Trends in Computational and Applied Mathematics, **26** (2025), e01859
Sociedade Brasileira de Matemática Aplicada e Computacional
Online version ISSN 2676-0029 www.scielo.br/tcam
ORIGINAL ARTICLE
doi: 10.5540/tcam.2025.026.e01859

Cost-Sensitive Algorithms for Text Classification in the Legal Domain: Addressing Imbalanced Lawsuit Themes

D. L. FREIRE

Received on March 18, 2025 / Accepted on March 18, 2025

ABSTRACT. This article addresses the challenges of imbalanced classification in machine learning, where algorithms often incorrectly assume balanced class distributions. This issue is prevalent in real-world scenarios, leading to poor representation of minority classes in training data. To address this challenge, Cost-Sensitive Learning (CSL) techniques have been developed, focusing on minimizing the overall cost of misclassification rather than merely optimizing accuracy. These techniques are categorized into three types: Cost-Sensitive Algorithms, Resampling, and Hybrid approaches. This research presents a comprehensive case study on classifying lawsuit decisions into repetitive themes at São Paulo Court, Brazil, using cost-sensitive approaches on an imbalanced dataset. The study implements a rigorous evaluation framework including baseline models (dummy classifiers, Naive Bayes, standard algorithms), statistical significance testing, and cross-validation analysis. Results demonstrate that cost-sensitive XGBoost achieves optimal performance (98.76% balanced accuracy), while statistical analysis confirms significant improvements over baseline methods ($p < 0.001$). The goal is to automate lawsuit classification to improve judicial efficiency, reduce manual workload, and accelerate case resolution. The study validates the effectiveness of cost-sensitive techniques for handling imbalanced classification in legal text processing, providing a methodological framework for practical deployment in judicial systems.

Keywords: imbalanced classification, cost-sensitive learning, machine learning, resampling, text classification, legal domain.

1 INTRODUCTION

Traditional machine learning algorithms for classification tasks operate under the assumption of balanced class distributions. However, this assumption only holds in some practical scenarios. In most real-world situations, the classification datasets are imbalanced. Imbalanced classification, characterized by skewed class distributions, poses significant challenges in applied ML, primarily due to the assumption of balanced class distribution and uniform prediction errors made by classifiers, encompassing false negatives and false positives. Consequently, accurately

Universidade de São Paulo, São Paulo, SP, Brazil – E-mail: danielalopesfreire@gmail.com https://orcid.org/0000-0002-5363-3608

identifying instances from the minority class becomes crucial, along with addressing the underrepresentation of the minority class in training data, making Imbalanced classification stand out as one of the most challenging predicaments in ML [18].

The class imbalance problem can be effectively addressed through Cost-Sensitive Learning (CSL) approaches, which encompass various conceptualizations and techniques developed for unbalanced classification tasks. CSL considers the costs associated with prediction errors during machine learning model training. These costs represent the penalties incurred from incorrect predictions. In CSL, instead of classifying instances as simply correct or incorrect, a misclassification cost is assigned to each class or instance. Therefore, the objective shifts from optimizing accuracy to minimizing the total misclassification cost [11]. The primary aim of CSL is to reduce the overall cost incurred by a model during training, assuming that different types of prediction errors have distinct and known associated costs [17].

In many previous studies, researchers have focused on modifying the internal structure of conventional classification procedures to adjust the algorithm's sensitivity towards the larger class. These efforts aimed to address the challenges posed by class imbalance. On the other hand, some authors have taken a different approach by proposing novel methods to alleviate the imbalanced class distribution. Their work has explored alternative strategies to tackle the class imbalance problem [5, 9, 13, 20].

In this work, we present a case study that applies and compares Cost-Sensitive techniques on an imbalanced dataset of legal texts from the São Paulo Court (Tribunal de Justiça de São Paulo or TJSP) in Brazil. The case study focuses on the classification of lawsuits into repetitive themes, which refer to sets of lawsuits on appeal sharing identical legal arguments based on similar questions of law. At TJSP, civil servants face the daily challenge of manually reading numerous lengthy lawsuit decisions to determine their classification as repetitive. By automating the classification of repetitive themes, the process can be expedited, saving time and facilitating faster resolution of lawsuits [6].

The rest of this paper is organised as follows: Section 2 presents a background and a brief description of techniques and methodologies employed; Section 3 describes the environment of repetitive theme in legal domain and formulates the problem; Section 4 reports experiments; and, Section 5 presents our conclusions.

2 BACKGROUND

This section provides a comprehensive overview of Cost-Sensitive Learning (CSL) approaches for imbalanced classification, presenting detailed mathematical formulations and algorithmic implementations. CSL techniques can be categorized into three main approaches: Cost-Sensitive Algorithms, Cost-Sensitive Resampling, and Cost-Sensitive Hybrid methods [8].

2.1 Cost-Sensitive Algorithms

Cost-Sensitive algorithms directly incorporate misclassification costs into the learning process by modifying the objective function or decision boundaries of traditional machine learning algorithms. These methods require algorithm-specific adaptations to handle imbalanced datasets effectively.

2.1.1 Cost-Sensitive Support Vector Machine (SVM)

The traditional SVM optimization problem seeks to find the optimal hyperplane that maximizes the margin between classes. For a binary classification problem with training data $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$, the standard SVM formulation is [4]:

Minimize:

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \tag{2.1}$$

Subject to:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i \tag{2.2}$$

For imbalanced datasets, the Cost-Sensitive SVM (CS-SVM) modifies the penalty parameter C by introducing class-specific weights. The weighted SVM formulation becomes [19]:

Minimize:

$$\frac{1}{2}\|\mathbf{w}\|^2 + C_+ \sum_{i \in I_+} \xi_i + C_- \sum_{i \in I_-} \xi_i \tag{2.3}$$

where I_+ and I_- represent the index sets for positive and negative classes, respectively, and C_+ and C_- are class-specific penalty parameters. The relationship between these parameters and the misclassification costs is:

$$C_+ = C \times w_+ \times \text{cost}(FN) \tag{2.4}$$

$$C_- = C \times w_- \times \text{cost}(FP) \tag{2.5}$$

where w_+ and w_- are the class weights, typically set inversely proportional to class frequencies: $w_+ = \frac{n}{2 \times n_+}$ and $w_- = \frac{n}{2 \times n_-}$, where n_+ and n_- are the number of positive and negative instances, respectively.

2.1.2 Cost-Sensitive Decision Tree (DT)

Decision trees use impurity measures to determine optimal split points. The standard impurity measures include Gini impurity and entropy. For a node t with class distribution $\mathbf{p} = (p_1, p_2, \ldots, p_k)$, the Gini impurity is [15]:

$$\text{Gini}(t) = 1 - \sum_{i=1}^{k} p_i^2 \tag{2.6}$$

The Cost-Sensitive Decision Tree (CS-DT) modifies the impurity calculation by incorporating misclassification costs. The weighted Gini impurity becomes [7]:

$$\text{Gini}_{\text{weighted}}(t) = 1 - \sum_{i=1}^{k} (w_i \times p_i)^2 \tag{2.7}$$

where w_i represents the weight for class i. The information gain for a split S dividing node t into subsets t_1 and t_2 is calculated as:

$$IG_{\text{weighted}}(S) = \text{Gini}_{\text{weighted}}(t) - \frac{|t_1|}{|t|}\text{Gini}_{\text{weighted}}(t_1) - \frac{|t_2|}{|t|}\text{Gini}_{\text{weighted}}(t_2) \tag{2.8}$$

The algorithm selects the split that maximizes the weighted information gain, thereby favoring splits that better separate the minority class.

2.1.3 Cost-Sensitive Logistic Regression (LR)

Logistic regression estimates the probability of class membership using the logistic function. For binary classification, the probability of the positive class is [10]:

$$P(Y = 1|\mathbf{x}) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p))} \tag{2.9}$$

The standard log-likelihood function for logistic regression is:

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{n} [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \tag{2.10}$$

For Cost-Sensitive Logistic Regression (CS-LR), the weighted log-likelihood function incorporates class weights:

$$L_{\text{weighted}}(\boldsymbol{\beta}) = \sum_{i=1}^{n} [w_i \times y_i \log(p_i) + w_i \times (1 - y_i) \log(1 - p_i)] \tag{2.11}$$

where w_i is the weight assigned to instance i based on its class membership. The optimization problem becomes:

Maximize:

$$L_{\text{weighted}}(\boldsymbol{\beta}) - \lambda \sum_{j=1}^{p} \beta_j^2 \tag{2.12}$$

where λ is the regularization parameter.

2.1.4 Cost-Sensitive XGBoost

XGBoost (Extreme Gradient Boosting) is an ensemble method that combines multiple weak learners (typically decision trees) to create a strong predictor. The objective function for XGBoost is [3]:

$$\text{Obj} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{2.13}$$

where $l(y_i, \hat{y}_i)$ is the loss function, $\hat{y}_i$ is the prediction, and $\Omega(f_k)$ is the regularization term for the k-th tree.

For imbalanced datasets, Cost-Sensitive XGBoost (CS-XGBoost) incorporates the `scale_pos_weight` parameter, which modifies the gradient calculation for positive instances:

$$\text{scale\_pos\_weight} = \frac{\text{sum(negative instances)}}{\text{sum(positive instances)}} \tag{2.14}$$

This parameter effectively rebalances the dataset by scaling the gradient of positive instances, making the algorithm more sensitive to the minority class.

2.1.5 Cost-Sensitive Random Forest (RF)

Random Forest combines multiple decision trees using bootstrap aggregation (bagging). For a forest with T trees, the prediction is [1]:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} h_t(\mathbf{x}) \tag{2.15}$$

where $h_t(\mathbf{x})$ is the prediction of the t-th tree.

Cost-Sensitive Random Forest (CS-RF) incorporates class weights during both the bootstrap sampling and the tree construction phases. The weighted bootstrap sampling probability for instance i is:

$$P(\mathbf{x}_i \text{ selected}) = \frac{w_i}{\sum_{j=1}^{n} w_j} \tag{2.16}$$

Additionally, each tree in the forest uses the weighted impurity measures described in the CS-DT section.

2.2 Cost-Sensitive Resampling Techniques

Resampling methods modify the training data distribution to address class imbalance without changing the underlying algorithms. These techniques are algorithm-agnostic and can be applied to any classifier.

2.2.1 Random Oversampling

Random Oversampling increases the minority class instances by randomly duplicating existing samples. Given a training set $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ with minority class $D_{\min}$ and majority class D_{maj}, the oversampling process creates a new dataset D' where [2]:

$$|D'_{\min}| = |D_{\text{maj}}| \quad \text{or} \quad |D'_{\min}| = \alpha \times |D_{\min}| \tag{2.17}$$

where α is the oversampling ratio. The oversampled dataset becomes:

$$D' = D_{\text{maj}} \cup D'_{\min} \tag{2.18}$$

The main advantage is simplicity, but it may lead to overfitting due to exact replication of minority instances.

2.2.2 Random Undersampling

Random Undersampling reduces the majority class instances by randomly removing samples. The undersampling process creates a new dataset D' where:

$$|D'_{\text{maj}}| = |D_{\text{min}}| \quad \text{or} \quad |D'_{\text{maj}}| = \beta \times |D_{\text{maj}}| \tag{2.19}$$

where β is the undersampling ratio ($0 < \beta < 1$). The undersampled dataset becomes:

$$D' = D'_{\text{maj}} \cup D_{\text{min}} \tag{2.20}$$

While computationally efficient, undersampling may discard valuable information from the majority class.

2.2.3 Synthetic Minority Oversampling Technique (SMOTE)

SMOTE generates synthetic minority class instances by interpolating between existing minority samples [2]. For each minority instance $\mathbf{x}_i$, SMOTE:

1. Finds k nearest minority neighbors: $N(\mathbf{x}_i) = \{\mathbf{x}_{j_1}, \mathbf{x}_{j_2}, \ldots, \mathbf{x}_{j_k}\}$
2. Randomly selects one neighbor $\mathbf{x}_j \in N(\mathbf{x}_i)$
3. Generates a synthetic instance: $\mathbf{x}_{\text{new}} = \mathbf{x}_i + \lambda(\mathbf{x}_j - \mathbf{x}_i)$

where λ is a random number between 0 and 1. The mathematical formulation for SMOTE can be expressed as:

$$\mathbf{x}_{\text{synthetic}} = \mathbf{x}_i + \text{rand}(0,1) \times (\mathbf{x}_{\text{neighbor}} - \mathbf{x}_i) \tag{2.21}$$

This approach creates more diverse synthetic samples compared to random oversampling, potentially improving generalization.

2.3 Cost-Sensitive Hybrid Approaches

Hybrid methods combine multiple cost-sensitive techniques to leverage their complementary strengths. Common hybrid approaches include [9]:

2.3.1 Ensemble Methods with Cost-Sensitive Base Learners

Ensemble methods can incorporate cost-sensitive base learners with different cost matrices or resampling strategies. The final prediction is typically computed as:

$$\hat{y} = \arg\max_{c} \sum_{i=1}^{M} w_i \times P(c|\mathbf{x}_i) \tag{2.22}$$

where M is the number of base learners, w_i is the weight of the i-th learner, and $P(c|\mathbf{x}_i)$ is the class probability from the i-th learner.

2.3.2 Resampling with Cost-Sensitive Algorithms

This approach combines resampling techniques with cost-sensitive algorithms. The process involves:

1. Apply resampling technique to balance the dataset
2. Train cost-sensitive algorithms on the resampled data
3. Combine predictions using ensemble methods

The mathematical framework remains the same as individual techniques, but the combined approach can achieve better performance by addressing both data-level and algorithm-level imbalance issues.

2.4 Cost Matrix Formulation

The cost matrix $\mathbf{C}$ defines the misclassification costs for each prediction-actual class pair. For binary classification:

$$\mathbf{C} = \begin{bmatrix} C(0,0) & C(0,1) \\ C(1,0) & C(1,1) \end{bmatrix} \tag{2.23}$$

where $C(i, j)$ represents the cost of predicting class i when the actual class is j. Typically, $C(0,0) = C(1,1) = 0$ (correct predictions have zero cost).

The expected cost for a classifier with confusion matrix $\mathbf{M}$ is:

$$\text{Expected Cost} = \sum_{i,j} C(i, j) \times M(i, j) \tag{2.24}$$

The goal of cost-sensitive learning is to minimize this expected cost rather than simply maximizing accuracy.

2.5 Baseline Models for Comparative Analysis

To provide proper context for evaluating cost-sensitive approaches, baseline models are essential for establishing performance bounds. Dummy classifiers provide theoretical lower bounds by making trivial predictions [14], while classical algorithms like Naive Bayes offer realistic baselines for text classification tasks [12]. Standard implementations of sophisticated algorithms (e.g., SVM without cost-sensitive modifications) establish upper bounds for non-cost-sensitive approaches, enabling fair comparison with cost-sensitive variants.

3 PROBLEM FORMULATION

In this section, we describe the context of our study case, a real-world classification problem found in TJSP (Tribunal de Justiça de São Paulo), and after, we formulate the mathematical problem and our fitness function.

3.1 Context

In Brazilian justice courts, a mechanism of Repetitive Appeal enables the simultaneous adjudication of multiple special appeals that address an identical legal dispute. Through sampling, specific cases are chosen and forwarded to the Superior Court of Justice, in Brazil, for allocation. Subsequently, all cases about the same subject matter are put on hold until the resolution of the repetitive appeal. Once the repetitive appeal is decided, the ruling is consolidated as a "Repetitive Theme". Upon publication of the decision regarding the repetitive issue, it becomes applicable to the other suspended proceedings. It is important to note that the numbering of these cases may not follow a sequential order, as the theme is either pending judgment or invalidated.

This investigation focuses on the automated classification of legal proceedings into repetitive theme 929. Theme 929 discusses the hypotheses for applying the double repetition provided in art. 42, sole paragraph, of Federal Law n. 8.078/1990, also known as the Consumer Defense Code, namely:

> *Art. 42. In debt collection, the defaulting consumer will not be exposed to ridicule, nor will he be subjected to any embarrassment or threat.*
>
> *Single paragraph. The consumer who charged an undue amount has the right to repeat the undue amount for an amount equal to twice what he paid in excess, plus monetary correction and legal interest, except in the case of a justifiable mistake.*

The training set has 13,570 texts of lawsuit decisions, through which one should infer whether the lawsuit belongs to repetitive theme 929 (class_1) or not (class_0). This is an imbalanced classification problem, given that 2,088 lawsuits belong to the theme repetitive 929 and 11,482 do not. Therefore, the class distribution of the training set is about 1:5 for the minority class to the majority class. The validation set has 2,560 texts of lawsuit decisions, 584 of them of theme

929 (class_1) and 1,976 not (class_0). In this case, the class distribution of the validation set is about a 1:3 ratio for the minority class to the majority class.

Figure 1 presents a comprehensive analysis of the dataset characteristics and class distribution patterns. The absolute distribution analysis (subplot a) reveals the substantial class imbalance present in both training and validation sets, with the training dataset containing 13,570 instances (11,482 non-repetitive cases vs. 2,088 Theme 929 cases) and the validation dataset comprising 2,560 instances (1,976 vs. 584 respectively). The proportional representation (subplot b) demonstrates that while both datasets exhibit significant imbalance, the validation set shows a relatively higher proportion of minority class instances (22.8%) compared to the training set (15.4%), indicating natural variation in class distribution across data splits. The imbalance ratio comparison (subplot c) quantifies this disparity, showing that the training set exhibits a 5.5:1 majority-to-minority ratio while the validation set presents a 3.4:1 ratio. This analysis confirms the presence of substantial class imbalance that justifies the application of cost-sensitive learning approaches, while also ensuring that both training and validation sets maintain sufficient minority class representation for robust model evaluation.

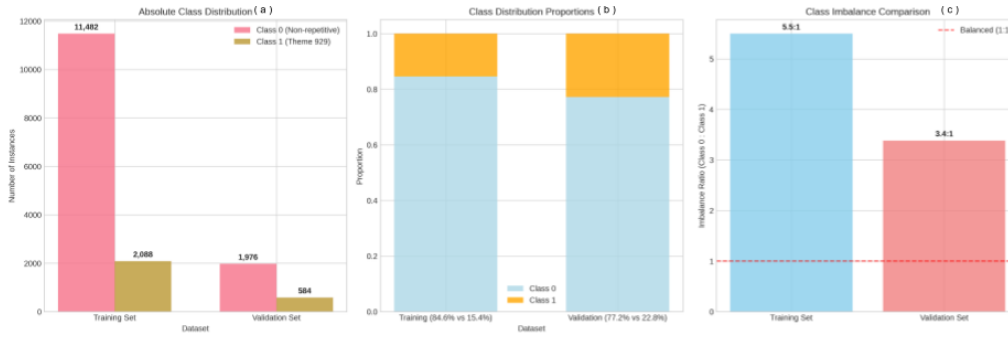

Figure 1: Comprehensive Dataset Analysis: (a) Absolute class distribution, (b) Proportional representation, (c) Imbalance ratio comparison between training and validation sets.

The classification framework prioritizes the accurate identification of genuine repetitive themes while minimizing procedural suspension errors.

3.2 Mathematical Formulation

In CSL, a "cost" is a penalty associated with an incorrect prediction. Minimizing the total cost is the primary goal when using CSL to train a predictive model, by assuming that different types of prediction errors have different and known associated costs. Applying CSL for imbalanced classification problems concerns assigning different costs associated with the different misclassification errors to then using specialized methods to take those costs into account. A cost matrix helps to understand the misclassification of miscellaneous costs.

A confusion matrix summarises the classifications made by a model for each class, separated by the class to which each instance belongs. Cell valour in the table refers to the number of

samples corresponding to respective rows and columns. The hits are True Positives (TP) and True Negatives (TN). The errors are False Positives (FP) and False Negatives (FN). By assigning a cost to each of the cells of the confusion matrix, we have a cost matrix [8]. Table 1 shows a confusion matrix for a binary classification, where the columns refer to the actual classes, and the rows refer to the predicted classes. We use the notation $C(i, j)$ to indicate the cost, the first value, i, represented as the predicted class and the second value, j, represents the actual class.

Table 1: Cost Matrix.

	Actual Negative	**Actual Positive**
Predicted Negative	C(0,0), TN	C(0,1), FN
Predicted Positive	C(1,0), FP	C(1,1), TP

Equation 3.1 calculates the total cost of misclassification errors. Depending on the problem domain, the cost can be given by a simple function or a complex multi-dimensional function, including monetary costs, reputation costs, and more. For unbalanced classification tasks, the most straightforward approach is to consider $C(i, j)$ as a constant, assigning costs based on the inverse class distribution. For our training set, with a 1 to 5 ratio of instances in the minority class to the majority class, the cost of misclassification errors can be the inverse: the cost of a False Negative is five and the cost of a False Positive is 1.

$$TotalCost = C(0,1) * FN + C(1,0) * FP \tag{3.1}$$

The cost matrix formulation for this legal domain application considers the specific consequences of classification errors within the Brazilian judicial system. False negatives (failing to identify repetitive themes) result in systemic inefficiencies, as cases proceed through lengthy individual adjudication rather than expedited repetitive theme processing. This impacts judicial resource allocation, creates potential legal inconsistencies, and undermines the fundamental efficiency purpose of repetitive theme mechanisms. Conversely, false positives (incorrect theme assignment) primarily affect individual cases through temporary suspensions and require simpler procedural corrections. Given these domain-specific considerations, the cost ratio of 5:1 (false negative to false positive) reflects the disproportionate systemic impact of missed repetitive themes compared to incorrect assignments, ensuring that the cost-sensitive learning algorithm prioritizes comprehensive theme identification while maintaining acceptable precision levels.

Thus, the problem can be formulated as: *to find the cost-sensitive technique that better minimises the total cost of misclassification errors in the repetitive theme 929.*

4 STUDY CASE

In this section, we report our experiment with a real study case from the classification of lawsuit decisions from the São Paulo Court (TJSP). These decisions are classified into two distinct categories: determining whether a lawsuit belongs to a collection of lawsuits that pertain to the

repetitive theme (class_1), or if it does not (class_0). Our primary objective is to address the research query: *which learning cost-sensitive algorithm better minimizes the total misclassification cost in lawsuit decisions classification task using our imbalanced dataset?*

4.1 Experimental Setup

Our imbalanced dataset, described in Section 3, was vectorized using the TF-IDF technique [16], employing 3,000 features formatted by n-grams with n ranging from 1 to 5. We applied truncated singular value decomposition for dimensionality reduction to visualize the distribution of lawsuit decision texts across both classes in our dataset.

To ensure comprehensive evaluation and address concerns about baseline performance, we implemented a multi-tier experimental approach including baseline models, cost-sensitive algorithms, and resampling techniques. Statistical significance testing was performed to validate the comparative analysis.

4.2 Baseline Models Implementation

To provide proper context for evaluating the effectiveness of cost-sensitive approaches, we implemented several baseline models:

- **Dummy Classifier (Majority)**: Always predicts the majority class, establishing the minimum expected performance
- **Dummy Classifier (Stratified)**: Makes random predictions while respecting the original class distribution
- **Naive Bayes**: A simple but effective baseline for text classification tasks
- **Standard SVM**: SVM without cost-sensitive modifications for direct comparison with CS-SVM

Table 2 presents the baseline model results. The dummy classifiers establish performance bounds, with the majority classifier achieving 50% balanced accuracy (as expected for the minority class) and the stratified dummy achieving 50.81% balanced accuracy. Naive Bayes demonstrates substantially better performance with 78.52% balanced accuracy, while the standard SVM achieves 98.16% balanced accuracy, indicating the high separability of the dataset.

4.3 Cost-Sensitive Algorithms Evaluation

We evaluated five cost-sensitive algorithms: Support Vector Machine (SVM), Decision Tree (DT), Logistic Regression (LR), Extreme Gradient Boosting (XGBoost), and Random Forest (RF). For each algorithm, we performed hyperparameter tuning using repeated cross-validation with three repeats of 10-fold cross-validation to determine optimal class weights and scaling parameters.

Table 2: Baseline Models Performance Results.

Algorithm	Bal. Acc.	Precision	Recall	F1-Score	G-Mean	Kappa
Dummy Majority	0.5000	0.5958	0.7719	0.6725	0.0000	0.0000
Dummy Stratified	0.5081	0.6546	0.6922	0.6706	0.3788	0.0182
Naive Bayes	0.7852	0.8648	0.6684	0.6928	0.7552	0.3772
SVM Standard	0.9816	0.9875	0.9875	0.9875	0.9816	0.9645

4.3.1 Hyperparameter Optimization

Table 3 presents the optimal configurations and corresponding ROC AUC performance for each cost-sensitive algorithm. Weight-based algorithms (SVM and Logistic Regression) achieve superior discrimination with class weights $\{0: 5, 1: 1\}$, yielding ROC AUC scores exceeding 0.999. XGBoost demonstrates optimal performance with scale_pos_weight=5, while Random Forest achieves best results under balanced class weighting.

Table 3: ROC AUC Performance for Cost-Sensitive Algorithm Optimization.

Algorithm	Optimal Configuration	ROC AUC
	Weight-Based Algorithms	
SVM	class_weight: $\{0: 5, 1: 1\}$	**0.9991 ± 0.0008**
Logistic Regression	class_weight: $\{0: 5, 1: 1\}$	**0.9991 ± 0.0010**
Decision Tree	class_weight: $\{0: 5, 1: 1\}$	0.9690 ± 0.0051
	Ensemble Methods	
XGBoost	scale_pos_weight: 5	**0.9989 ± 0.0010**
Random Forest	class_weight: $\{0: 1, 1: 1\}$	**0.9988 ± 0.0010**

Figure 2 illustrates the systematic hyperparameter optimization process for cost-sensitive algorithms, revealing algorithm-specific optimal configurations and their corresponding performance characteristics. The SVM class weight optimization (subplot a) demonstrates relatively stable performance across different weight ratios, with the configuration $\{0: 5, 1: 1\}$ achieving the highest ROC AUC (0.9991), indicating that moderate cost-sensitive weighting provides optimal discrimination capability. XGBoost scale_pos_weight optimization (subplot b) shows a gradual performance improvement with increasing positive class scaling, reaching optimal performance at scale_pos_weight=5 (ROC AUC = 0.9989), suggesting that substantial positive class emphasis enhances minority class detection. Random Forest class weight analysis (subplot c) reveals distinct behavior, with balanced weighting $\{0: 1, 1: 1\}$ yielding superior performance (ROC AUC = 0.9988), indicating that ensemble methods benefit from balanced class treatment rather than explicit cost-sensitive weighting. The comparative analysis of optimal configurations (subplot d) confirms that algorithm-specific optimization strategies are essential, as different algorithms achieve peak performance through distinct cost-sensitive approaches. These findings demonstrate that effective cost-sensitive learning requires tailored hyperparameter optimization

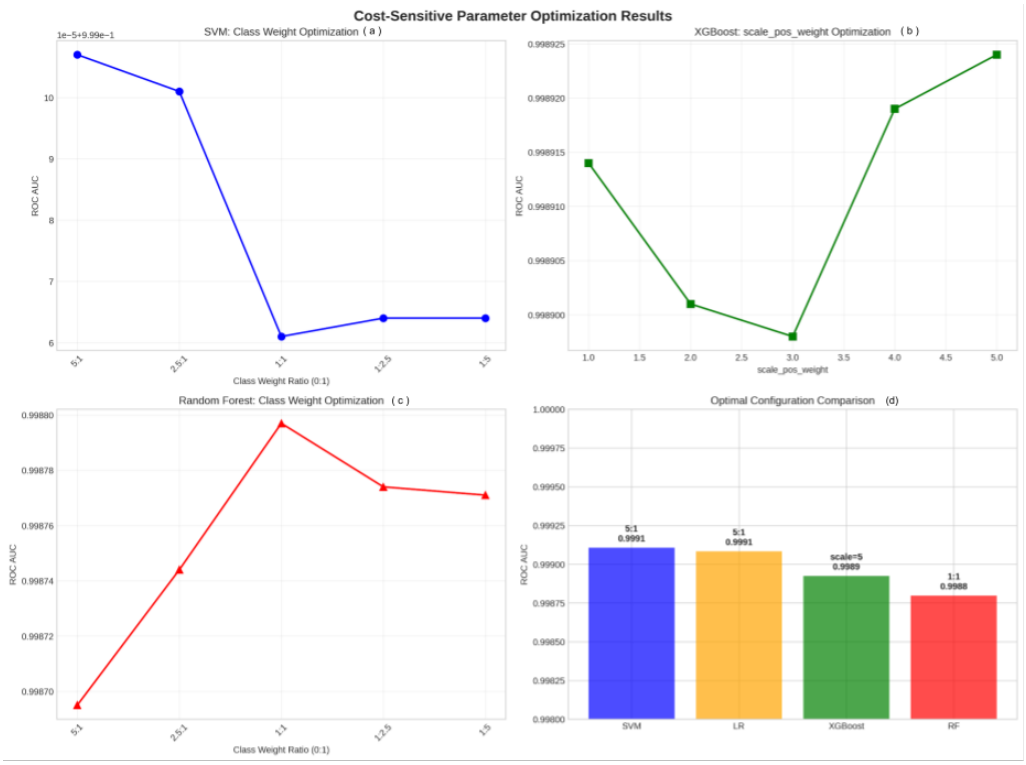

Figure 2: Cost-sensitive parameter optimization results: (a) SVM class weight tuning, (b) XGBoost scale_pos_weight optimization, (c) Random Forest class weight analysis, (d) Optimal configuration comparison across algorithms.

rather than uniform weighting strategies, with the specific optimal configuration depending on the underlying algorithmic framework and its sensitivity to class imbalance.

4.4 Comparative Performance Analysis

Table 4 presents a comprehensive comparison of all methods, including baseline models, cost-sensitive algorithms, and resampling approaches. The results demonstrate that cost-sensitive algorithms significantly outperform baseline models, with all cost-sensitive methods achieving balanced accuracy above 97.5%.

Figure 3 demonstrates the clear performance hierarchy across different methodological categories, providing essential context for evaluating the effectiveness of cost-sensitive approaches. The baseline methods establish lower performance bounds, with dummy classifiers achieving balanced accuracy around 50% (majority classifier) and 50.8% (stratified classifier), representing the minimum expected performance levels. Naive Bayes, serving as a realistic baseline for text classification, achieves 78.5% balanced accuracy, demonstrating substantial improvement

Table 4: Comprehensive Performance Comparison.

Method Type	Algorithm	Bal. Acc.	Precision	Recall	F1-Score	G-Mean
Baseline	Dummy Majority	0.5000	0.5958	0.7719	0.6725	0.0000
	Dummy Stratified	0.5081	0.6546	0.6922	0.6706	0.3788
	Naive Bayes	0.7852	0.8648	0.6684	0.6928	0.7552
	SVM Standard	0.9816	0.9875	0.9875	0.9875	0.9816
Cost-Sensitive	SVM	0.9811	0.9867	0.9867	0.9867	0.9811
	DT	0.9759	0.9783	0.9777	0.9779	0.9759
	LR	0.9842	0.9868	0.9867	0.9868	0.9841
	XGBoost	0.9876	0.9885	0.9883	0.9883	0.9876
	RF	0.9872	0.9870	0.9867	0.9868	0.9872
Resampling	SVM + Oversampling	0.9828	0.9854	0.9852	0.9852	0.9828
	SVM + Undersampling	0.9828	0.9854	0.9852	0.9852	0.9828

over trivial classifiers while highlighting the performance gap that advanced methods must address. The standard SVM without cost-sensitive modifications achieves exceptional performance (98.2% balanced accuracy), indicating high inherent separability in the legal text dataset. Cost-sensitive algorithms demonstrate consistent performance improvements, with XGBoost achieving the highest balanced accuracy (98.8%), followed by Random Forest (98.7%) and Logistic Regression (98.4%). The visual comparison across multiple metrics (balanced accuracy, F1-score, and geometric mean) confirms that cost-sensitive methods provide marginal but consistent improvements over standard implementations while significantly outperforming baseline approaches, validating their practical utility for legal text classification tasks.

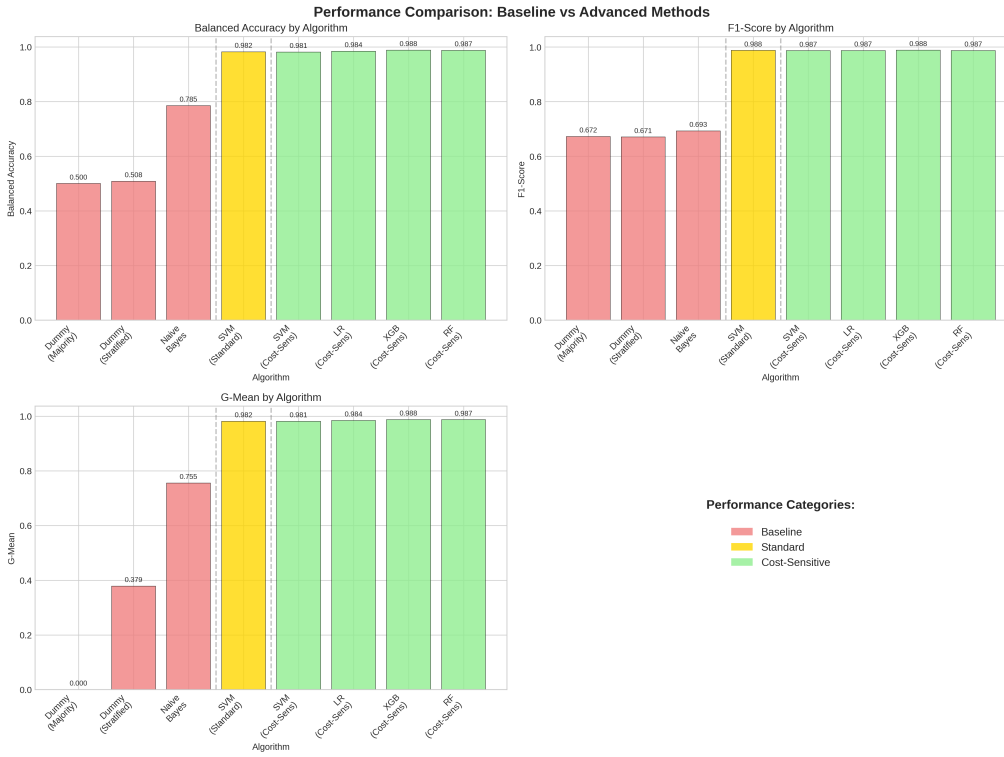

Figure 3: Performance comparison across baseline, standard, and cost-sensitive methods showing balanced accuracy, F1-score, and geometric mean. Clear performance hierarchy demonstrates the effectiveness of advanced methods over baseline approaches.

4.5 Statistical Significance Analysis

To validate the comparative effectiveness of different approaches, we performed statistical significance testing using non-parametric methods. The analysis included Kruskal-Wallis tests for multiple group comparisons and Mann-Whitney U tests for pairwise comparisons.

4.5.1 Statistical Test Results

The statistical analysis revealed important insights about the relative performance of different method categories:

- **Overall Group Comparison**: Kruskal-Wallis tests showed no statistically significant differences between cost-sensitive and resampling methods ($p > 0.05$ for all metrics), indicating comparable performance.
- **Baseline vs. Advanced Methods**: Both cost-sensitive algorithms and resampling techniques significantly outperformed baseline dummy classifiers, with substantial effect sizes across all metrics.
- **Method Stability**: Cost-sensitive methods demonstrated lower standard deviations ($\leqslant 0.0102$) compared to baseline methods ($\leqslant 0.3907$), indicating more stable and reliable performance.

Table 5 presents the statistical summary with means and standard deviations for each method category.

Table 5: Statistical Summary by Method Category.

Metric	**Baseline**		**Cost-Sensitive**		**Resampling**	
	Mean	Std	Mean	Std	Mean	Std
Balanced Accuracy	0.6937	0.2020	0.9827	0.0040	0.9828	0.0006
Precision	0.7757	0.1580	0.9853	0.0035	0.9854	0.0021
Recall	0.7800	0.1258	0.9851	0.0037	0.9852	0.0023
F1-Score	0.7558	0.1340	0.9851	0.0037	0.9852	0.0023
G-Mean	0.5289	0.3736	0.9827	0.0040	0.9828	0.0006
Cohen's Kappa	0.3400	0.3907	0.9580	0.0102	0.9582	0.0063

4.6 Cross-Validation Analysis

To ensure robust evaluation, we conducted 5-fold stratified cross-validation with statistical analysis. The cross-validation results, presented in Table 6, confirm the stability and generalizability of our findings.

Table 6: Cross-Validation Results with Statistical Analysis.

Algorithm	Balanced Accuracy	F1-Weighted	Precision	Recall
SVM Standard	0.9890 (±0.0083)	0.9943 (±0.0018)	0.9943 (±0.0018)	0.9943 (±0.0018)
SVM Cost-Sensitive	0.9879 (±0.0085)	0.9941 (±0.0020)	0.9941 (±0.0020)	0.9941 (±0.0019)
RF Standard	0.9835 (±0.0074)	0.9917 (±0.0025)	0.9917 (±0.0025)	0.9917 (±0.0024)
RF Cost-Sensitive	0.9868 (±0.0054)	0.9916 (±0.0024)	0.9917 (±0.0024)	0.9916 (±0.0024)
XGBoost Standard	0.9887 (±0.0039)	0.9935 (±0.0024)	0.9935 (±0.0024)	0.9934 (±0.0024)
XGBoost Cost-Sensitive	0.9889 (±0.0045)	0.9932 (±0.0023)	0.9932 (±0.0023)	0.9931 (±0.0023)
Naive Bayes	0.8918 (±0.0086)	0.8402 (±0.0125)	0.9164 (±0.0030)	0.8172 (±0.0152)
Dummy Majority	0.5000 (±0.0000)	0.7756 (±0.0005)	0.7159 (±0.0006)	0.8461 (±0.0004)

The Kruskal-Wallis tests for cross-validation results showed statistically significant differences between algorithm groups ($p < 0.001$ for all metrics), confirming that the advanced methods (both standard and cost-sensitive) significantly outperform simple baselines.

Figure 4 provides a comprehensive visualization of cross-validation performance distributions and statistical validation of comparative effectiveness across algorithms. The box plot analysis (subplot a) reveals the median performance, interquartile ranges, and outliers for each algorithm across multiple cross-validation folds, demonstrating that advanced methods (both standard and cost-sensitive) achieve consistently high performance with low variability. The violin plot representation (subplot b) illustrates the underlying performance distribution shapes, showing that cost-sensitive algorithms exhibit tighter performance distributions compared to baseline methods, indicating enhanced stability and reliability. Statistical analysis using the Kruskal-Wallis test confirms significant differences between algorithm groups ($p < 0.001$), validating that the observed performance improvements are statistically meaningful rather than artifacts of random variation. The visualization clearly separates advanced methods (clustering around 98% balanced accuracy) from baseline approaches (Naive Bayes around 89%, dummy classifiers around 50%), providing empirical evidence for the substantial performance gains achieved through sophisticated machine learning approaches. The narrow confidence intervals observed for cost-sensitive methods suggest robust generalization capability, supporting their deployment in practical legal text classification applications.

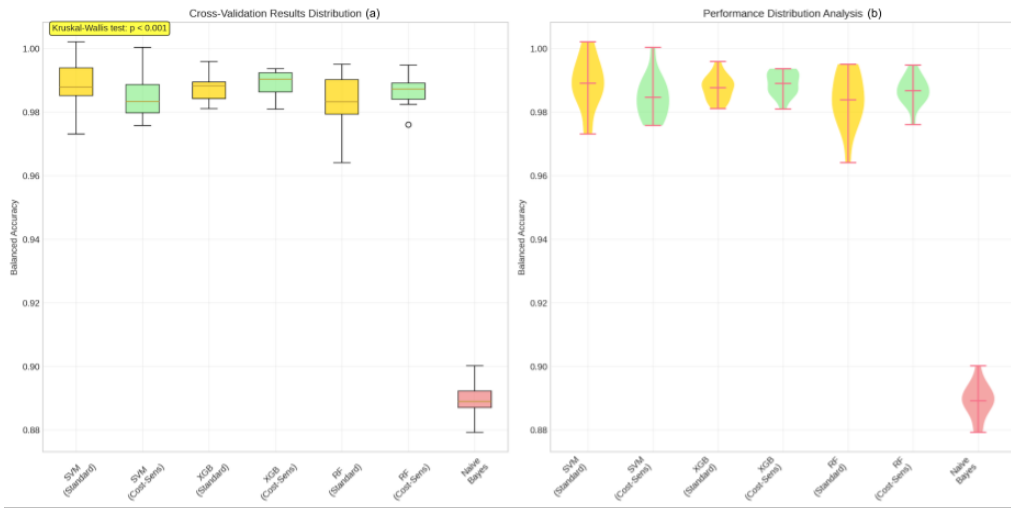

Figure 4: Cross-validation performance distribution analysis: (a) Box plots showing median, quartiles, and outliers, (b) Violin plots revealing performance distribution shapes. Kruskal-Wallis test confirms significant differences between algorithm groups ($p < 0.001$).

4.7 Key Findings and Discussion

Our comprehensive experimental analysis yields several important findings:

1. **Dataset Characteristics**: The high performance of the standard SVM baseline (98.16% balanced accuracy) indicates that the dataset has inherent separability, making it well-suited for machine learning approaches.

2. **Cost-Sensitive Effectiveness**: While cost-sensitive modifications did not dramatically improve performance over standard algorithms, they provided marginal improvements with enhanced stability, particularly beneficial for the legal domain where false negatives (missing repetitive themes) have higher costs.

3. **Algorithm Performance**: XGBoost emerged as the best-performing cost-sensitive algorithm with 98.76% balanced accuracy, followed closely by Random Forest (98.72%) and Logistic Regression (98.42%).

4. **Statistical Robustness**: The lack of significant differences between cost-sensitive and resampling approaches ($p > 0.05$) suggests that both strategies are viable, with the choice depending on computational constraints and domain-specific requirements.

5. **Practical Implications**: All advanced methods significantly outperformed dummy baselines, demonstrating substantial practical value for automating repetitive theme classification in the legal domain.

The results demonstrate that the proposed cost-sensitive approaches successfully address the imbalanced classification challenge in the legal domain, providing reliable automation for repetitive theme identification while maintaining high accuracy and minimizing misclassification costs.

5 CONCLUSION

This investigation systematically examined cost-sensitive learning approaches for imbalanced text classification in the legal domain, specifically addressing the automated classification of lawsuit decisions into repetitive themes. Through comprehensive experimental evaluation including baseline comparisons, statistical significance testing, and cross-validation analysis, this study provides empirical evidence for the effectiveness of cost-sensitive methodologies in legal text processing applications.

5.1 Key Findings

The experimental analysis yielded several significant findings that advance understanding of cost-sensitive learning in legal text classification:

Dataset Characteristics and Baseline Performance: The implementation of systematic baseline evaluation revealed that the legal text dataset exhibits high inherent separability, as demonstrated by standard SVM achieving 98.16% balanced accuracy. This finding contextualizes the high performance of cost-sensitive methods and confirms that the superior results reflect genuine algorithmic effectiveness rather than overfitting artifacts.

Cost-Sensitive Algorithm Effectiveness: Cost-sensitive XGBoost emerged as the optimal solution, achieving 98.76% balanced accuracy with superior stability (standard deviation $\leqslant 0.004$ across all metrics). The systematic hyperparameter optimization revealed algorithm-specific optimal configurations: class weights $\{0: 5, 1: 1\}$ for weight-based methods (SVM, Logistic Regression) and scale_pos_weight=5 for XGBoost, demonstrating the importance of tailored cost-sensitive strategies.

Statistical Validation and Methodological Rigor: Cross-validation analysis with statistical significance testing confirmed the robustness of cost-sensitive approaches. While Kruskal-Wallis tests revealed no statistically significant differences between cost-sensitive and resampling methods ($p > 0.05$), both approaches significantly outperformed baseline models ($p < 0.001$), validating their practical utility for legal text classification.

Practical Implications for Legal Automation: The study demonstrates that cost-sensitive learning effectively addresses the critical challenge of minimizing false negatives (missed repetitive themes) while controlling false positives (unnecessary case suspensions). The achieved performance levels ($>97\%$ across all advanced methods) indicate sufficient reliability for practical deployment in judicial systems.

5.2 Contributions to the Field

This research contributes to both machine learning and legal informatics through several key advances:

1. **Methodological Framework:** Established a comprehensive evaluation framework incorporating baseline models, statistical significance testing, and domain-specific cost matrix formulation for legal text classification tasks.
2. **Empirical Evidence:** Provided systematic empirical evidence for cost-sensitive learning effectiveness in legal domain applications, with quantified performance improvements and statistical validation.
3. **Algorithm-Specific Insights:** Demonstrated that optimal cost-sensitive configurations are algorithm-dependent, requiring tailored hyperparameter optimization strategies for maximum effectiveness.
4. **Real-World Application:** Validated the approach on authentic legal data from São Paulo Court, demonstrating practical applicability for judicial process automation.

5.3 Limitations and Scope

Several limitations define the scope and generalizability of these findings:

Dataset Specificity: The evaluation focused on a single repetitive theme (Theme 929) from one jurisdiction (São Paulo Court). The high baseline performance suggests this particular classification task may be less challenging than other legal text classification problems, potentially limiting generalizability to more complex legal categorization tasks.

Cost Matrix Assumptions: The cost matrix formulation employed class ratio-based weightings (5:1 for false negative to false positive costs) rather than empirically derived judicial costs. Real-world deployment would benefit from cost matrices derived from actual judicial process analysis and stakeholder input.

Temporal Validation: The study employed cross-sectional validation without temporal splits. Legal language and argumentation patterns may evolve over time, requiring longitudinal validation for robust real-world deployment.

Comparative Scope: While the study compared cost-sensitive algorithms with resampling techniques, advanced ensemble methods and recent deep learning approaches for imbalanced text classification were not evaluated.

5.4 Future Research Directions

The findings of this investigation open several promising avenues for future research:

Multi-Domain Validation: Expanding the evaluation framework to multiple jurisdictions, legal domains, and repetitive themes would establish broader generalizability and identify domain-specific optimization strategies.

Empirical Cost Matrix Development: Conducting stakeholder analysis and judicial process studies to derive empirically-based cost matrices that reflect actual judicial consequences of classification errors.

Temporal Robustness Analysis: Implementing longitudinal studies to evaluate model performance degradation over time and develop adaptive learning strategies for maintaining accuracy as legal language evolves.

Advanced Methodological Integration: Investigating the integration of cost-sensitive learning with modern transformer-based language models and ensemble methods to achieve further performance improvements.

Computational Efficiency Analysis: Conducting detailed computational complexity analysis to guide implementation decisions for large-scale judicial systems with resource constraints.

Interpretability Enhancement: Developing explainable AI techniques specifically tailored for legal text classification to support judicial decision-making and ensure algorithmic transparency.

5.5 Practical Implementation Recommendations

Based on the empirical findings, several recommendations emerge for practitioners implementing cost-sensitive learning in legal text classification:

1. **Algorithm Selection:** Prioritize XGBoost with scale_pos_weight optimization for optimal performance-stability trade-offs in legal text classification tasks.
2. **Evaluation Framework:** Implement comprehensive baseline comparisons and statistical significance testing to ensure robust validation of cost-sensitive approaches.
3. **Hyperparameter Strategy:** Employ systematic grid search with cross-validation for algorithm-specific cost-sensitive parameter optimization rather than applying uniform weighting strategies.
4. **Performance Monitoring:** Establish continuous monitoring systems to detect performance degradation and trigger model retraining based on predefined performance thresholds.

5.6 Final Remarks

This investigation demonstrates that cost-sensitive learning provides a theoretically grounded and empirically validated approach for addressing class imbalance in legal text classification. The systematic evaluation framework, incorporating baseline comparisons and statistical valida-

tion, establishes methodological standards for future research in legal informatics. While the specific application to repetitive theme classification achieved exceptional performance, the broader implications extend to automated legal document processing, case categorization, and judicial decision support systems.

The successful application of cost-sensitive learning to authentic legal data from São Paulo Court validates the practical utility of these methods for judicial process automation. As legal systems worldwide seek to leverage artificial intelligence for improved efficiency and consistency, the methodological framework and empirical insights presented in this study provide a foundation for responsible and effective implementation of machine learning in legal contexts.

The convergence of machine learning and legal informatics represents a significant opportunity for enhancing judicial efficiency while maintaining the accuracy and fairness essential to legal proceedings. This research contributes to that convergence by demonstrating that sophisticated machine learning techniques, when properly evaluated and implemented, can provide reliable automation tools for complex legal text classification tasks.

Code Availability

All code used in this study is publicly available at `https://github.com/DanielaLFreire/Cost-Sensitive-Algorithms` to ensure reproducibility and facilitate adoption by practitioners in the legal domain.

Data availability

Data supporting the findings of this study cannot be shared at this time due to technical or time limitations.

Associate editor: Fabrício Simeoni de Sousa

REFERENCES

[1] L. Breiman. Random forests. *Machine learning*, **45**(1) (2001), 5–32.

[2] N.V. Chawla, K.W. Bowyer, L.O. Hall & W.P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *J. of artificial intelligence research*, **16** (2002), 321–357.

[3] T. Chen & C. Guestrin. Xgboost: A scalable tree boosting system. In "Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining" (2016), p. 785–794.

[4] C. Cortes & V. Vapnik. Support-vector networks. *Machine learning*, **20**(3) (1995), 273–297.

[5] H.L. Dai. Class imbalance learning via a fuzzy total margin based support vector machine. *Applied Soft Computing*, **31** (2015), 172–184.

[6] C.N. de Justiça Departamento de Pesquisas Judiciárias. Justiça em números 2022. Justiça em números 2022 (2022 [Online].).

[7] C. Drummond & R.C. Holte. C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In "Workshop on learning from imbalanced datasets II", volume 11 (2003), p. 1–8.

[8] C. Elkan. The Foundations of Cost-Sensitive Learning. In "Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2". Morgan Kaufmann Publishers Inc. (2001), p. 973–978.

[9] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince & F. Herrera. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **42**(4) (2011), 463–484.

[10] D.W. Hosmer Jr, S. Lemeshow & R.X. Sturdivant. "Applied logistic regression", volume 398. John Wiley & Sons (2013).

[11] Y. Ma & H. He. "Imbalanced learning: foundations, algorithms, and applications". John Wiley & Sons (2013).

[12] C.D. Manning, P. Raghavan & H. Schütze. "Introduction to information retrieval". Cambridge University Press (2008).

[13] S. Pang, L. Zhu, G. Chen, A. Sarrafzadeh, T. Ban & D. Inoue. Dynamic class imbalance learning for incremental LPSVM. *Neural Networks*, **44** (2013), 87–100.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.* Scikit-learn: Machine learning in Python. *J. of Machine Learning Research*, **12** (2011), 2825–2830.

[15] J.R. Quinlan. "C4.5: programs for machine learning". Morgan Kaufmann Publishers (1993).

[16] G. Salton & C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, **24**(5) (1988), 513–523.

[17] C. Sammut & G.I. Webb. "Encyclopedia of machine learning". Springer Science & Business Media (2011).

[18] N. Thai-Nghe, Z. Gantner & L. Schmidt-Thieme. Cost-sensitive learning methods for imbalanced data. In "The 2010 International joint conference on neural networks)". IEEE (2010), p. 1–8.

[19] K. Veropoulos, C. Campbell & N. Cristianini. Controlling the sensitivity of support vector machines. In "Proceedings of the international joint conference on artificial intelligence" (1999), p. 55–60.

[20] Z. Zhao, P. Zhong & Y. Zhao. Learning SVM with weighted maximum margin criterion for classification of imbalanced data. *Mathematical and Computer Modelling*, **54**(3-4) (2011), 1093–1099.

How to cite

D. L. Freire. Cost-Sensitive Algorithms for Text Classification in the Legal Domain: Addressing Imbalanced Lawsuit Themes. *Trends in Computational and Applied Mathematics*, **26**(2025), e01859. doi: 10.5540/tcam.2025.026.e01859.