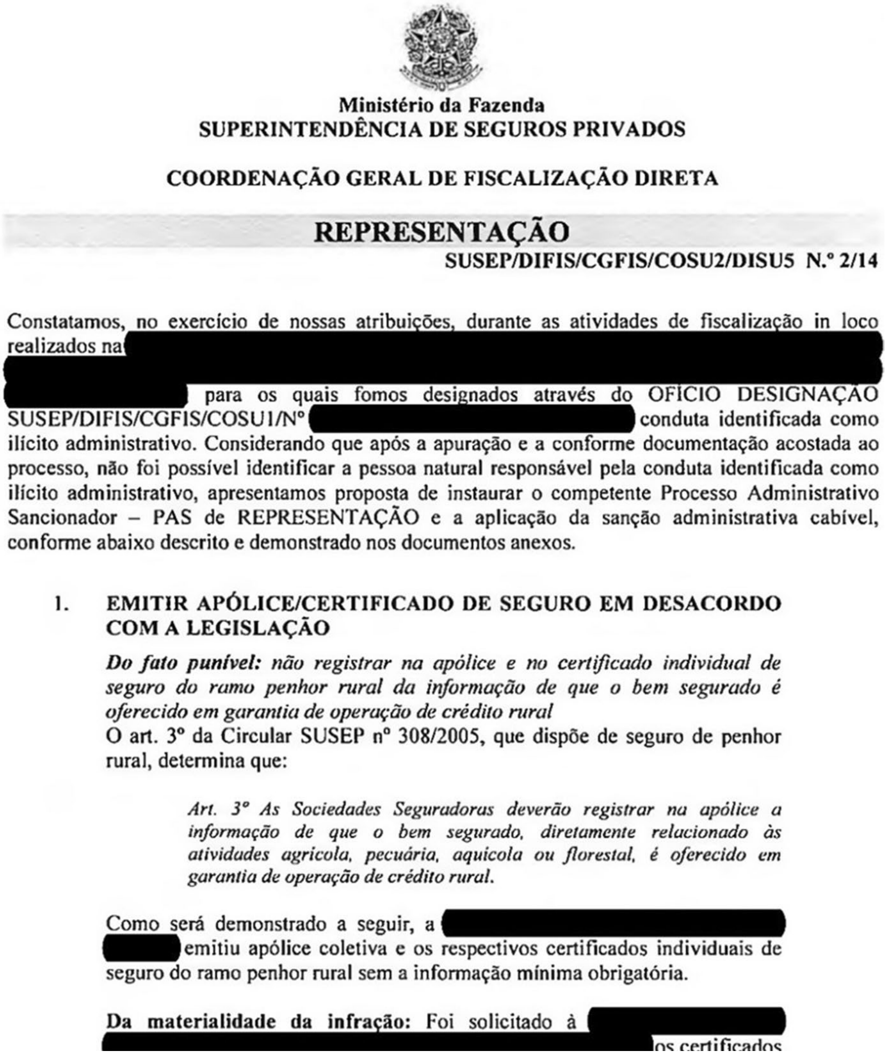
Three criteria were considered to create the dataset. First, the potential administrative infringement must have had a decision by an administrative judge, which means having a decision registered in the penalties system. Second, the administrative proceeding must have been born-digital so that data can be adequately parsed (documents in HTML or PDF format), and third, the alleged infringement must still be among the rules in force.

Each proceeding may have had one or more potential infringements, regularly described in a single complaint or infraction notice, from now on referred to as the primary document. This fact created the need to split the text related to each infraction. The infringement was used as the unit of analysis, and the respective part of the primary document was extracted for each of them. Some proceedings also contained drafts and rectifications, meaning, in these cases, that there were two or more copies of the same procedural documents. As a general rule, the last versions were believed to be final.

The computer program BeautifulSoup (Richardson 2007) extracted text from the HTML files and pdfminer.six (Shinyama et al. 2019) served the same purpose for PDF files. Different parsers were built for complaints and infraction notices, in which regular expressions helped remove prologues and epilogues and extract the core text for each infringement. Figure 1 shows an example of an infraction notice stored as a PDF file, and Appendix 2 contains a free English translation of the text.

3.2 Dataset characteristics

The result was a dataset containing 1108 infringements, including their primary document's text extract. The following metadata was of interest for this study: infringed regulation identifier (*infringementID*), whether the offender was a person or a company (*pessoaFisica*), infringement date (*infringementDate*), decision date (*decisionDate*), and decision type (*julgamentoDecisaoId*). Excepting the features related to the decision, all others are available at the beginning of an infringement analysis



Ministério da Fazenda
SUPERINTENDÊNCIA DE SEGUROS PRIVADOS

COORDENAÇÃO GERAL DE FISCALIZAÇÃO DIRETA

REPRESENTAÇÃO
SUSEP/DIFIS/CGFIS/COSU2/DISU5 N.° 2/14

Constatamos, no exercício de nossas atribuições, durante as atividades de fiscalização in loco realizados na [redacted] para os quais fomos designados através do OFÍCIO DESIGNAÇÃO SUSEP/DIFIS/CGFIS/COSU1/N° [redacted] conduta identificada como ilícito administrativo. Considerando que após a apuração e a conforme documentação acostada ao processo, não foi possível identificar a pessoa natural responsável pela conduta identificada como ilícito administrativo, apresentamos proposta de instaurar o competente Processo Administrativo Sancionador – PAS de REPRESENTAÇÃO e a aplicação da sanção administrativa cabível, conforme abaixo descrito e demonstrado nos documentos anexos.

1. **EMITIR APÓLICE/CERTIFICADO DE SEGURO EM DESACORDO COM A LEGISLAÇÃO**

Do fato punível: *não registrar na apólice e no certificado individual de seguro do ramo penhor rural da informação de que o bem segurado é oferecido em garantia de operação de crédito rural*

O art. 3° da Circular SUSEP n° 308/2005, que dispõe de seguro de penhor rural, determina que:

> *Art. 3° As Sociedades Seguradoras deverão registrar na apólice a informação de que o bem segurado, diretamente relacionado às atividades agricola, pecuária, aquicola ou florestal, é oferecido em garantia de operação de crédito rural.*

Como será demonstrado a seguir, a [redacted] emitiu apólice coletiva e os respectivos certificados individuais de seguro do ramo penhor rural sem a informação mínima obrigatória.

Da materialidade da infração: Foi solicitado à [redacted] os certificados

Fig. 1 PDF Infraction notice sample. This document describes the findings of an inspection team and may contain one or more potential infringements

providing an early prediction of its outcome. Hence, the model can be reproduced with the same variables set as soon as the decision date is fixed or estimated.

A subset of 109 samples (9.8%) did not have a judgment date. Instead of discarding such a noticeable percentage of samples, the decision date in these cases was set to be the average date among all previous decisions. Decision dates ranged from June 2016 to August 2020.

The possible outcomes identified by *julgamentoDecisaoId* were five. One appeared in 830 samples (74.9%) and represented the positive case, *i.e.*, a situation

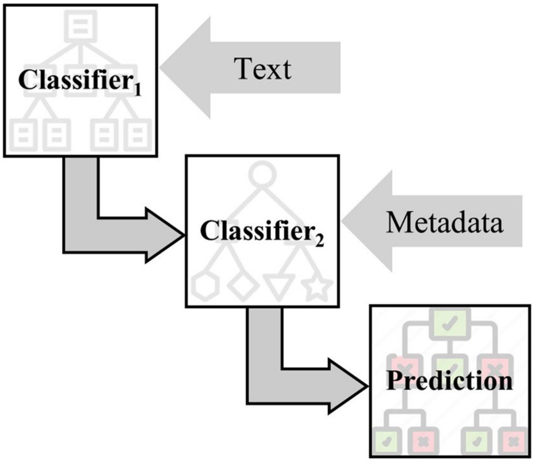


Fig. 2 Cascading classifiers to combine text and metadata

in which the decision-maker considered an infringement had occurred. The remaining four included the following circumstances: proceedings filed without judgment on the merits (1.6%), those in which the judge(s) decided that an infringement had not occurred (16.6%), proceedings that generated a recommendation for the company, and not a penalty (0.5%), or even those extinct by any legal reason (6.3%). Jointly, these four outcomes represented the negative case.

Finally, in 14.0% of the samples, the proceedings' origin was a customer complaint, and the 86.0% remaining originated from infraction notices.

3.3 Study hypothesis

As illustrated in Fig. 2, this study employs a two-layer cascade classification model to predict whether an administrative judge will confirm a potential infringement. The hypothesis under evaluation is that the proposed model obtains a better combination of Precision and Recall than two baselines: a metadata-only classifier and a text-only classifier. The proposed model uses features extracted from the text (*n*-grams) and features extracted from the metadata.

From a practical perspective, the objective is to evaluate one outcome against the others, turning the task into a binary classification. We implement an architecture based on the Cascade Generalization (Gama and Brazdil 2000), feeding the predictor that uses metadata with the prediction obtained from the textual features.

This multistage method is a better choice than other ensemble techniques, like bagging or boosting, as we want later to examine features' contribution to the final classification. Also, the computational cost of training the metadata-based and text-based learners can be divided, contrarily to using all features in a single classifier. Finally, selecting the best text-based classifier first, which is more costly, allows testing parameters for the second classifier in less time.

Based on Katz et al. (2017) contributions, this study satisfies the concepts of generality, consistency, and out-of-sample applicability, proposing a practical approach for various administrations, provided they store decisions' text and metadata. The generality property is addressed by evaluating the model across different decision-makers, a single judge, or the board of directors, with varying compositions across time. Out-of-sample applicability is intrinsic to the validation strategy since all training samples precede the test samples in time, guaranteeing that all information the model needs to produce an estimate is known before the predicted decision's date.

Consistency across time is directly addressed by time-based cross-validation. Furthermore, as mentioned in Sect. 3, the data span a period with two different administrative judges, two board compositions, and several different faults. Thus, this validation method also indirectly delivers consistency across case issues and courts.

Text-based features were used to train the first classifier. The decision to use text representations based on *n*-grams and embed the text into term frequency-inverse document frequency (TF-IDF) (Luhn 1957; Spärck Jones 1972) vectors or Latent Dirichlet Allocation (LDA) (Blei et al. 2003) topics relate to preserving visibility over the textual variables that contribute to the first stage's prediction. Although neural network-based models may capture semantic features (*e.g.*, Doc2Vec (Le and Mikolov 2014), Word2Vec (Mikolov et al. 2013), GloVe (Pennington et al. 2014)) and transformer-based models (*e.g.*, Universal Sentence Encoder (Cer et al. 2018), BERT (Devlin et al. 2019)), they are not interpretable by their nature. Moreover, the cascading strategy for combining heterogeneous features through two classifiers enables one to evaluate the classification's contribution from the first stage to the second stage.

The second classifier was fed with the predicted class from the early stage as an additional feature, along with infringement metadata. It should be noted that ensemble techniques such as bagging and boosting are not feasible to connect the first and second stages in this assembly because they apply bootstrapping as the sampling method from a single data set. Similarly, the stacking technique could only be employed in case the second stage was a meta-classifier.

Following the need for explainability when predicting administrative decisions through an ML model, we chose word embeddings and an ensemble technique that provides human-interpretable features. The *n*-grams relevant to the prediction relate to the facts behind a decision, and the legal article is a feature present in the metadata. In this sense, we used the SHapley Additive exPlanations (SHAP) detailed by Lundberg and Lee (2017) to approximate each feature's contribution and reverse-engineer the output. This model agnostic approach provides local explainability, *i.e.*, a specific prediction made by the model can be understood by approximating the decision through an interpretable local model. This local model does not globally explain the whole model but instead provides clues on why a specific prediction has been made (Bibal et al. 2021).

Preserving human-recognizable features to interpret their contributions allows understanding the decision prediction standalone stages. In this sense, not only does the proposed model address how the variables combine to reach a decision, but it also allows identifying the text extracts (facts) that contribute to the decision (first stage) and the legal article(s) relevant to the case (second stage).

3.4 Feature engineering

Three features were also engineered from data to address relevant aspects of the decisions and to feed the second stage classifier. First, the time difference between the decision date and the infringement date, *i.e.*, the time the administrative court took to deal with the case and decide on it, was stored in the feature *tempoAteJulgar*. Next, the *textLength* feature was created to store the primary document length for each infringement. Finally, the time difference between the infringement date and January 1st, 2000, was stored in the *daysFrom2000* feature and substituted *infringementDate*, addressing the case recency and treating the date as an ordinal and continuous variable. In summary, the following features were made available to the second stage classifier:

- *infringementID:* infringed regulation identifier;
- *pessoaFisica*: whether the offender is a person or a company;
- *decisionDate*: decision date;
- *tempoAteJulgar*: time difference between the decision date and the infringement date;
- *textLength*: primary document length;
- *daysFrom2000*: time difference between the infringement date and January 1st, 2000;
- *subsistent*: the binary target feature, corresponding to the decision result.

The *infringementID* feature had a high cardinality when considering the number of samples, with 97 different values. As a data preparation measure, *infringementID* classes were sorted by the number of occurrences, and the values representing 90% of the cases were kept, resulting in 40 categories. The remaining possible *infringementID* were grouped into a default category. Finally, this feature was encoded into dummy binary variables, making it suitable for the classification algorithms. *InfringementDate* was transformed into a continuous variable ranging between the dataset's minimum and maximum date.

3.5 Baselines

Our study aimed to compare the experimental results with both a classifier based only on metadata and a classifier based on features extracted from the text. Both on the baselines and the cascade classifier stages, we employed seven estimators based on different learning methods: a lazy (or instance-based) learner represented by *k*-Nearest Neighbors (*k*NN), a neural network (or Multilayer Perceptron), a tree-based classifier (Decision Tree), a Bagging Classifier using decision trees as its base, a bagging classifier employing random subsampling of features (Random Forest), a boosting classifier (XGBoost) and an analogy-based learner (Support Vector Classifier). All estimators used versions implemented on Scikit-Learn (Pedregosa et al. 2011), except XGBoost implemented in Python by T. Chen and Guestrin (2016).

3.6 First stage classifier

Text preprocessing was the first step of the pipeline, removing ubiquitous and non-discriminative words from the primary documents using regular expressions (RegEx), *e.g.*, phone numbers, URLs, and business unit names. Part-of-speech (POS) tagging was also used to remove articles, conjunctions, pronouns, and currency symbols. Several POS taggers were tested. The Mac-Morpho Brazilian Portuguese corpus (Fonseca et al. 2015) associated with Brill Tagger (Brill 1992) from the Natural Language Toolkit—NLTK (Bird et al. 2009) obtained the best performance in classifying sentences. NLTK was also used to apply, in sequence, tokenization, stopword removal, and stemming. Tokenization is the text segmentation into basic units (tokens) such as words and punctuation, while Stemming accounts for reducing words to their root form based on pre-established rules. For example, a rule may state that any expression with *-ing* as a suffix will be reduced by removing the suffix (Bird et al. 2009). The last step used the RSLP Stemmer for Portuguese (Orengo and Huyck 2001).

We also used POS tagging to extract words and summarize the text according to three summarization strategies: the first returned the full stemmed text, and the second and third summaries were composed of the nouns (concepts) and nouns and verbs (concepts and relations), respectively. In the latter cases, the resulting text was also stemmed.

A bag-of-words model was used to represent the text corresponding to each infringement on the vectorization step. The bag-of-words approach assumes that a text is nothing more than a histogram of the words it contains. Thus, words' order or context are not considered (Theodoridis 2020). For vectorial document representation, two approaches were tested. First, words' occurrence frequency was normalized by the number of infringements in which each word occurs, *i.e.*, through TF-IDF. Also, in this case, words appearing in only one document or more than 80% of the corpus were ignored. The second approach involved counting the number of occurrences of each word in the documents and applying the Latent Dirichlet Allocation (LDA) model, representing each document by its composition in LDA topics.

The TF-IDF representation proved effective in different scenarios, generally with up to four words in each *n*-gram. In our case, *n*-gram sizes varied from 1 to 4. Each vectorization trial utilized pairs of the following *n*-gram sizes: (1,2), (2,3), and (3,4). The vocabulary size, *i.e.*, the number of different *n*-grams obtained when mapping terms to features, reached 185.921 with *n*-grams of size (1,2) and the full stemmed text.

Due to the large number of features, both for performance reasons and to avoid the *curse of dimensionality*, a dimensionality reduction technique was applied. This phenomenon is observed when estimating an arbitrary function by a high number of features minimizing the approximation error. L. Chen (2009) states that the number of samples needed for this estimation with a given level of Accuracy grows exponentially with the number of input variables (*i.e.*, the dimensionality) of the function. The computational effort also increases exponentially with the number of unknown variables. In such cases, dimensionality reduction techniques transform

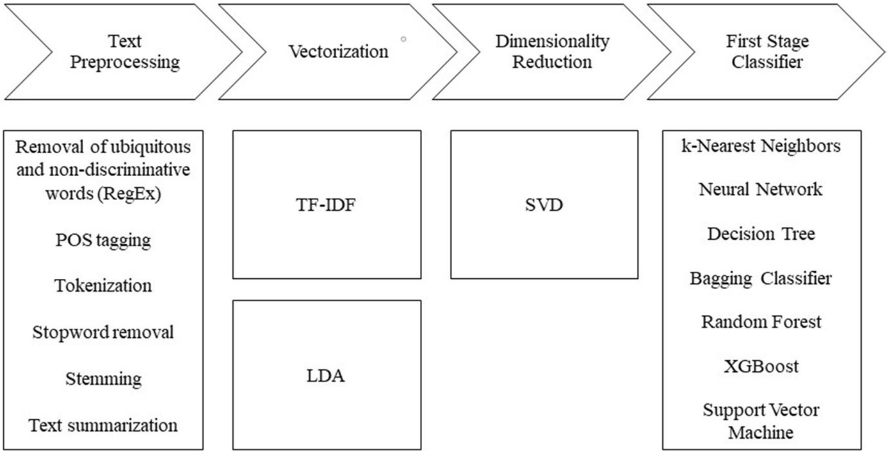


Fig. 3 Pipeline for the first stage classifier

the data from a high-dimensional space into a low-dimensional space, retaining the original representation's properties as much as possible.

Hence, Singular Value Decomposition (SVD) was applied to the vectorized representation for each n-gram size using a components number equal to 1% of the feature space. It means approximating the feature-space matrix $\boldsymbol{A}$ with another lower-rank matrix $\boldsymbol{B}$, *i.e.*, truncating $\boldsymbol{A}$ to a specific rank r corresponding to 1% of this study's dataset features. After obtaining the variance explained by a projection to each principal component, the number of components necessary to explain 95% of the total variance was selected. Thus, it was possible to get a dimensionality that best suited each n-gram set. The correspondence between text summarization strategy, n-grams size, and the number of components varied from 655 components for the full text and n-grams of size (3,4) to 550 components for the concepts and relations text with n-grams of size (1,2). Consequently, it was possible to grid search on various text summaries, n-gram sizes, and their corresponding number of features.

Differently, it was not necessary to apply dimensionality reduction techniques to text representations that used LDA because, in this case, the vector dimension is specified by the number of topics. This study tested configurations with 20, 40, and 80 topics. The last step in the first stage classifier was the classification algorithm, for which the same implementations described in Sect. 3.5 were investigated (Fig. 3).

Following the literature considerations about out-of-sample prediction and model generality, using timely ordered data is mandatory for cross-validation and real-world applicability. Despite being a challenging premise, it was possible to demonstrate that reasonably high performance is still achievable. The samples for training were limited to only that available before all decisions in the test set. On the other hand, the model used a novel approach to divide the samples between the training and test sets. The previously adopted strategies considering the temporal dimension consisted of ordering the samples and defining cutting points on

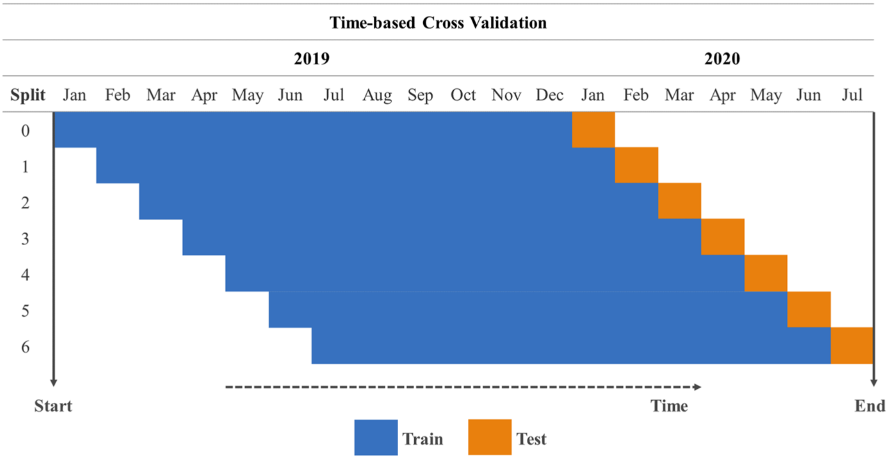


Fig. 4 Cross-validation strategy. Adapted from "Time based cross validation" by Herman-Saffar (2020)

time for training and testing sets, e. g., setting the training set's final date to be December 31st every year and predicting an arbitrary timeframe within the following year. As proposed by Herman–Saffar (2020), our study used time-based cross-validation (see Fig. 4).

The predictor was trained with samples taken from 12 months and tested on samples corresponding to an adjacent month. All test sets were constrained to start after January 1st 2020. Consequently, we utilized seven train/test splits for cross-validation based on a time-based splitting strategy, selecting the best model by the best mean F1 weighted score. The significant difference in this approach is that the splitting points slide along with the samples, and all of them may be used to apply time-based cross-validation. It is worth mentioning that this method creates sets containing a fixed time frame instead of a fixed number of records. Table 2 present the train-test splits.

This study did not aim to evaluate the best train-test split scheme or fine-tune the classifiers' settings but to evaluate the hypothesis of better performance of a cascading model over metadata-only or text-only models. Consequently, the search space for hyperparameters on classifiers included only the learning rate control and the classifier base architecture (activation functions, number of layers, nodes, neighbors, leaves, and estimators). Table 3 summarizes the parameter space used on grid-search for the first stage.

F1 Score was chosen as the primary evaluation metric because it best suits datasets with imbalanced classes. It is an evenly-weighted combination of Precision and Recall, defined as the harmonic mean of these two metrics. The Precision is the ratio between the number of positive samples classified as positive (true positives) and all the samples predicted to be positive. It is also called positive predictive value (PPV). The Recall represents the ratio between the *true positives* and the positive samples (sum of true positives and false negatives).

Table 2 Train-test split for 12 months × 1 month

Split	Train period	Test period	Train (1) samples	Train (0) samples	Test (1) samples	Test (0) samples	Test/Train proportion (%)
0	2019–01-01–2020–01-01	2020–01-01–2020–02-01	175	85	42	18	23
1	2019–02-01–2020–02-01	2020–02-01–2020–03-01	216	103	26	0	8
2	2019–03-01–2020–03-01	2020–03-01–2020–04-01	240	86	81	13	29
3	2019–04-01–2020–04-01	2020–04-01–2020–05-01	320	99	76	15	22
4	2019–05-01–2020–05-01	2020–05-01–2020–06-01	393	110	97	36	26
5	2019–06-01–2020–06-01	2020–06-01–2020–07-01	486	135	150	8	25
6	2019–07-01–2020–07-01	2020–07-01–2020–08-01	635	139	64	78	18
	Test/Train proportion mean						22

Table 3 First stage parameter space

Pipeline step	Algorithm	Unchanging hyperparameters	Hyperparameters search space		
Vectorization	TF-IDF + SVD	Maximum document frequency	Text representation	N-gram range	Components
		0.8	Full	(1, 2)	605
			Concepts	(1, 2)	550
			Concepts and relations	(1, 2)	550
			Full	(2, 3)	637
			Concepts	(2, 3)	595
			Concepts and relations	(2, 3)	593
			Full	(3, 4)	655
			Concepts	(3, 4)	620
			Concepts and relations	(3, 4)	616
	LDA	–	Number of topics: 20, 40, 80		
Classification	k-nearest neighbors	–	Number of neighbors: 3, 5, 7 Weights: uniform, distance inverse		
	Neural network	–	Hidden layer sizes: 100, 200, (100, 100) Activation function: relu, logistic Solver for weight optimization: adam, lbfgs Alpha: 0.01, 0.03, 0.1, 0.3, 1.0		
	Decision tree	–	Splitter: best, random Minimum number of samples in a leaf: 1, 6, 11 Maximum number of features: all, log2		
	Bagging classifier	–	Number of estimators: 10, 25, 50		
	Random forest	–	Number of estimators: 10, 50, 100 Minimum number of samples in a leaf: 1, 6, 11 Maximum number of features: all, log2		
	XGBoost	–	Number of estimators: 25, 50, 100 Minimum weight needed in a child: 1, 6, 11		
	Support vector machine	–	Kernel: radial basis function, polynomial, linear Regularization: 0.001, 0.01, 0.1, 1.0		

The weighted average implementation of Scikit-Learn was used to select the best classifier for each stage. This case calculates the F1 Scores for the positive and negative classes and finds their average weighted by the Support (the number of true instances for each label). Otherwise, because of the higher occurrence of positive samples in the dataset, it would be possible to have good F1 Scores even if the model always predicted positives.

The best overall result was obtained with a Random Forest Classifier, using LDA vectors created from the full-text representation. None of the vectorizers or text representations consistently outperformed other configurations. The first stage grid-search results are summarized in Table 4, represented by the F1 weighted scores. The best scoring setup is highlighted.

Finally, two techniques were tested to attenuate the imbalance effects generated by an asymmetrical class distribution of 3:1 towards the positive outcome. Random oversampling of the minority class and synthetic minority over-sampling technique (SMOTE) (Chawla et al. 2002) were integrated into the Random Forest classifier assembly. Still, they did not improve the model performance.

3.7 Second stage classifier baseline

To compare the results of the proposed model with metadata-only classifiers, we performed a hyperparameter search on the classifiers cited in this study employing only non-textual features. Unlike the text-based predictions, the results summarized in Table 5 show the gradient boosting-based algorithm as the best classifier.

3.8 Second stage classifier

The classifiers of the first stage (Table 3) were later applied to predict outcomes for the entire dataset, and the result was made available to the second stage as a new feature. Thus, the variables were the same as those used for the metadata-only predictors added from the previous step predictions. Dimensionality reduction was employed to avoid overfitting, and finally, grid-search with the same classifiers mentioned in Sect. 3.5 was applied in the second stage.

The entire cascade must be refitted for each training/test split to select the best second-stage classifier. If not, there will be *statistical leakage* between the phases, *i.e.*, a classifier fitted in the first stage to the entire dataset would already incorporate the whole dataset's characteristics. It means including data that would not be available when training the second stage in a real-world problem. For the same reason, the transformations implemented in *infringementID* needed to be included in each training/test split.

4 Results and discussion

As mentioned in Sect. 3.8, a grid-search among the classifiers from Sect. 3.5 was performed, with cross-validation according to the technique described in Sect. 3.6. Table 6 presents the results obtained in three assemblies: the first stage (baseline 1), built only on n-grams extracted from infringements' text, the baseline 2, using only infringements' metadata, and the proposed cascade classifier (composed of the first and second stages).

The results show that the two-stage model surpassed both the *n*-gram and the metadata-based approaches. It obtained the best F1-weighted, benefiting from the prediction extracted from the text. Still, the full-text representation provided the

Table 4 First stage classifier F1 weighted scores per classifier, vectorization, and text representation

	Classifier	F1 Score		
		Full	Concepts	Concepts and relations
Vectorization X Classifier X Text representation				
TF-IDF	k-nearest neighbors	0.838	0.854	**0.868**
	Neural network	**0.852**	0.851	0.851
	Decision tree	**0.863**	0.831	0.857
	Bagging classifier	0.840	0.849	**0.865**
	Random forest	0.838	0.849	**0.866**
	XGBoost	**0.874**	0.872	0.862
	Support vector machine	**0.851**	0.850	0.850
LDA	k-nearest neighbors	**0.864**	0.861	0.832
	Neural network	0.851	**0.885**	0.840
	Decision tree	0.841	**0.847**	0.830
	Bagging classifier	0.844	0.828	**0.859**
	Random forest	**0.887**	0.843	0.854
	XGBoost	0.852	**0.857**	0.830
	Support vector machine	0.823	**0.823**	0.780

Values in bold represent the best score for a classifier among different text representations

best results for feeding the cascade classifier with the first stage predictions. With a higher number of n-grams, this representation seems to benefit from more variance in the input data, despite being computationally more expensive.

The best overall performance was obtained with a Random Forest classifier applied on LDA topics in the first stage and an XGBoost classifier in the second stage. Its F1 weighted score was 1.24% higher than the best score achieved with a baseline.

The confusion matrix in Table 7 provides more insights through the class-wise performance distribution of each assembly. Since the test sets were non-overlapping, the results were totalized by summing each train-test split result. Compared to the first baseline and second-best model, the cascade classifier showed a slightly worse performance when predicting the positive class. On the other hand, it significantly improved the performance in predicting the negative class, better limiting false positives. This feature deserves to be explored more extensively as it may be particularly desirable in predictions applied to the legal context.

The standard deviation obtained by the proposed approach is compared to the standard deviations of the two baselines in Table 8. The cascading classifier has a higher overall standard deviation (and variance) than the text-based classifier. This model will return results varying in a broader range when predicting future data. It can be beneficial to the final predictor due to the bias-variance tradeoff.

Bias refers to assumptions in the learning algorithm that narrow the scope of what can be learned. It can accelerate learning and lead to stable results at the cost

Table 5 Metadata-only classifiers F1-weighted scores

	Classifier	F1 score
Baseline 2 (metadata-based)	k-Nearest neighbors	0.818
	Neural network	0.771
	Decision tree	0.842
	Bagging classifier	0.851
	Random forest	0.848
	XGBoost	**0.852**
	Support vector machine	0.790

The value in bold identifies the classifier used as the baseline 2

Table 6 F1 weighted score compared text-based prediction (first stage and baseline 1), metadata-based prediction (second stage and baseline 2), and the final assembly

F1 weighted score	
Classifier	F1 score
Baseline 1 (text-based, Random Forest)	0.887
Baseline 2 (metadata-based, XGBoost)	0.852
Cascade classifier (First stage, Random Forest) (Second stage, XGBoost)	0.900

of the assumption differing from reality (Browlee 2018). In this sense, models built from the cascading method tend to learn more from new data while balancing the bias-variance tradeoff compared to metadata-based models. Considering that the small size of the dataset is a cause for high variance, fitting the model on more training data can reduce the overall variance.

Table 9 compares the baselines and the proposed model according to the Area Under the Receiver Operating Characteristic Curve (ROC AUC) scores. The ROC plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various probability thresholds, *i.e.*, the probability of belonging to the majority class. Also, from this perspective, the cascade classifier performs better by differentiating classes in different probability thresholds.

Taking the training/test split in which the cascade classifier obtains the best results compared to the baselines (test split 2), Fig. 5 presents the Receiver Operating Characteristic (ROC) for a range of probability threshold values on the classification algorithms' predictions. Two probability thresholds are marked as examples.

At the discrimination point of 0.691 for the majority class, the cascade classifier has an FPR of 0.462, and the TPR is 0.988. For an FPR of 0.231 and a TPR of 0.802, the baseline classifier threshold is 0.898. The curve represents the Recall (or True Positive Rate—TPR) versus the False Positive Rate (FPR). It is a standard technique for presenting a classifier performance over a range of tradeoffs between TPR (benefits) and FPR (costs). The optimal performance, or the optimal probability cut-point, combines the highest benefit with an acceptable cost, depending on the problem to which the model is applied.

Table 7 Confusion matrix when all test samples are considered

Baseline 1 (text-based)		**Predicted**	
		1	0
Actual	1	524	13
	0	62	105

Baseline 2 (metadata-based)		**Predicted**	
		1	0
Actual	1	500	37
	0	64	103

Cascade classifier		**Predicted**	
		1	0
Actual	1	518	19
	0	49	118

Table 8 Comparison of standard deviations between the proposed model and the baselines

Classifier	Standard deviation
Baseline 1 First stage (text-based)	0.037
Baseline 2 (metadata-based)	0.078
Cascade	0.050

The example shown in Table 10 for the sample under analysis highlights the probability threshold of 0.691, where the cascade classifier has a TPR of 0.988 and an FPR of 0.462. The choice model depends on the tradeoff that best suits the task under analysis.

The improved performance of the cascade classifier becomes even more evident when the ROC curves are cross-validated using all training/test time splits (Fig. 6). The cascade classifier outperforms the baselines in most splits. It shows a higher mean value for the Area Under the ROC curve (AUC), indicating better discrimination power between the two classes.

In their turn, the Precision-Recall curves of Fig. 7 show a comparison of the tradeoffs between Precision and Recall for different probability thresholds. Also, under this perspective, the cascade classifier outperforms the baselines with a better tradeoff behavior. We can observe a drop in Precision when the Recall achieves values closer to 0.7 in the metadata-only and text-only baselines. However, the cascade classifier maintains Precision with a smooth decay path as Recall increases.

These results are consistent with the work premise, *i.e.*, the cascade classifier proposed in this work outperforms both the metadata-only and the text-only assemblies. The results also reveal that despite being a more challenging task to take the time dimension into account while maintaining out-of-sample applicability, it is possible to achieve reasonably high results, as the authors observed when working with this dataset. Additionally, the difference in performance observed in the characteristic curves shows the significance of evaluating this modeling in prediction tasks when text and metadata are available. We emphasize the importance of continually incorporating new judgments into the dataset to reduce variance.

We also wanted to evaluate the relative performance between the models through statistical tests, i.e., whether the results ensured a better performance of the cascade classifier. For this purpose, we resorted to Dietterich (1998) to determine McNemar's test as the one to be used. The author argues that, given two learning algorithms and a small dataset, none of the tests analyzed can answer which one will

Table 9 Comparison of ROC AUC scores between the proposed model and the baselines

Classifier	ROC AUC
Area under the receiver operating characteristic curve (ROC AUC) scores	
Baseline 1 First stage (text-based)	0.820
Baseline 2 (metadata-based)	0.823
Cascade	0.861

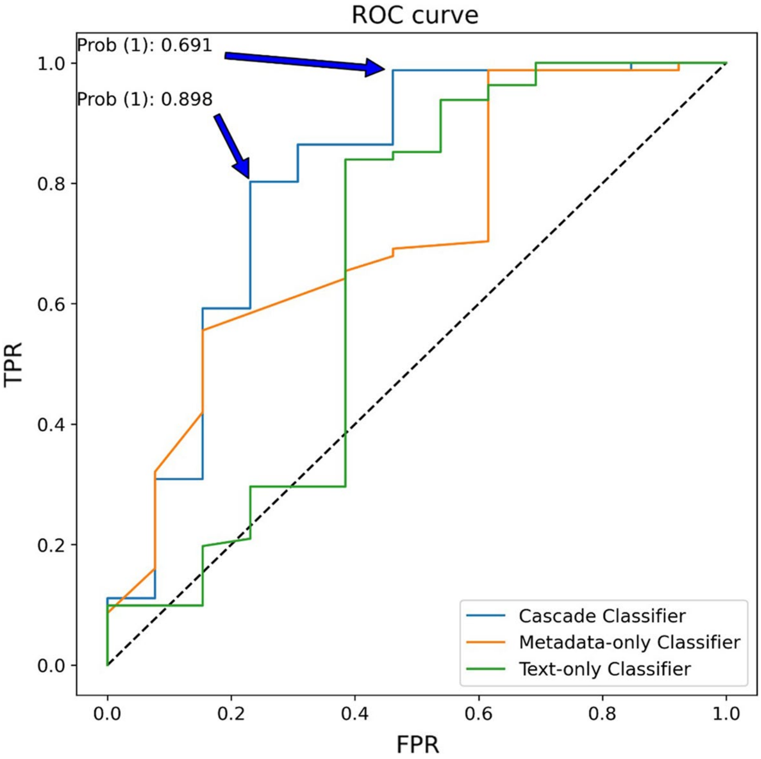


Fig. 5 ROC curves for cascade and baseline classifiers

produce more accurate classifiers when trained on a dataset of the same size and obtained from the same population, limiting the result to the training sample used. In addition, he emphasizes that each statistical test has its limitations, and the results must be interpreted as approximations.

The null hypothesis in our study is that the assemblies have the same error rate, i.e., the number of samples misclassified by one model but not by the other is equal to the number of samples misclassified by the second model and not by the first. Comparing the metadata-based classifier with the cascade classifier, we obtained a *p* value of 0.000090, rejecting the null hypothesis. In comparing the text-based

Table 10 TPR and FPR for different discrimination thresholds (sample split 2)

TPR	FPR	Probability threshold
0.111	0.000	0.990
0.309	0.077	0.978
0.593	0.154	0.958
0.802	0.231	0.898
0.864	0.308	0.825
0.988	**0.462**	**0.691**
1.000	0.846	0.167

The example mentioned in the text above is highlighted in bold

and the cascade classifier, the result was equal to 0.208668, the reason why the null hypothesis cannot be rejected.

Finally, we use SHAP values, an additive feature attribution method, to approximate each feature's contribution to the prediction and reverse-engineer the output, providing local explainability. The best explanation of a simple model is the model itself; it perfectly represents itself and is easy to understand. However, we cannot use the original model as its own best explanation for complex models, such as ensemble methods or deep networks, because it is not easy to understand. Instead, we must use a simpler explanation model, which we define as any interpretable approximation of the original model (Lundberg and Lee 2017).

SHAP assigns each feature an importance value for a particular prediction. Our study used complex models as classifiers in the standalone stages. However, we preserved the visibility of the features by using TF-IDF vectors or LDA topics in the first stage and metadata in the second stage.

We present the cascade classifier with the best overall performance as an example. The first stage is constituted by an LDA transformer whose output is fed into a Random Forest classifier. LDA assumes that each document is represented by a distribution of a fixed number of topics, and each topic is a distribution of words. For a random infringement whose prediction in both stages was "positive" or one, Fig. 8 shows the influence of the most significant topics in the first stage prediction. The base value is the averaged predicted probability across all samples. The red topics positively influence the prediction (driving it beyond the base value). The blue arrows represent the topics that drive the prediction down. The bold value is the actual prediction for this sample.

The relevance of an n-gram to a topic, in its turn, is defined as:

$$r(w,k|\lambda) = \lambda \log \phi_{kw} + (1-\lambda) \log \frac{\phi_{kw}}{p_{kw}} \tag{1}$$

Equation (1) Relevance of an n-gram to a topic.

In Eq. (1), ϕ_{kw} is the probability of an n-gram w occurring in topic k, and ϕ_kw/p_kw is the lift calculated by the n-gram's probability within a topic and its marginal probability across the entire corpus. The second term helps to discard globally frequent n-grams. Therefore, a lower λ gives more importance to n-grams' topic

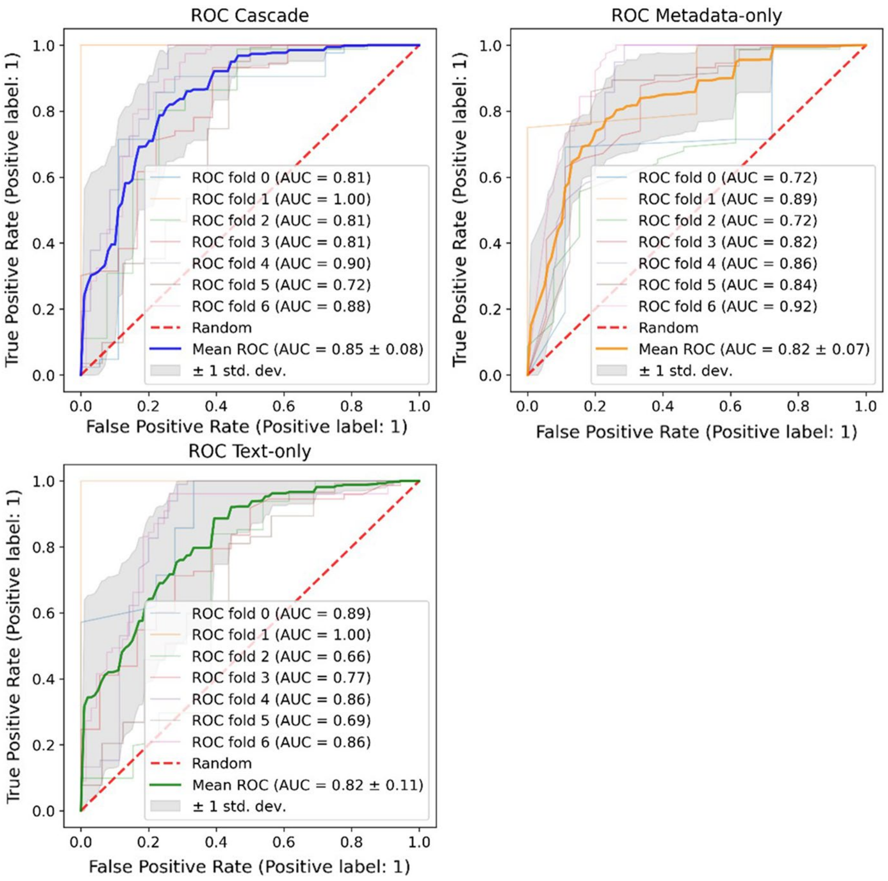
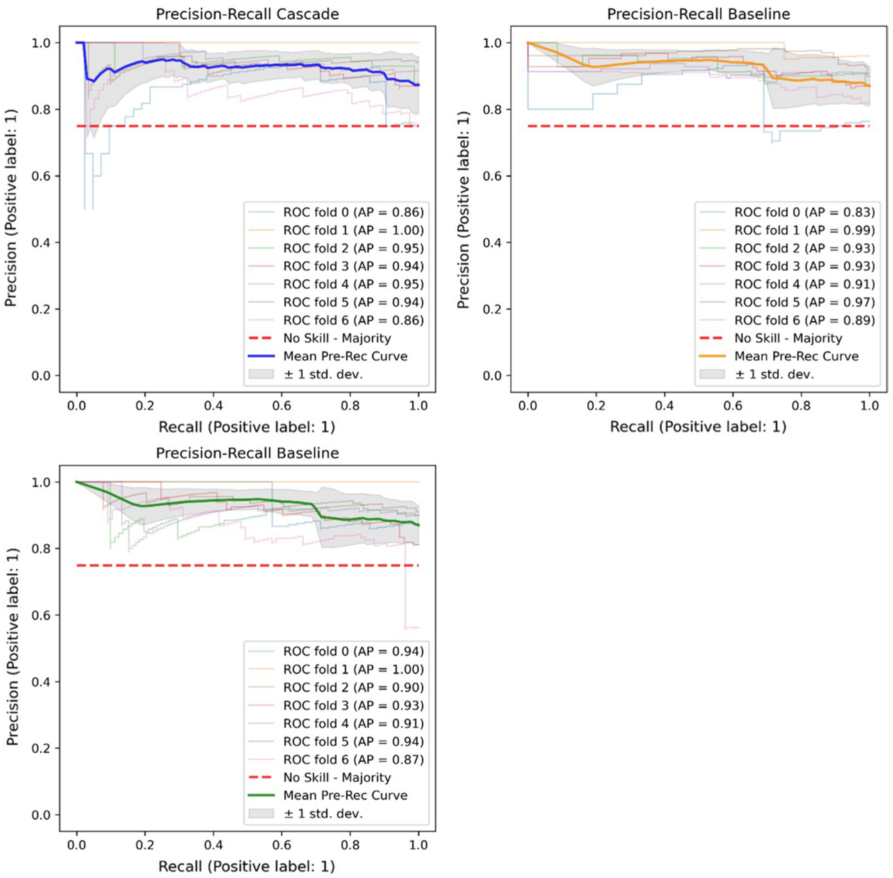


Fig. 6 Cross-validated ROC curves

exclusivity. Using the pyLDAvis library (Mabey and English 2015), we visualize the topics' interrelation and *n*-grams relevance. In Fig. 9, by lowering λ to 0.5, we can find that topic 31 top-ranked stemmed *n*-grams related to ethics in the Portuguese language, *e.g.*, "étic", "comit étic" and "códig étic".

Next, we use the SHAP values to estimate the importance of variables in the second stage (Fig. 10), which employs an XGBoost Classifier. In this case, the time difference between the infringement date and January 1st, 2000 ("daysFrom2000"), the infringed regulation ("5534"), and the previous stage prediction ("pred_subsistente") positively influenced the output. The time difference between the decision date and the infringement date ("tempoAteJulgar") drove the prediction towards the negative output.


Precision-Recall Cascade

Precision-Recall Baseline

Precision-Recall Baseline

Fig. 7 Cross-validated precision-recall curves

5 Conclusions

This study employed several NLP and ML techniques through an innovative approach, combining different types of data with state-of-the-art classifiers through a cascade assembly. Previous studies have used text and metadata separately, and in this study, both types contribute to the prediction. The cascade approach was chosen to preserve visibility into the contributions of variables from both stages.

We obtained an F1 score of 0.900 with a Precision of 0.929 and a Recall of 0.873. These results mean that in the vast majority of cases, the system correctly predicts the final decision given the initial document of the proceeding. As seen in Fig. 7, our cascaded approach provides a robust estimate, in particular eliminating the drop in performance for recalls greater than 0.65 that occurs when classifiers are

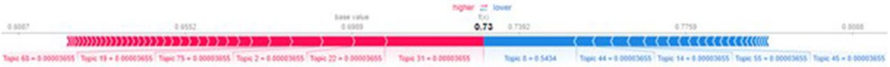


Fig. 8 SHAP values for the first stage prediction. Topics in red influence prediction positively, and topics in blue influence negatively

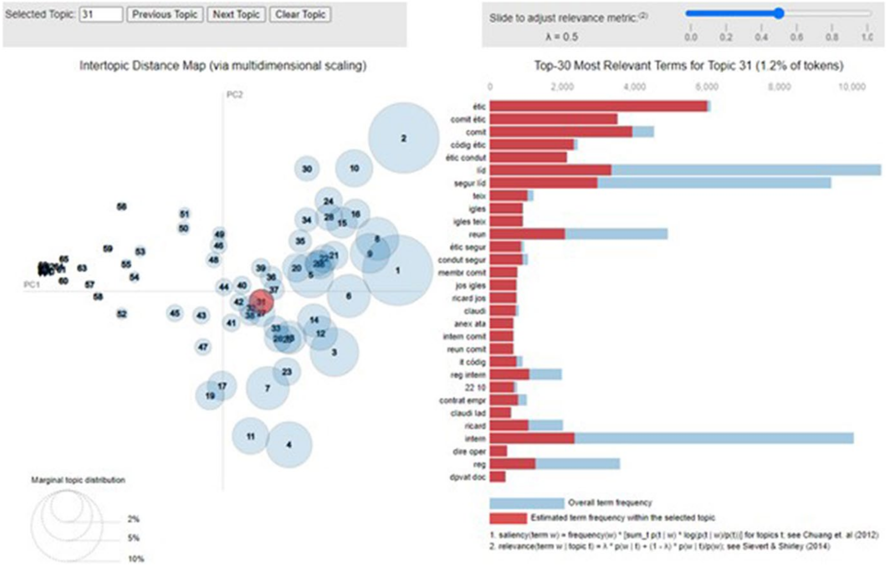

Selected Topic: 31
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 0.5
0.0 0.2 0.4 0.6 0.8 1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 31 (1.2% of tokens)
0 2,000 4,000 6,000 8,000 10,000
ètic
comit ètic
comit
còdig ètic
ètic condut
lid
segur lid
teix
igles
igles teix
reun
ètic segur
condut segur
membr comit
jos igles
ricard jos
claudi
anex ata
intern comit
reun comit
it còdig
reg intern
22 10
contrat empr
claudi lad
ricard
intern
dire oper
reg
dpvat doc
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Fig. 9 Topics relatedness and n-grams relevance on topic 31

used separately. This approach allows for very high Precision, even with the highest Recall values.

This study also addressed the time dependency of judicial decisions innovatively. Many reviewed approaches did not take time into account, in effect using all data even to predict decisions that occurred in the past. In this study, we only used data samples that occurred before the decision was predicted. This limitation naturally leads to a drop in performance, but thanks to the cascade approach, we managed to have an acceptable performance, even taking time into account.

The good results obtained in the paper were achieved using only the first document from each proceeding, *i.e.*, neither external variables nor any particular assumption about the data was used. Therefore, our method allows predicting the final outcome as soon as the decision date can be estimated, which is precious information for practitioners saving time and resources.

A possible application of this method is to aid decision-makers with predictions, along with the preliminary analysis done by legal analysts. If the analysis

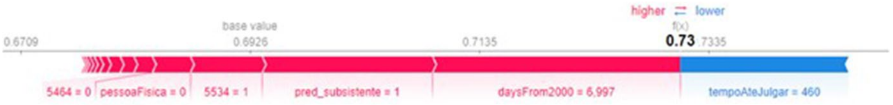


Fig. 10 SHAP values for the second stage prediction

and the result suggested by the model point in the same direction, it might increase the conviction in decision-making. On the other hand, a contradiction between the preliminary analysis and the model's result would undoubtedly lead to a more detailed assessment of the case. In both cases, the assertiveness of the result may be increased.

Nonetheless, identifying variables that influence the classification of infringements is critical to the model's usefulness. In this sense, by using SHAP, the state-of-the-art method to identify the variable contributions, and by preserving the visibility of the variables extracted from the text, it is possible to explain each prediction to the user. Other important information, such as predictions of similar cases (precedents), may also be used to inform courts.

Despite people not expecting a future in which trained machines can substitute decision-makers, this study demonstrates how an ensemble model can improve existing decision-support models. As previously stated, the more accurate the prediction model, the better the advice it provides to administrative courts, offering agility and consistency in decisions by facilitating jurisprudence usage.

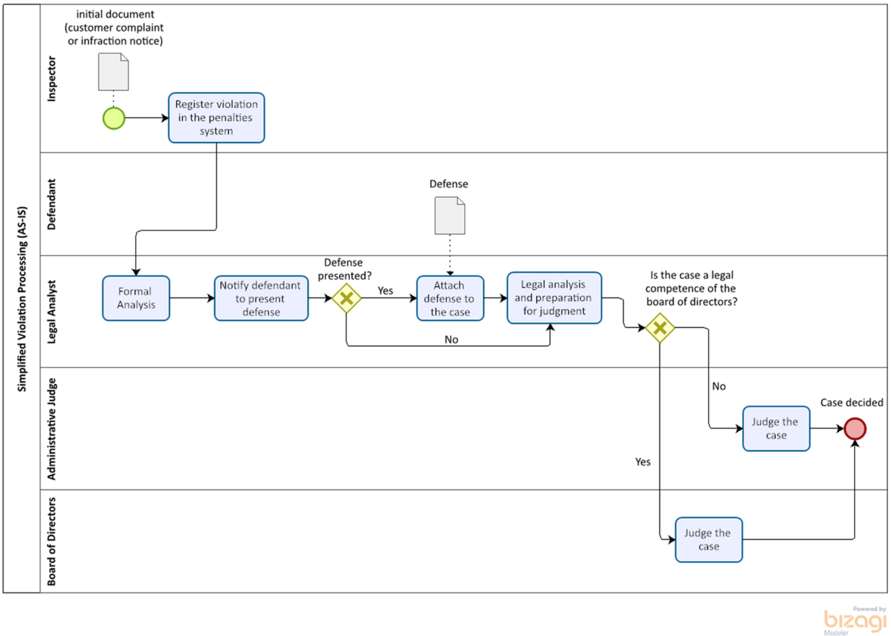
The results confirm that it is possible to improve administrative court decisions' efficiency and consistency using AI-based legal assistance techniques. The present study showed objective and robust evidence that the proposed model, applicable when metadata and textual features are available, reliably outperforms the most used models in important metrics by a gain rate that makes its implementation attractive. The predictive performance was also probed under generalization and out-of-sample applicability conditions.

5.1 Limitations and future work

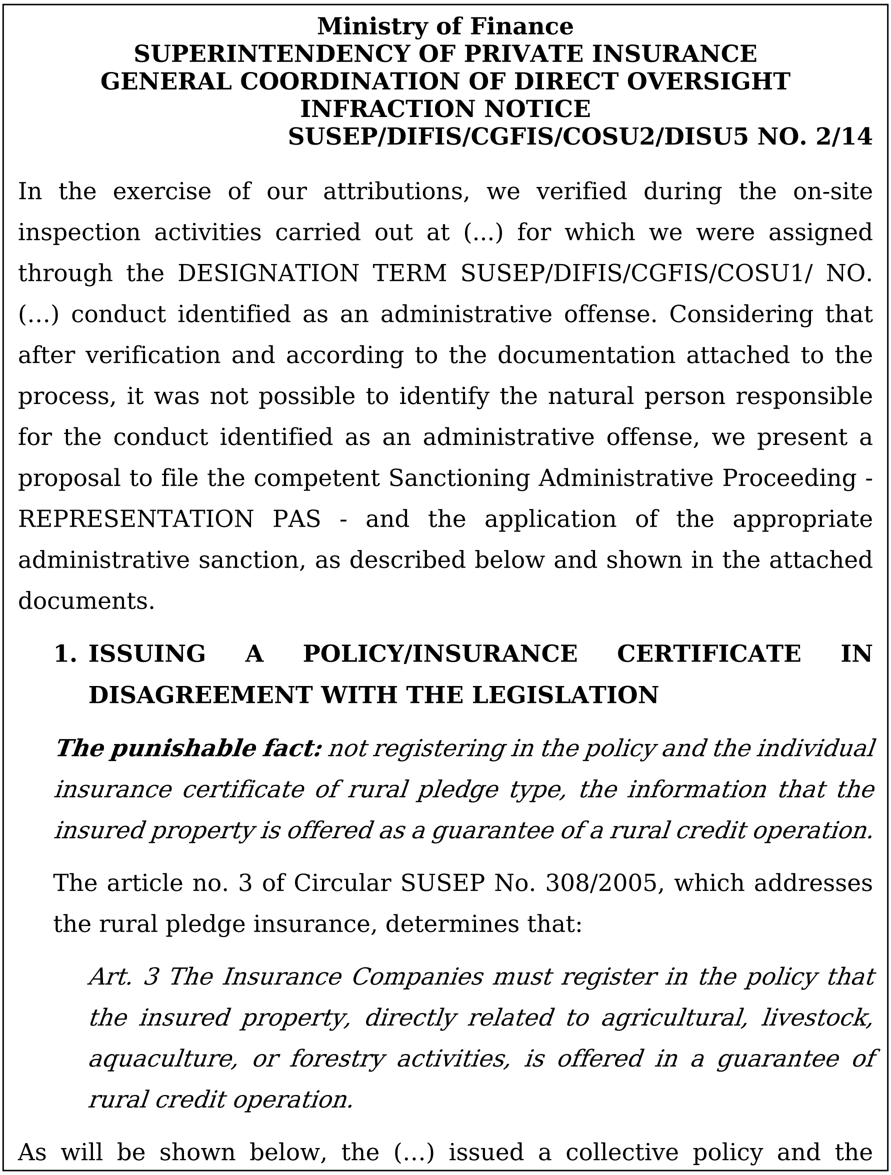
The promising results pave the way for future work to investigate whether the proposed approach is the right choice for multiclass problems. Furthermore, larger datasets and ensembles of predictions from two-stage models may be explored to introduce bias and decrease the variance. The possible contribution of using semantic analysis for extracting forecasts in the first stage is another topic that deserves further investigation from the author's point of view. There is still a vast field to investigate models that offer global explainability so that humans can easily interpret how classifications are carried out. Like others, this study is not without limitations. Due to restricted data available, further tests should be conducted to validate the model's stability. Reproducing these results may depend on the organization and the type of problems addressed. Furthermore, employing the stemming technique to reduce the feature space in the first stage brought adverse consequences to local explainability from topics and *n*-grams.

The cascade model is meant to be applied as a real-time query system, recommending decision outcomes when a new infringement case and the respective initial document are analyzed. In this scenario, predicting decisions 1 month in advance, for instance, may require a well-suited pipeline for data integration and training. In this case, models need to be pre-trained and updated promptly. Therefore, future studies should also focus on understanding how similar models could be deployed in production environments.

Appendix 1: Infringement examination business process


Simplified Violation Processing (AS-IS)
Inspector
Defendant
Legal Analyst
Administrative Judge
Board of Directors
initial document (customer complaint or infraction notice)
Register violation in the penalties system
Defense
Formal Analysis
Notify defendant to present defense
Defense presented?
Yes
No
Attach defense to the case
Legal analysis and preparation for judgment
Is the case a legal competence of the board of directors?
No
Yes
Judge the case
Case decided
Judge the case
Powered by bizagi Modeler

Appendix 2: Infraction notice (English translation)



Ministry of Finance
SUPERINTENDENCY OF PRIVATE INSURANCE
GENERAL COORDINATION OF DIRECT OVERSIGHT
INFRACTION NOTICE
SUSEP/DIFIS/CGFIS/COSU2/DISU5 NO. 2/14

In the exercise of our attributions, we verified during the on-site inspection activities carried out at (...) for which we were assigned through the DESIGNATION TERM SUSEP/DIFIS/CGFIS/COSU1/ NO. (...) conduct identified as an administrative offense. Considering that after verification and according to the documentation attached to the process, it was not possible to identify the natural person responsible for the conduct identified as an administrative offense, we present a proposal to file the competent Sanctioning Administrative Proceeding - REPRESENTATION PAS - and the application of the appropriate administrative sanction, as described below and shown in the attached documents.

1. **ISSUING A POLICY/INSURANCE CERTIFICATE IN DISAGREEMENT WITH THE LEGISLATION**

The punishable fact: *not registering in the policy and the individual insurance certificate of rural pledge type, the information that the insured property is offered as a guarantee of a rural credit operation.*

The article no. 3 of Circular SUSEP No. 308/2005, which addresses the rural pledge insurance, determines that:

> *Art. 3 The Insurance Companies must register in the policy that the insured property, directly related to agricultural, livestock, aquaculture, or forestry activities, is offered in a guarantee of rural credit operation.*

As will be shown below, the (...) issued a collective policy and the

Acknowledgements The authors acknowledge Brazil's Superintendency of Private Insurance for supporting and providing data for this work.

Funding Open access funding provided by FCT|FCCN (b-on). This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

Aletras N, Tsarapatsanis D, Preoţiuc-Pietro D, Lampos V (2016) Predicting judicial decisions of the European court of human rights: a natural language processing perspective. PeerJ Comput Sci 2016(10):1–19. https://doi.org/10.7717/peerj-cs.93

Bibal A, Lognoul M, De Streel A, Frénay B (2021) Legal requirements on explainability in machine learning. Artif Intell Law 29(2):149–169. https://doi.org/10.1007/s10506-020-09270-4

Bird S, Klein E, Loper E (2009) Natural language processing with python. O'Reilly Med. https://doi.org/10.5555/1717171

Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3(4–5):993–1022. https://doi.org/10.1016/b978-0-12-411519-4.00006-9

Brill E (1992) A simple rule-based part of speech tagger. In: Proceedings of the third conference on applied natural language processing. Association for Computational Linguistics. https://doi.org/10.3115/974499.974526

Browlee J (2018) How to reduce variance in a final machine learning model. Mach Learn Mast. https://machinelearningmastery.com/how-to-reduce-model-variance/

Cer D, Yang Y, Kong SYI, Hua N, Limtiaco N, John SR, Constant N, Guajardo-Céspedes M, Yuan S, Tar C, Sung YH, Strope B, Kurzweil R (2018) Universal sentence encoder. In: EMNLP 2018–conference on empirical methods in natural language processing: system demonstrations, Proceedings. https://doi.org/10.18653/v1/d18-2029

Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357

Chen DL, Eagel J (2017) Can machine learning help predict the outcome of asylum adjudications? In: Proceedings of the international conference on artificial intelligence and law, pp 237–240. https://doi.org/10.1145/3086512.3086538

Chen T, Guestrin C (2016) XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM sigkdd international conference on knowledge discovery and data mining, pp 785–794. https://doi.org/10.1145/2939672.2939785

Chen L (2009). Curse of dimensionality. In: Encyclopedia of database systems pp 545–546. Springer. https://doi.org/10.1007/978-0-387-39940-9_133

Devlin J, Chang MW, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL HLT 2019–2019 conference of the north american chapter of the association for computational linguistics: human language technologies–proceedings of the conference, vol 1, pp 4171–4186. https://github.com/tensorflow/tensor2tensor

Dietterich TG (1998) Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput 10(7):1895–1923. https://doi.org/10.1162/089976698300017197

Fonseca ER, Rosa JGL, Aluísio SM (2015) Evaluating word embeddings and a revised corpus for part-of-speech tagging in Portuguese. J Br Comput Soc. https://doi.org/10.1186/s13173-014-0020-x

Gama J, Brazdil P (2000) Cascade generalization. Mach Learn 41(3):315–343. https://doi.org/10.1023/A:1007652114878

Herman-Saffar O (2020) Time based cross validation. Towards Data Science. https://towardsdatascience.com/time-based-cross-validation-d259b13d42b8

IAIS (2017) Insurance core principles. https://www.iaisweb.org/file/69922/insurance-core-principles-updated-november-2017

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. https://doi.org/10.1371/journal.pone.0174698

Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: 31st International conference on machine learning, ICML vol 4, pp 2931–2939

Luhn HP (1957) A statistical approach to mechanized encoding and searching of literary information. IBM J Res Dev 1(4):309–317. https://doi.org/10.1147/rd.14.0309

Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. In: Proceedings of the 31st international conference on neural information processing systems, pp 4768–4777

Mabey B, English P (2015) *pyLDAvis* (2.1.2). https://pyldavis.readthedocs.io/en/latest/

Medvedeva M, Vols M, Wieling M (2020) Using machine learning to predict decisions of the European court of human rights. Artif Intell Law 28(2):237–266. https://doi.org/10.1007/s10506-019-09255-y

Mikolov T, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: NIPS'13: proceedings of the 26th international conference on neural information processing systems, vol 2, pp 3111–3119

Nason S (2018) Administrative justice can make countries fairer and more equal—if it is implemented properly. The Conversation. https://theconversation.com/administrative-justice-can-make-countries-fairer-and-more-equal-if-it-is-implemented-properly-108238

Orengo VM, Huyck C (2001) A stemming algorithm for the portuguese language. In: Proceedings 8th symposium on string processing and information retrieval, pp 186–193. https://doi.org/10.1109/spire.2001.989755

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 324:2825–2830

Pennington J, Socher R, Manning CD (2014) GloVe: global vectors for word representation. In: EMNLP 2014–2014 conference on empirical methods in natural language processing, proceedings of the conference, pp 1532–1543. https://doi.org/10.3115/v1/d14-1162

Pillai VG, Chandran LR (2020) Verdict prediction for indian courts using bag of words and convolutional neural network. In: Proceedings of the 3rd international conference on smart systems and inventive technology, ICSSIT 2020, pp 676–683. https://doi.org/10.1109/ICSSIT48917.2020.9214278

Richardson L (2007) BeautifulSoup. https://www.crummy.com/software/BeautifulSoup/

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 Rev 104(4):1150–1210. https://doi.org/10.2307/4099370

Shinyama Y, Guglielmetti P, Marsman P (2019) pdfminer.six. https://github.com/pdfminer/pdfminer.six

Sivaranjani N, Jayabharathy J, Teja PC (2021) Predicting the supreme court decision on appeal cases using hierarchical convolutional neural network. Int J Speech Technol 24(3):643–650. https://doi.org/10.1007/s10772-021-09820-4

Spärck Jones K (1972) A statistical interpretation of term specificity and its application in retrieval. J Document 28:11–21. https://doi.org/10.1108/00220410410560573

Statista (2020) Global insurance industry–statistics and facts. https://www.statista.com/topics/6529/global-insurance-industry/

SUSEP (2020a) 8° Relatório de Análise e Acompanhamento dos Mercados Supervisionados. pp 1–24. http://www.susep.gov.br/menuestatistica/SES/relat-acomp-mercado-2020a.pdf

SUSEP (2020b) Brokers statistics. https://www2.susep.gov.br/safe/Corretores/estatisticas

Theodoridis S (2020) Machine learning: a bayesian and optimization perspective, 2nd edn. Elsevier, Amsterdam

Authors and Affiliations

Hugo Mentzingen[1] · Nuno Antonio[1] · Victor Lobo[1,2]

✉ Hugo Mentzingen
hsilva@novaims.unl.pt

Nuno Antonio
nantonio@novaims.unl.pt

Victor Lobo
vlobo@novaims.unl.pt

[1] NOVA Information Management School (NOVA IMS), Universidade Nova de Lisboa, Campus de Campolide, 1070-312 Lisbon, Portugal

[2] CINAV, Portuguese Naval Academy, Alfeite, 2810-001 Almada, Portugal