Article

Identification of judicial outcomes in judgements: A Generalized Gini-PLS approach

Gildas Tagny-Ngompé[1], **Stéphane Mussard**[2], **Guillaume Zambrano**[2], **Sébastien Harispe**[1] **and Jacky Montmain**[1]

[1] EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Alès, Alès, France
[2] UNIV. NIMES, CHROME
* Correspondence: tagnyngompe@gmail.com

Version July 17, 2020 submitted to Stats

Abstract: This paper presents and compare several text classification models that can be used to extract the outcome of a judgment from justice decisions, i.e. legal documents summarizing the different rulings made by a judge. Such models can be used to gather important statistics about cases, e.g. success rate based on specific characteristics of cases' parties or jurisdiction, and are therefore important for the development of Judicial prediction not to mention the study of Law enforcement in general. We propose in particular the generalized Gini-PLS which better consider the information in the distribution tails while attenuating, as in the simple Gini-PLS, the influence exerted by outliers. Modelling the studied task as a supervised binary classification, we also introduce the LOGIT-Gini-PLS suited to the explanation of a binary target variable. In addition, various technical aspects regarding the evaluated text classification approaches which consists of combinations of representations of judgments and classification algorithms are studied using an annotated corpora of French justice decisions.

Keywords: Gini-PLS; text classification; court decisions; judge opinion identification

1. Introduction

Judicial prediction is the ability to predict what a judge will decide on a given case. Is it possible to develop efficient predictive models to automatize such predictions? This question has long been driving several initiatives at the crossroads of Artificial Intelligence and Law; in particular through the development of predictive models based on the alignment of computable features of the case that were available to the judge prior to the judgment, with computable features of the judge's decision on the case. In this line of works, this paper presents a study towards the development of such predictive models taking advantage of Machine Learning and Natural Language Processing techniques. The legal vocabulary being notoriously ambiguous, we first detail important concepts that will be used thereafter.

A *case* begins with a complaint requesting remedy against the wrongful doing of the defendant. The features of the case are the circumstances existing prior to the filing of the complaint, that is a set of facts sufficient to justify a right to file a complaint.

A *claim* is a request made by a plaintiff against a defendant, seeking legal remedy. Claims can be grouped into different categories, depending on the rule applicable and the type of remedy sought (e.g. injunctive relief, cease and desist order, damages).

A *judgment* summarizes the different rulings made by a judge about a certain case into a document. Judgments therefore contains many features that can be extracted (e.g., type of court, name of the parties, claims made by the parties, judge's decisions on the claims). A complaint is a judgement that can contain many different claims, seeking different types of remedy. Therefore, in general, a judgment concern different types of claims.

The decision is a ruling made on a particular claim. We further consider that the judge's decision on a claim is either accepted or rejected. Note that a judgment must be distinguished from the judge's decision on a specific claim.

These last years, the methodology of judicial predictions were mostly exclusively based on the employ of neural networks, which may be seen as the most flexible models for classification and predictions of legal decisions when large datasets are available. Chalkidis and Androutsopoulos [1] use a Bi-LSTM network running on words on a task of extracting contractual clauses. Wei et al. [2] have shown the superiority of convolutional networks over Suport Vector Machines for the classification of texts on large specific datasets. The use of a Bi-GRU has become a standard approach, see [3]. Performance of 92% was obtained on the identification of criminal charges and on judicial outcomes from Chinese criminal decisions [4]. This type of approaches can also be used successfully on judgements in civil matters [5]. Bi-LSTM networks coupled with a representation of the judgement in the form of a tensor achieve performance around 93% on a corpus of 1.8 million Chinese criminal judgments [6]. This work has been successfully replicated on a body of judgments of the European Court of Human Rights in English, with F-measure performance of 80% for bi-GRU networks with attention, and Hierarchical BERT [7]. On the same corpus, the development of a specific lexical embedding ECHR2Vec makes it possible to reach performances around 86% [8]. Similar performances of 79% are obtained by TF-IDF (Term Frequency - Inverse Document Frequency) in the Portuguese language [9]. Although neural networks enable very good performances to be achieved, we defend in this paper the use of compression machine learning models based on word representations *à la* TF-IDF with different variants corresponding to different weighting schemes. These approaches are particularly suited dealing with small to medium size annotated datasets.

As we stressed, claims can be grouped into specific categories depending on their nature, e.g. several claims may refer to the notion of "child care"; such categories are defined *a priori* by jurists for the analysis of a corpus of judgments of interest. In addition, a judgment most of the time only contains a single claim of a given category.[1] In this context, we are interested in the definition of predictive models able to predict the judge's decision expressed in a judgment for a specific category of claims. Otherwise stated, knowing that a judgment contains a single claim of a given category, the model will have to answer the following question analysing the judgment (textual document): has the claim been accepted or rejected? Developing efficient predictors of the outcome of specific categories of claims is of major interest for the analysis of large corpus of judgments. It for instance paves the way for large statistical analysis of correlations between aspects of the case (e.g. parties, location of the court) and outcomes for specific categories of claims. Such analyses are important for theoretical studies on law enforcement and future development of models able to predict the outcome of cases.[2]

The methodology of judicial predictions therefore depends on the ability of a model to predict the judge's decision on a claim inherent to a given category - without knowing the precise localization of the statement of the judge's decision inside a judgment. In this context, extracting the result of a claim can be formulated as a task of binary text classification. To tackle this task, we consider in this paper the supervised machine learning paradigm assuming that a set of annotated judgments, i.e. labelled dataset, is provided for each category of claims of interest. We therefore aim to use the labelled dataset for training an algorithm to recognize whether the request has been rejected or

[1] A corpus description and a descriptive analysis is provided in the next section.

[2] Note that traditional text classification techniques obtain good performance predicting if a judgment contains a claim of a specific category, see [10]. Obtaining relevant statistics about judge's decisions on a given category of claim would therefore be based on (i) applying aforementioned model to distinguish judgments containing a claim of the category of interest, and (ii) applying the type of models studied in this paper to know the outcome of previously identified judgments.

accepted. Considering this setting, the paper presents various models and empirically compares them on a corpus of French judgments. A statistical analysis of the impact of various technical aspects generally involved in the classification of texts which consists of a combination of representations of judgments and classification algorithms is proposed. This analysis sheds light on certain configurations making it possible to determine judges' decisions of a claim. We also propose the generalized Gini-PLS algorithm which is an extension of the simple Gini-PLS model [11]. This generalized Gini-PLS consists in adding a regularization parameter that makes it possible to better adapt the regression with respect to the information in the distribution tails while attenuating, as in the simple Gini-PLS, the influence exerted by outliers. We also propose a new regression (LOGIT-Gini-PLS) which is better suited to the explanation of a target variable when the latter is a binary variable. These two models have never been applied to text classification.

The paper is organized as follows: Section 2 presents characteristics of the corpus used for this study and motivates the modelling of the task adopted in this paper (i.e. decision outcome prediction as a binary text classification). Section 3 presents the different TF-IDF vectorizations of the judgments. Section 4 presents the proposed generalized (LOGIT) Gini-PLS algorithms for text classification. Section 5 presents our experiments and results. Section 6 concludes our study.

2. Datasets and modelling motivations

We assume in this paper that predicting judge's decisions may be studied through the lens of the definition of binary text classification models. This positioning is based on discussions with jurists and motivated by analyses performed on labelled datasets of French judgments. Six datasets built from a corpus of French judgments are considered in our study, one for each of the six categories of claims introduced in Table 1. A total of 341 judgments have been manually annotated by a jurist.

Dataset	Description	Number of judgments	*# claims*	*# docs*
ACPA	Civil fine for abuse of process	246	*23*	*23*
CONCDEL	Damages for unfair competition	238	*58*	*30*
DANAIS	Damages for abuse of process	421	*208*	
DCPPC	declaration of claim to liabilities of the collective procedure	218	*129*	
DORIS	damages for neighborhood disturbance	164	*103*	
STYX	irrecoverable expenditure	123	*89*	

Table 1. Categories of claims of the study

The semantics of the membership of a judgments into a category is: the judgments contains a claim of that category, i.e. all the judgments into the ACPA category contain a claim related to Civil fine for abuse of process. Table 2 presents parts of a judgment of that category [ACPA]. The parts refer to the mentions of the claim and to corresponding decision respectively. Figure 1 presents additional details about the datasets.

Observation 1. *Decisions most often only contain a single claim of a specific category.*

On the one hand, the statistics on the labelled data show that the judgments contain for the most part a single claim of a category. The percentage of judgments having only one request of a category is respectively: 100% for ACPA, 63.33% for CONCDEL, 95.45% for DANAIS, 80.22% for DCPPC, and 76.21% for DORIS. However, we note the exception of the STYX category (damages on article 700 CPC), where in most of the judgments, there are rather 2 claims. This exception can be justified by the fact that each party generally makes this type of request because it relates to the reimbursement of legal costs.

On the other hand, it exists few judgements with two or more claims. In this case, the classification task of any claim becomes difficult since specific vocabulary and sentences may appear in the judgement related to other claims (although there are in the same category). This may be embodied by

	In French	In English
claim	À l'audience, la SA SFP reprenant oralement ses conclusions visées par le greffier, result à la cour de: - confirmer le jugement déféré - débouter M. S. de l'ensemble de ses demandes - le condamner à payer une amende civile de 1.500 pour procédure abusive en application de l'article 32-1 du code de procédure civile - le condamner à payer la somme...	At the hearing, SA SFP orally resuming its conclusions referred to in the clerk, requests the court to: - confirm the judgment referred - dismiss Mr S. of all his requests - order him to pay a civil fine of 1,500 for abusive procedure in application of article 32-1 of the code of civil procedure - order him to pay the sum ...
decision	PAR CES discussion LA COUR, CONFIRME le jugement déféré en toutes ses dispositions; Y ajoutant, DIT n'y avoir lieu à application des dispositions de l'article 700 du code de procédure civile; REJETTE le surplus des demandes ; CONDAMNE M Khellil S. aux dépens d'appel.	FOR THESE REASONS THE COURTYARD, CONFIRMS the judgment referred in all its provisions; Adding to it, SAID to take place there in application of the provisions of article 700 of the code of Civil Procedure; REJECTS excess requests; ORDERS M Khellil S. at costs of appeal.

Table 2. Extract from "Cour d'appel, Paris, Pôle 6, chambre 9, 18 Mai 2016 – n14/11380"

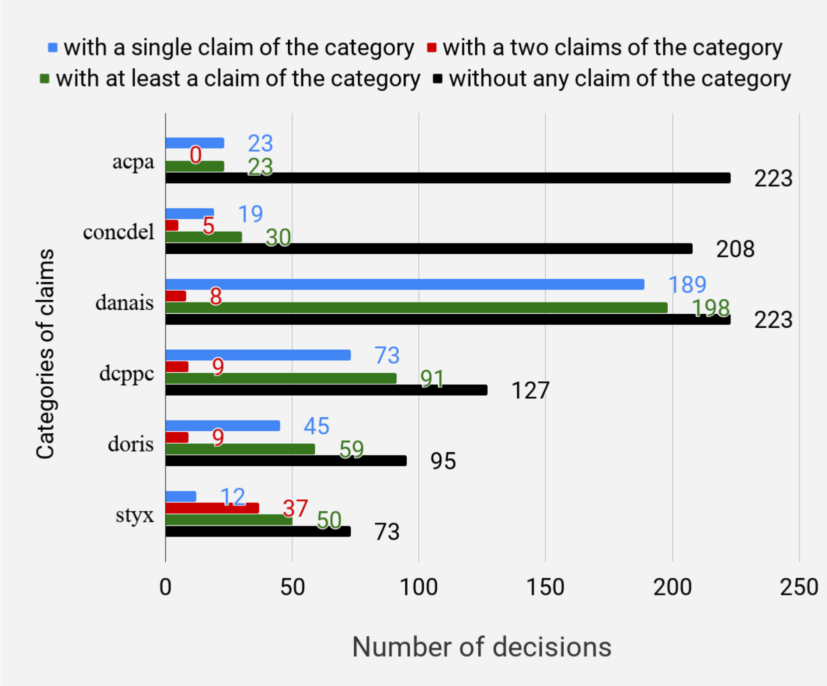

Figure 1. Number of claims in judgments

noise or outliers in the dataset of each claim category. The use of Gini estimators is therefore welcome to handle outlying observations.

Observation 2. *The judge's decisions are binary: accept or reject.*

Figure 2 highlights the fact that outcomes of a given claim are most often accepted or rejected, and that other forms of results are very rare.

Theses observations motivate the interest of developing binary classifier for predicting the outcome of a claim appertaining to a specific category.

Observation 3. *The algorithm must be able to deal with the important number of tokens of judgments.*

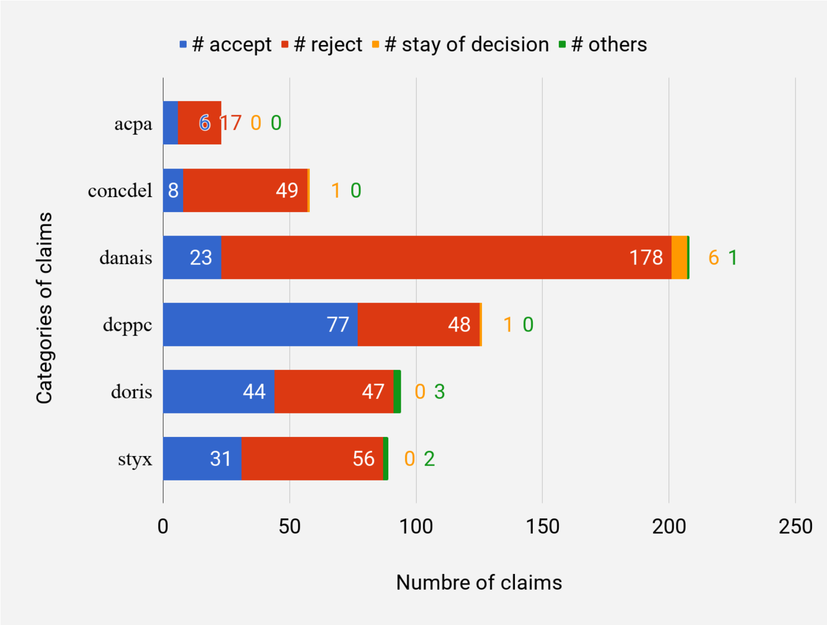

Figure 2. Distribution of judges' decisions within each category of claims

122 Figure 3 illustrates the distribution of the judgments lengths (number of tokens, i.e. words). 123 We note that the texts are long in comparison to those usually considered by state-of-the-art text 124 classification approaches. As we will discuss later, this particularity will hamper the use of some 125 efficient existing approaches such as PLS algorithms for compression.

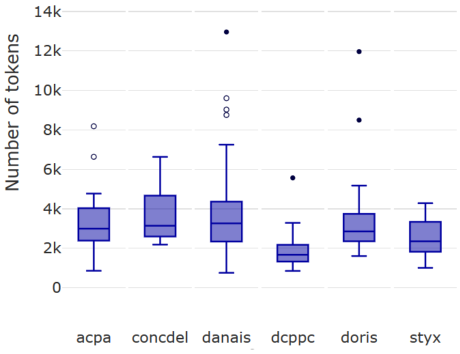

Figure 3. Distribution of the size of the decision by tokens

126 **Observation 4.** *In some claim categories, there may exist strong imbalance between the outcomes accept/reject.*

127 Table 3 presents the final statistics of the dataset used for both training and evaluating the 128 predictive models evaluated in this study. As can be seen, 4 claim categories out of 6 exhibit strong 129 unbalanced decisions.

Dataset	Accepted	Rejected	Total
ACPA	6 (26.09%)	17 (73.91%)	23
CONCDEL	4 (22.22%)	14 (77.78%)	18
DANAIS	21 (11.23%)	166 (88.77%)	187
DCPPC	48 (66.66%)	24 (33.33%)	72
DORIS	23 (52.27%)	21 (47.72%)	44
STYX	4 (33.33%)	8 (66.67%)	12

Table 3. Class distributions per claim category

3. Texts classification

Text classification allows judgments to be organized in predefined groups. This technique has received a large audience for a long time. Two technical choices mainly influence the performance of the classification: the representation of the texts and the choice of the classification algorithm. In the following, the predicted variable is denoted y, the predictors are denoted x, the learning base including the observations of the sample is expressed as $D = \{(x_i, y_i)_{i=1...N}\}$, and C represents claim categories.

Notation	Description
t	a term
d	a judgement (document)
$\lvert d \rvert$	size of d (number of tokens)
c	a label
$\bar{c}$	the other labels
D	the global set of documents ($N = \lvert D \rvert$)
D_c	the set of documents labeled with c
$D_{\bar{c}}$	the set of documents not labeled with c
N_t	the number of documents containing t
$N_{\bar{t}}$	the number of documents without t
$N_{t,c}$	the number of documents of c with t
$N_{\bar{t},c}$	the number of documents of c without t
$N_{t,\bar{c}}$	the number of documents of $\bar{c}$ with t
$N_{\bar{t},\bar{c}}$	the number of documents of $\bar{c}$ without t
$DF_{t\|c}$	proportion of documents of c with t ($DF_{t\|c} = \frac{N_{t,c}}{\lvert D_c \rvert}$)
$DF_{c\|t}$	proportion of documents of c in the global set of documents with t

Table 4. Notation used in formulas.

Considering a vocabulary $V = \{t_1, t_2, \ldots, t_n\}$, we further assume that every judgment $d \in D$ is represented as a TF-IDF vector embedding (*Term Frequency - Inverse Document Frequency*) [22] $\vec{d} \in \mathbb{R}^n$, where each dimension $1 \leq k \leq n$ refers to word $t_k \in V$ and $\vec{d}[k] = w(t_k, d)$ is the weight of t_k in d defined as the normalized product of a global weight $g(t_k)$ depending on the training corpus and a local weight $l(t_k, d)$ stressing the importance of t_k in judgment d:

$$w(t_k, d) = l(t_k, d) \times g(t_k) \times nf(d)$$

with nf a normalization factor. Table 4 summarizes the notations used in the paper. The global weight is computed following one of the methods presented in Table 5. The local weight is computed from the frequency of occurrences of the word in the judgment using one of the methods of Table 6.

The vector representation of texts generally results in high-dimensional vectors whose coordinates are mostly zero. Consequently, dimension reduction (compression) techniques, such as PLS regressions make it possible to obtain vectors more relevant to classification tasks.

4. Generalized Gini-PLS algorithms for text classification

The Gini-PLS regression has been introduced by [11]. In what follows we propose two Gini-PLS algorithms: a generalized Gini-PLS regression based on the Gini generalized covariance operator, and a combination of the latter to the logistic regression. We first review the PLS algorithm.

4.1. PLS

The advantage of the Gini-PLS algorithm is to reduce the sensitivity to outliers. It is an extension of the PLS analysis (*partial least square*) [24]. The PLS analysis explains the dependence between one or more predicted variables y and predictors $\mathbf{x} = (x_1, x_2, \cdots, x_m)$. It mainly consists in transforming the predictors into a reduced number of h orthogonal principal components $t_1, \cdots, t_h$. It is therefore a method of dimension reduction in the same way as the principal component analysis (PCA), the

Description	Formula
Inverse document frequency (IDF) [12]	$idf(t) = \log_2\left(\frac{N}{N_t}\right)$
Probabilistic IDF [13]	$pidf(t) = \log_2\left(\frac{N}{N_t} - 1\right)$
BM25 IDF [14]	$bidf(t) = \log_2\left(\frac{N_{\bar{t}}+0.5}{N_t+0.5}\right)$
Frequency difference	$\Delta_{DF}(t,c) = DF_{t\|c} - DF_{t\|\bar{c}}$
Information gain [15]	$ig(t,c) =$ $\frac{N_{t,c}}{N}\log_2\left(\frac{N_{t,c}N}{N_t}\right) + \frac{N_{\bar{t},c}}{N}\log_2\left(\frac{N_{\bar{t},c}N}{N_{\bar{t}}\|D_c\|}\right) + \frac{N_{t,\bar{c}}}{N}\log_2\left(\frac{N_{t,\bar{c}}N}{N_t\|D_{\bar{c}}\|}\right) + \frac{N_{\bar{t},\bar{c}}}{N}\log_2\left(\frac{N_{\bar{t},\bar{c}}N}{N_{\bar{t}}\|D_c\|}\right)$
Relevance frequency [16]	$rf(t,c) = \log\left(2 + \frac{N_{t,c}}{max(1,N_{t,\bar{c}})}\right)$
χ^2 coefficient [17]	$\chi^2(t,c) = \frac{N((N_{t,c}N_{\bar{t},\bar{c}})-(N_{t,\bar{c}}N_{\bar{t},c}))^2}{N_tN_{\bar{t}}\|D_c\|\|D_{\bar{c}}\|}$
Correlation coefficient [18]	$ngl(t,c) = \frac{\sqrt{N}(N_{t,c}N_{\bar{t},\bar{c}})-(N_{t,\bar{c}}N_{\bar{t},c})}{\sqrt{N_tN_{\bar{t}}\|D_c\|\|D_{\bar{c}}\|}}$
GSS coefficient [19]	$gss(t,c) = (N_{t,c}N_{\bar{t},\bar{c}}) - (N_{t,\bar{c}}N_{\bar{t},c})$
Marascuilo coefficient [20]	$mar(t,c) =$ $\frac{\left((N_{t,c} - N_tN_{t,c}/N)^2 + (N_{t,\bar{c}} - N_t\|D_{\bar{c}}\|/N)^2 + (N_{\bar{t},c} - \|D_c\|N_{\bar{t}}/N)^2 + (N_{\bar{t},} - N_{\bar{t}}\|D_{\bar{c}}\|/N)^2\right)}{N}.$
Smoothed IDF delta [21]	$dsidf(t,c) = \log_2\left(\frac{\|D_{\bar{c}}\|(N_{t,c}+0.5)}{\|D_c\|(N_{t,\bar{c}}+0.5)}\right)$
BM25 IDF delta [21]	$dbidf(t,c) = \log_2\left(\frac{(\|D_{\bar{c}}\|-N_{t,\bar{c}}+0.5)\|(N_{t,c}+0.5)}{(\|D_c\|-N_{t,c}+0.5)(N_{t,\bar{c}}+0.5)}\right)$

Table 5. Global weighting metrics

Description	Formula
Gross term statement [22]	$tf(t,d) =$ Number of occurrences of t in d
Presence of the word [22]	$tp(t,d) = \begin{cases} 1 & \text{if } tf(t,d) > 0 \\ 0 & \text{otherwise} \end{cases}$
Log Normalization	$logtf(t,d) = 1 + \log(tf(t,d))$
Increased and standardized frequency of the word [22]	$atf(t,d) = k + (1-k)\frac{tf(t,d)}{\max\limits_{t\in V} tf(t,d)}$
Normalization based on the average frequency of the word [23] (*avg* is the average)	$logave(t,d) = \frac{1+\log tf(t,d)}{1+\log avg_{t\in V} tf(t,d)}$

Table 6. Local weighting metrics

linear discriminant analysis (LDA), and the quadratic discriminant analysis (QDA). The components $t_1, \ldots, t_h$ are built in different steps by applying the PLS algorithm repeatedly. More precisely, at each iteration $i \in [1, h]$, the component t_i is calculated by the formula $t_i = \mathbf{x} \cdot \mathbf{w}_i$, and then the target y is regressed by OLS on $\mathbf{x}$. PLS analysis has several advantages [25] including the robustness to the high-dimensional problem[3] and the ability to eliminate the multicollinearity problem[4] [26]. These problems are likely to arise on small corpora of texts with a large number of words as in our case. The

[3] When the number of predictors is very large compared to the number of training examples ($N << m$).

[4] Multicollinearity is a problem that arises when certain forecast variables in the model measure the same phenomenon.

PLS method is extended and successfully applied for various regression problems [25] or classification of data in general [27–29], and of texts in particular [30].

4.2. The Gini covariance operator

Schechtman and Yitzhaki [31] have recently generalized the Gini covariance operator, *i.e.* co-Gini, in order to impose more or less weight at the tails of distributions. This Gini covariance operator is given by:

$$\mathrm{cog}(x_\ell, x_k) := \mathrm{cov}(x_\ell, F(x_k)) = \frac{1}{N}\sum_{d=1}^{N}(x_{d\ell} - \bar{x}_\ell)(F(x_{dk}) - \bar{F}_{x_k}), \tag{1}$$

where $F(x_k)$ is the cdf of variable x_k. Let us denote $r_k = (R_\downarrow(x_{1k}), \ldots, R_\downarrow(x_{Nk}))$ the vector decreasing rank of variable x_k, in other words, the vector which assigns the lowest rank (1) of the observation with the highest value x_{dk}, and so on:

$$R_\downarrow(x_{dk}) := \begin{cases} N + 1 - \#\{x \leq x_{dk}\} & \text{no similar observation} \\ N + 1 - \frac{\sum_{d=1}^{p} \#\{x \leq x_{dk}\}}{p} & \text{if } p \text{ similar observations } x_{dk}. \end{cases}$$

The generalized co-Gini operator is given by Schechtman and Yitzhaki [31]:

$$\mathrm{cog}_\nu(x_\ell, x_k) := -\nu \mathrm{cov}(x_\ell, r_k^{\nu-1});\ \nu > 1. \tag{2}$$

The role of the co-Gini operator can be explained as follows. When $\nu \to 1$, the variability of the variables is attenuated so that $\mathrm{cog}_\nu(x_k, x_\ell)$ tends to zero (even if the variables x_k and x_ℓ are strongly correlated). On the contrary, if $\nu \to \infty$ then $\mathrm{cog}_\nu(x_k, x_\ell)$ allows one to focus on the distribution tails x_ℓ. The use of the co-Gini operator attenuates the influence of outliers, because the rank vector acts as an instrument in the regression of y on $\mathbf{x}$ (regression by instrumental variables) [32].

Thus, by proposing a Gini-PLS regression based on the ν parameter, we can calibrate the coefficient ν of the co-Gini operator in order to dilute the influence of the outlying observations. This generalized Gini-PLS regression becomes a regularized Gini-PLS regression where the parameter ν plays the role of a regularization parameter.

4.3. Generalized Gini-PLS regressions

The first Gini-PLS algorithm was proposed by [11]. We describe below the new Gini-PLS algorithm based on the generalized co-Gini opetaror. The generalized Gini-PLS algorithm is depicted in Figure 4.

Step 1: A weight vector $\mathbf{w}_1$ is first built to improve the link (in the co-Gini sense) between the predicted variable y and the predictors $\mathbf{x}$:

$$\max \mathrm{cog}_\nu(y, \mathbf{x}\mathbf{w}_1)\ ,\ \text{s.t. } \|\mathbf{w}_1\|_1 = 1.$$

The solution of this program is:

$$\mathbf{w}_1 = \frac{\mathrm{cog}_\nu(y, \mathbf{x})}{\|\mathrm{cog}_\nu(y, \mathbf{x})\|_1}.$$

As in the standard PLS case, the target y is regressed by OLS on the first component $t_1 = \mathbf{x}\mathbf{w}_1$:

$$y = \hat{c}_1 t_1 + \hat{\varepsilon}_1\ .$$

Step 2: The rank vector of each regressor $R_\downarrow(x_k)$ is regressed by OLS on t_1 (with residuals $\hat{\mathbf{u}}_1$):

$$R_\downarrow(\mathbf{x}) = \hat{\beta} t_1 + \hat{\mathbf{u}}_1.$$

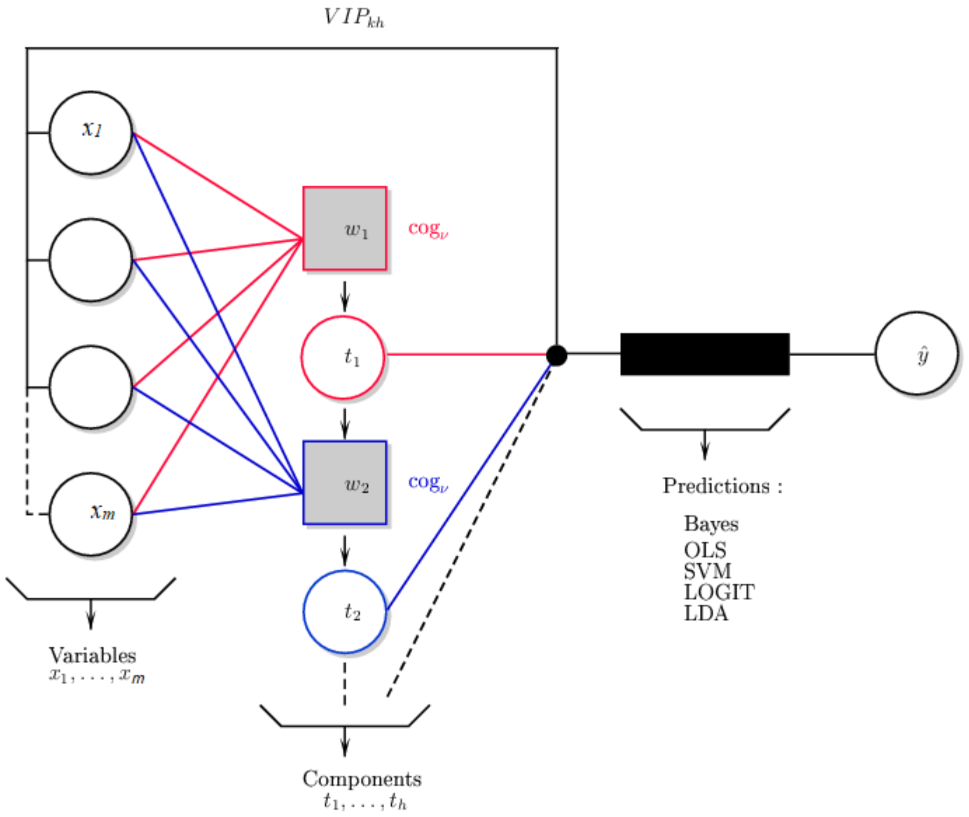

Figure 4. Generalized Gini-PLS algorithm

The second component t_2 is given by:

$$\max \mathrm{cog}_\nu(\hat{\varepsilon}_1, \hat{\mathbf{u}}_1\mathbf{w}_2) \text{ s.t. } \|\mathbf{w}_2\|_1 = 1 \implies \mathbf{w}_2 = \frac{\mathrm{cog}_\nu(\hat{\varepsilon}_1, \hat{\mathbf{u}}_1)}{\|\mathrm{cog}_\nu(\hat{\varepsilon}_1, \hat{\mathbf{u}}_1)\|_1} \implies t_2 = \hat{\mathbf{u}}_1\mathbf{w}_2.$$

Thereby, the components $t_1 \perp t_2$ allow a link to be established between y and $\mathbf{x}$ by OLS:

$$y = \hat{c}_1 t_1 + \hat{c}_2 t_2 + \hat{\varepsilon}_2 \ .$$

Step h: Partial regressions are run up to t_{h-1} :

$$R_\downarrow(\mathbf{x}) = \beta t_1 + \cdots + \gamma t_{h-1} + \hat{\mathbf{u}}_{h-1}.$$

Then, after maximisation:

$$\mathbf{w}_h = \frac{\mathrm{cog}_\nu(\hat{\varepsilon}_{h-1}, \hat{\mathbf{u}}_{h-1})}{\|\mathrm{cog}_\nu(\hat{\varepsilon}_{h-1}, \hat{\mathbf{u}}_{h-1})\|_1} \implies t_h = \hat{\mathbf{u}}_{h-1}\mathbf{w}_h,$$

we have by OLS,

$$y = \hat{c}_1 t_1 + \cdots + \hat{c}_h t_h + \varepsilon_h \ .$$

A cross validation makes it possible to find the optimal number of $h > 1$ components to retain. To test for a component t_h, we compute the model prediction with h components including document d, $\hat{y}_{h_d}$, and then without document d, $\hat{y}_{h_{(-d)}}$. The operation is iterated for all d varying from 1 to N: each time we remove the observation d and we re-estimate the model. To measure the significance of the model, we measure predicted residual sum of squared issued from the model with h components:

$$PRESS_h = \sum_{d=1}^{N} \left(y_d - \hat{y}_{h_{(-d)}} \right)^2 .$$

The sum of squared residuals of the model with $h-1$ components is:

$$RSS_{h-1} = \sum_{d=1}^{N} \left(y_d - \hat{y}_{(h-1)_d} \right)^2 .$$

The test statistics is:

$$Q_h^2 = 1 - \frac{PRESS_h}{RSS_{h-1}} .$$

The component t_h is retained in the analysis if $\sqrt{PRESS_h} \leq 0,95\sqrt{RSS_h}$. In other terms, if $Q_h^2 \geq 0,0975 = (1-0,95^2)$, t_h is significant in the sense that it improves the power of prediction of the model. In order to test for t_1, we use:

$$RSS_0 = \sum_{d=1}^{N} (y_d - \bar{y})^2 .$$

As in the standard PLS regression, the VIP_{hj} statistic is measured in order to select the word x_j which has the most significant impact on the decision y. The most significant words are those including $VIP_{hj} > 1$ with:

$$VIP_{hj} := \sqrt{\frac{m \sum_{\ell=1}^{h} Rd(y;t_\ell) w_{\ell j}^2}{Rd(y;t_1,\ldots,t_h)}}$$

and

$$Rd(y;t_1,\ldots,t_h) := \frac{1}{m} \sum_{\ell=1}^{h} \operatorname{cor}^2(y,t_\ell) =: \sum_{\ell=1}^{h} Rd(y;t_\ell).$$

with $\operatorname{cor}^2(y,t_\ell)$ is Pearson's correlation between y and component t_ℓ. This information is back propagated into the model (only once) in order to obtain the optimal number of components (on training data). The target variable y is then predicted as follows:

$$category(x) = \begin{cases} 0 & \text{if } \hat{y} < 0.5 \\ 1 & \text{otherwise.} \end{cases}$$

4.3.1. Generalized LOGIT-Gini-PLS

As can be seen in the generalized Gini-PLS algorithm, the weights $\mathbf{w}_j$ come from the generalized co-Gini operator applied to a Boolean variable $y \in \{0,1\}$. In order to find the weights $\mathbf{w}_j$ which maximize the link between the words x_j and the decision y, we propose to use the LOGIT regression, in other words, a sigmoid which is better adapted to Boolean variables. Thus, in each step of the Gini-PLS regression we replace the maximization of the co-Gini by measuring the following conditional probability:

$$P(y_d = 1/\mathbf{x} = \mathbf{x}_d) = \frac{\exp\{\mathbf{x}_d\beta\}}{1+\exp\{\mathbf{x}_d\beta\}} \tag{LOGIT}$$

where $\mathbf{x}_d$ is the d-th line of the matrix $\mathbf{x}$ of the predictors (being the words in judgment d). The estimation of the vector β is done by maximum likelihood. Therefore, at each step h of the PLS algorithm, the weights $\mathbf{w}_h$ are derived as follows:

$$\mathbf{w}_h = \frac{\beta}{\|\beta\|_2}$$

The generalized LOGIT-Gini-PLS algorithm is depicted in Algorithm 1.

Algorithm 1: Generalized LOGIT-Gini-PLS (training)

```
Data: $\mathbf{x}$ (predictors), $h_{max}$ (maximal number of components), $\nu_{max}$ (maximal value of $\nu$)
Result: Principal components $t_1, \dots, t_{h^*}$
repeat
    if $h == 1$: LOGIT equation $P(y/\mathbf{x}) \implies \mathbf{w}_1 = \frac{\beta}{\|\beta\|_2} \implies t_1 = \mathbf{x}\mathbf{w}_1$ ;
    repeat
        for $h > 1$ ;
        OLS equation: $R_{\downarrow}(\mathbf{x}) = \beta t_1 + \cdots + \beta t_{h-1} \implies \hat{\mathbf{u}}_{h-1}$ ;
        $\tilde{\mathbf{x}} := (\hat{\mathbf{u}}_{h-1} | t_1, \dots, t_{h-1}) \implies$ LOGIT equation $P(y/\tilde{\mathbf{x}}) \implies$ weights $\mathbf{w}_h = \frac{\beta}{\|\beta\|_2} \implies$
         $t_h = \hat{\mathbf{u}}_{h-1}\mathbf{w}_h$ ;
        OLS equation : $y = \sum_h c_h t_h + \varepsilon_h$ ;
    until $h = h_{max}$ $[h = h + 1]$;
    Compute $VIP_{kh}$, $Q_h^2$ ;
    Choose the optimal number of components $h^*$ ;
until $\nu = \nu_{max}$ $[\nu = \nu + 0.01]$;
Deduce the optimal parameter $\nu^*$ which minimizes the error ;
return $t_1, \dots, t_{h^*}, \nu^*$;
```

5. Experiments and results

We discuss the performance of various popular algorithms and the impact of data quantity and imbalance, heuristics, and explicit restriction of judgments to sections (regions) related to the claim category, as well as their ability to ignore other requests in the judgment. These experiments also aim to compare the effectiveness of Gini-LOGIT-PLS with other machine learning techniques. As in Im *et al.* [33], we compare different combinations of classification algorithms and term weighting methods (used for text representation). These combinations represent 600 experienced configurations including:[5]

- 12 algorithms of classification: Naive Bayes (NB), Support Vector Machine (SVM), K-nearest neighbors (KNN), Linear and quadratic discriminant analysis (LDA / QDA), Tree, fastText, Naive Bayes SVM (NBSVM), generalized Gini-PLS (Gini-PLS), Logit-PLS [34], generalized LOGIT-Gini-PLS (GiniLogitPLS), and the usual PLS algorithm (StandardPLS);
- 11 global weighting schemes (cf. Table 5): χ^2, $dbidf$, Δ_{DF}, $dsidf$, gss, idf, ig, mar, ngl, rf, avg_{global} (mean of the global metrics);
- 6 local weighting schemes (cf. Table 6): tf, tp, $logtf$, atf, $logave$, et avg_{local} (mean of the local metrics).

5.1. Assessment protocol

Two assessment metrics are used: precision and F_1-measure. To take into account the imbalance between the classes, the macro-average is preferred. It is the aggregation of the individual contribution of each class. It is calculated from the macro-averages of the precision (P_{macro}) and of the recall (R_{macro}),

[5] See https://github.com/tagnyngompe/taj-ginipls to enjoy the python code of the Gini-PLS algorithms.

which are calculated according to the average numbers of true positives ($\overline{TP}$) , false positives ($\overline{FP}$), and false negatives ($\overline{FN}$) as follows: [35]: $P_{macro} = \frac{\overline{TP}}{\overline{TP}+\overline{FP}}$, $R_{macro} = \frac{\overline{TP}}{\overline{TP}+\overline{FN}}$.

The efficiency of algorithms often depends on the meta-parameters for which optimal values must be determined. The *scikit-learn* [36] library implements two strategies for finding these values: RandomSearch and GridSearch. Despite the speed of the RandomSearch method, it is non-deterministic and the values it finds give a less accurate prediction than the default values. The same thing for the GridSearch method, which is very slow, and therefore impractical in view of the large number of configurations to be evaluated. Consequently, the values used for the experiments are the values defined by default (Table 7).

Algorithms	Hyperparameters
SVM	$C = 1.0; \gamma = \frac{1}{\lvert V\rvert \times var(X)}; noyau = RBF$
KNN	$k = 5$
LDA	$solver = svd, n\_components = 10$
QDA	
Tree	Gini criterion
NBSVM	n-grams of 1 to 3 words
Gini-PLS	$h_{max} = 10$
Logit-PLS	$h_{max} = 10$
Gini-Logit-PLS	$h_{max} = 10; \nu = 14$

Table 7. Values of the hyperparameters of the algorithms.

5.2. *Classification on the basis of the whole judgment*

By representing the entire judgment using various vector representations, the algorithms are compared with the representations that are optimal for them. We note from the results of Table 8 that the trees are on average better on all the categories even if on average the F_1 -measure is limited to 0.668. The results of PLS extensions are not very far from those of trees with differences of F_1-measure around 0.1 (if we choose the right representation scheme).

Representation	**Algorithm**	F_1	**min**	**Cat. min**	**max**	**Cat. max**	$Best(F_1)$ **-** F_1	**max - min**	**rang**
$tf - gss$	Tree	0.668	0.5	*doris*	0.92	*dcppc*	0	0.42	1
$tf - avg_{global}$	LogitPLS	0.648	0.518	*danais*	0.781	*dcppc*	0.02	0.263	13
$tf - avg_{global}$	StandardPLS	0.636	0.49	*danais*	0.836	*dcppc*	0.032	0.346	24
$tf - \Delta_{DF}$	GiniPLS	0.586	0.411	*danais*	0.837	*dcppc*	0.082	0.426	169
$tf - \Delta_{DF}$	GiniLogitPLS	0.578	0.225	*styx*	0.772	*dcppc*	0.09	0.547	220
-	NBSVM	0.494	0.4	*styx*	0.834	*dcppc*	0.174	0.434	
-	fastText	0.412	0.343	*doris*	0.47	*danais*	0.256	0.127	

Table 8. Comparison of word representation and algorithms to detect the the judicial outcome.

The F_1 average scores of the NBSVM and fastText algorithms generally do not exceed 0.5 despite being specially designed for texts. It can be noticed that they are very sensitive to the imbalance of data between the categories (more rejections than acceptances). Furthermore, it is more difficult to detect the acceptance of the requests. Indeed, these algorithms classify all the test data with the majority label (meaning) i.e. rejection, and therefore, they hardly detect some request acceptance. The case of the categories *doris* and *dcppc* for the NBSVM ($F_{1macro} = 0.834$) tends to demonstrate the strong sensitivity to negative cases of these algorithms since the F_1-measure of "reject" is always higher than that of "accept" (Table 9).

PLS algorithms systematically exceed the performance (F_1 -measurement) of fastText and NBSVM from 10 to 20 points. This tends to demonstrate the effectiveness of PLS techniques in their role of reduction of dimensions. Gini-PLS algorithms do not look any better operate than conventional PLS algorithms. Presumably the reduction of dimensions is done while still retaining too much noise in the

Cat.	Algo.	Prec.	Prec. equi.	err-0	err-1	$F_1(0)$	$F_1(1)$	F_{1macro}
dcppc	NBSVM	0.875	0.812	0	0.375	0.916	0.752	**0.834**
danais	fastText	0.888	0.5	0	1	0.941	0	0.47
danais	NBSVM	0.888	0.5	0	1	0.941	0	0.47
concdel	fastText	0.775	0.5	0	1	0.853	0	0.437
concdel	NBSVM	0.775	0.5	0	1	0.873	0	0.437
acpa	fastText	0.745	0.5	0	1	0.853	0	0.426
acpa	NBSVM	0.745	0.5	0	1	0.853	0	0.426
doris	NBSVM	0.5	0.492	0.167	0.85	0.63	0.174	0.402
dcppc	fastText	0.667	0.5	0	1	0.8	0	0.4
styx	fastText	0.667	0.5	0	1	0.8	0	0.4
styx	NBSVM	0.667	0.5	0	1	0.8	0	0.4
doris	fastText	0.523	0.5	0	1	0.686	0	0.343

0 = "reject" et 1 == "accept"
Cat.: Categories of claim
Algo. : algorithm
err-0: error rate of "reject"
err-1: error rate of "accept"
Prec.: global precision ($accuracy = \frac{TP}{N}$)
Prec. equi.: $\frac{1}{2}(accuracy(0) + accuracy(1))$

Table 9. Evaluation of fastText and NBSVM for detecting judicial outcomes for each claim category.

data. This is confirmed by the results of the trees which remain very mixed for which the F_1-measure (0.668) that exceeds barely that of Logit-PLS (0.648). It therefore seems necessary to proceed with zoning in the judgement that would better identify relevant information and thereby reduce the noise.

5.3. Classification based on sections of judgements including the vocabulary of the category

Since the judgements relate to several categories of claim, we experiment the restriction of the judgment to regions including vocabulary of the category of interest: request, result, previous result (result_a) stated under the terms of the category in the part of the judgment related to the context (Opinion section). The region-vector representation-algorithm combinations are compared in Table 10. The accuracy rate (F_1) increases significantly with the reduction of the judgement to regions, except for the category *doris*. The best restriction combines regions including the vocabulary of the category in the section Facts and Proceedings (request and previous result), in the Opinion section (context), and in the Holding section (result). After reducing the size of the judgment, the trees provide excellent results, followed very closely by our GiniPLS and LogitGiniPLS algorithms. For example, in the dcppc category (see Table 5), Tree performance (F_1 = 0.985) slightly exceed the LogitPLS (0.94) and standard PLS (0.934) algorithms. In the category concdel, Tree performance (F_1 = 0.798) is still closely followed by GiniLogitPLS (0.703) and standard PLS (0657) algorithms.

The most interesting case concerns neighborhood disturbances (doris category). These judgements often involve multiple information that is sometimes difficult to synthesize, even for humans. The argumentation exposed in *doris* is related to multiple information (problems of views, sunshine, trees, etc.) so that the factual elements that condition the identification of the judicial outcomes are sometimes complex. This information can be either under-represented or over-represented depending on the vectorization scheme. Our GiniPLS algorithm (like our GiniLogitPLS) seems to be particularly suitable for this category of request. The F_1-measures found in this category amount to 0.806 (for GiniPLS and GiniLogitPLS) and 0.772 for StandardPLS while the trees of decisions are not part of the relevant algorithms for this category of request (no allowed the best three algorithms). This result reinforces the idea that our GiniPLS algorithms can sometimes compete with decision trees that act as a benchmark in the literature. This result would make it possible in the future to consider including our GiniPLS algorithms in ensemble methods to broaden the spectrum of algorithms robust to outliers and which at the same time play a role of data compression.

Category	Region	Representation	Algorithm	F_1
acpa	**claim_result_a_result_context**	$tf - dbidf$	**Tree**	**0.846**
	facts and proceedings_opinion_holding	$tf - dbidf$	StandardPLS	0.697
	facts and proceedings_opinion_holding	$tf - avg_{global}$	LogitPLS	0.683
concdel	**facts and proceedings_opinion_holding**	$tf - gss$	**Tree**	**0.798**
	opinion	$tf - idf$	GiniLogitPLS	0.703
	context	$logave - dbidf$	StandardPLS	0.657
danais	**claim_result_a_result_context**	$avg_{local} - \chi^2$	**Tree**	**0.813**
	claim_result_a_result_context	$atf - avg_{global}$	LogitPLS	0.721
	claim_result_a_result_context	$atf - avg_{global}$	StandardPLS	0.695
dcppc	**claim_result_a_result_context**	$tf - \chi^2$	**Tree**	**0.985**
	claim_result_a_result_context	$tf - \chi^2$	LogitPLS	0.94
	facts and proceedings_opinion_holding	$tp - mar$	StandardPLS	0.934
doris	**facts and proceedings_opinion_holding**	$tp - dsidf$	**GiniPLS**	**0.806**
	facts and proceedings_opinion_holding	$tp - dsidf$	GiniLogitPLS	0.806
	facts and proceedings_opinion_holding	$atf - ig$	StandardPLS	0.772
styx	**opinion**	$tf - dsidf$	**Tree**	**1**
	demande_result_a_result_context	$logave - dsidf$	GiniLogitPLS	0.917
	facts and proceedings_opinion_holding	$tf - rf$	GiniPLS	0.833

Table 10. Accuracy of the classification with restriction of judgments to specif regions.

6. Conclusion

This article attempts to simplify the extraction of the meaning of the result rendered by the judges on a request for a given claim category. It consists in formulating the problem as a task of classifying judgments. Ten classification algorithms have been tested over 55 methods of vector embeddings. We have noticed that the classification results are mainly influenced by 3 characteristics of our data. First of all, the very small number of training examples disadvantages certain algorithms (sensitivity to outliers), such as fastText, which requires several thousand examples to update its parameters. Then, the strong imbalance between the classes ("accept" vs. "reject") makes it difficult to recognize the minority class which is generally the "accept" class. The strong gap between the errors on "reject" and those on "accept", as well as the good results obtained on *dcppc* constitute an evidence. Finally, the presence of other claim categories in the judgment degrades the efficiency of the classification because the algorithms do not manage alone to find the elements in direct relation with the chosen category. This is demonstrated by the positive impact of the restriction of the content to be classified in certain particular regions of the decision, even if the appropriate restriction depends on the category.

Finally, the decision trees are suitable for the classification task, but the use of Gini-PLS and Gini-Logit-PLS makes it possible to obtain performances fairly close to those of trees and sometimes higher. It would be interesting to combine these variants of PLS algorithms, with others ones such as Sparse-PLS which could perhaps help to solve the problem of vectors of zeros. There is also a large number of neural architectures for the classification of judgment and very large numbers of weighting metrics for the representation of texts, but none seems to fit all categories. Therefore, a study on the use of semantic embedding representations like Sent2Vec [37] or Doc2Vec [38] would be interesting.

Author Contributions: This article is drawn from a Chapter of Gildas Tagny-Ngompé's Ph.D. dissertation.

References

1. Chalkidis, I.e.a. A deep learning approach to contract element extraction. JURIX, Luxembourg, 2017, pp. 155–164.
2. WEI F., H. QIN, S.Y.; ZHAO, H. Empirical study of deep learning for text classification in legal document review. IEEE Int. Conf. Big Data, Seattle, 2018, pp. 3317–3320.

3. Luo Bingfeng, Yansong Feng, J.X.X.Z.D.Z. Learning to Predict Charges for Criminal Cases with Legal Basis. Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2017, pp. 2727–2736.
4. Legal judgment prediction via topological learning.
5. Automatic judgment prediction via legal reading comprehension.
6. RnRTD: Intelligent Approach Based on the Relationship-Driven Neural Network and Restricted Tensor Decomposition for Multiple Accusation Judgment.
7. Neural Legal Judgment Prediction in English.
8. Predicting the Outcome of Judicial Decisions made by the European Court of Human Rights.
9. Predicting Brazilian court decisions.
10. Tagny-Ngompé., G. *Méthodes d'analyse sémantique de corpus de décisions Jurisprudentielles, PhD dissertation, IMT Mines Alès*; 2020.
11. Mussard, S.; Souissi-Benrejab, F. Gini-PLS Regressions. *Journal of Quantitative Economics* **2018**, pp. 1–36. doi:{10.1007/s40953-018-0132-9}.
12. Sparck Jones, K. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* **1972**, *28*, 11–21.
13. Wu, H.; Salton, G. A comparison of search term weighting: term relevance vs. inverse document frequency. ACM SIGIR Forum. ACM, 1981, Vol. 16, pp. 30–39.
14. Jones, K.S.; Walker, S.; Robertson, S.E. A Probabilistic Model Of Information Retrieval: Development And Comparative Experiments. *Information Processing & Management* **2000**, *36*, 809–840.
15. Yang, Y.; Pedersen, J.O. A comparative study on feature selection in text categorization. ICML, 1997, Vol. 97, pp. 412–420.
16. Lan, M.; Tan, C.L.; Su, J.; Lu, Y. Supervised and traditional term weighting methods for automatic text categorization. *IEEE transactions on pattern analysis and machine intelligence* **2009**, *31*, 721–735.
17. Schütze, H.; Hull, D.A.; Pedersen, J.O. A comparison of classifiers and document representations for the routing problem. Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1995, pp. 229–237.
18. Ng, H.T.; Goh, W.B.; Low, K.L. Feature selection, perceptron learning, and a usability case study for text categorization. ACM SIGIR Forum. ACM, 1997, Vol. 31, pp. 67–73.
19. Galavotti, L.; Sebastiani, F.; Simi, M. Experiments on the use of feature selection and negative evidence in automated text categorization. International Conference on Theory and Practice of Digital Libraries. Springer, 2000, pp. 59–68.
20. Marascuilo, L.A. Large-sample multiple comparisons. *Psychological bulletin* **1966**, *65*, 280.
21. Paltoglou, G.; Thelwall, M. A study of information retrieval weighting schemes for sentiment analysis. Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, 2010, pp. 1386–1395.
22. Salton, G.; Buckley, C. Term-weighting Approaches In Automatic Text Retrieval. *Information Processing & Management* **1988**, *24*, 513–523.
23. Manning, C.D.; Raghavan, P.; Schütze, H. Scoring, term weighting and the vector space model. In *Introduction to information retrieval*; Cambridge university press: Cambridge, 2009; chapter 6, pp. 109–133.
24. Wold, H. Estimation of principal components and related models by iterative least squares. *Multivariate Analysis* **1966**, pp. 391–420.
25. Lacroux, A. Les avantages et les limites de la méthode «Partial Least Square »(PLS): une illustration empirique dans le domaine de la GRH. *Revue de gestion des ressources humaines* **2011**, *80*, 45–64. doi:10.3917/grhu.080.0045.
26. Kroll, C.N.; Song, P. Impact of multicollinearity on small sample hydrologic regression models. *Water Resources Research* **2013**, *49*, 3756–3769.
27. Liu, Y.; Rayens, W. PLS and dimension reduction for classification. *Computational Statistics* **2007**, *22*, 189–208.
28. Durif, G.; Modolo, L.; Michaelsson, J.; Mold, J.E.; Lambert-Lacroix, S.; Picard, F. High dimensional classification with combined adaptive sparse PLS and logistic regression. *Bioinformatics* **2017**, *34*, 485–493.
29. Bazzoli, C.; Lambert-Lacroix, S. Classification based on extensions of LS-PLS using logistic regression: application to clinical and multiple genomic data. *BMC bioinformatics* **2018**, *19*, 314.

30. Zeng, X.Q.; Wang, M.W.; Nie, J.Y. Text classification based on partial least square analysis. Proceedings of the 2007 ACM symposium on Applied computing. ACM, 2007, pp. 834–838.

31. Schechtman, E.; Yitzhaki, S. A family of correlation coefficients based on the extended Gini index. *The Journal of Economic Inequality* **2003**, *1*, 129–146.

32. Olkin, I.; Yitzhaki, S. Gini regression analysis. *International Statistical Review/Revue Internationale de Statistique* **1992**, pp. 185–196.

33. Im, C.J.; Mandl, T.; others. Text Classification for Patents: Experiments with Unigrams, Bigrams and Different Weighting Methods. *International Journal of Contents* **2017**, *13*.

34. Tenenhaus, M. La regression logistique PLS. In *Modèles statistiques pour données qualitatives*; Droesbeke, Jean-Jacques and Lejeune, Michel and Saporta, Gilbert., Ed.; Editions Technip, 2005; chapter 12, pp. 263–276.

35. Van Asch, V. Macro- and micro-averaged evaluation measures. Technical report, Computational Linguistics & Psycholinguistics (CLiPS), Belgium, 2013. https://pdfs.semanticscholar.org/1d10/6a2730801b6210a67f7622e4d192bb309303.pdf.

36. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; others. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.

37. Pagliardini, M.; Gupta, P.; Jaggi, M. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. NAACL 2018 - Conference of the North American Chapter of the Association for Computational Linguistics, 2018.

38. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. International conference on machine learning, 2014, pp. 1188–1196.

Sample Availability: Samples of the compounds ...... are available from the authors.