Experimental Evaluation of Machine Learning Algorithms for Classifying Health-Related News with Indications of Irregularity

Alysson Guimarães[1], Methanias Colaço Junior[1,2], Samuel Almeida[1], Raphael Fontes[3], Helder Prado[3]

[1] Postgraduate Program in Computer Science (PROCC)
Federal University of Sergipe (UFS)
São Cristóvão – SE – Brazil

[2]Health Technological Innovation Laboratory (LAIS)
Onofre Lopes University Hospital
Federal University of Rio Grande do Norte (UFRN)
Natal – RN – Brazil

[3]Advanced Center for Technological Innovation (NAVI)
Federal University of Rio Grande do Norte (UFRN)
Natal – RN – Brazil

{alyssonalk,raphaelf.ti, helderprado}@gmail.com, {mjrse,samuel16.ti}@hotmail.com

Abstract. Research Context*: The increasing production of unstructured data, particularly textual data, has driven the application of Natural Language Processing (NLP) techniques in public administration. The analysis of multiple information sources enables the identification of patterns and the development of predictive models to optimize strategies, improve service delivery, and ensure population monitoring and safety.* ***Scientific and/or Practical Problem****: The auditing process is characterized by high costs, long duration, and heavy reliance on human and material resources, necessitating solutions capable of automating the analysis of corruption reports.* ***Proposed Solution and/or Analysis****: Focusing on the preliminary identification of potential irregularities, the use of machine learning models is proposed to support the auditing process by identifying health-related news that may indicate irregularities.* ***Related IS Theory****: This study is grounded in Cognitive Load Theory, as it examines methods to reduce information overload.* ***Research Method****: A controlled in vitro experiment was conducted to scientifically evaluate 54 machine learning models and compare metrics including Accuracy, Precision, Recall, and F1-score. Additionally, an asymptotic complexity analysis of the algorithms was performed.* ***Summary of Results****: The Random Forest model stood out in terms of effectiveness, achieving an accuracy of 99.90%, a recall of 98.62%, and an F1-score of 99.28%, while Naive Bayes and Logistic Regression excelled in efficiency, with linear complexity $O(nd)$ for both training and prediction and low memory usage.* ***Contributions and Impact to IS area****: The results demonstrate the feasibility of using machine learning models to identify health-related news with potential irregularities. This approach enhances the information gathering and corruption evidence stage performed by AudSUS auditors for detecting*

potential irregularities, thereby contributing to the efficiency of public resource management.

1. Introduction

The increasing production of unstructured data, particularly textual, has driven the application of Natural Language Processing (NLP) techniques in public administration. Governments and institutions employ these tools to process large volumes of documents with the aim of improving the quality of public services, strengthening citizens' trust, and enhancing efficiency in functional areas such as healthcare, education, and decision-making [Jiang et al. 2023].

Real-time analysis of multiple information sources, whether static or dynamic, enables the identification of patterns and the construction of predictive models to optimize strategies and improve service delivery, while also ensuring population monitoring and safety [Benjelloun et al. 2015]. In this context, the healthcare sector stands out as highly vulnerable to fraud and corruption, due to the complexity of its systems, the high financial expenditures involved, and the informational asymmetry among stakeholders [Mackey et al. 2018].

Reports indicate that a significant portion of the global population perceives the healthcare sector as highly susceptible to corruption, with estimates pointing to billion-dollar losses resulting from fraud [Mackey et al. 2018]. Such practices lead not only to financial losses but also to severe social impacts, particularly in low-income countries, where they contribute to increased morbidity and mortality and exacerbate inequalities. Consequently, mechanisms such as reporting channels and audits become essential, although they face challenges related to the large volume of data and informational overload [Paula et al. 2024, do Amaral et al. 2020].

The auditing process is characterized by high cost, long duration, and strong reliance on human and material resources, thereby requiring solutions capable of automating the analysis of corruption allegations. This process generally involves two stages: the preliminary identification of fraud indicators and the subsequent in-depth investigation [Paula et al. 2024].

During the information-gathering phase, various sources are utilized, including websites [Fontes et al. 2023]. To optimize data collection and analysis, *webscraping* techniques can be applied in combination with NLP and *machine learning* methods for the automatic classification of news. These approaches reduce the time and costs associated with information processing while assisting in the identification of content related to potential irregularities in the healthcare sector [Sanchez-Gomez et al. 2022, Benjelloun et al. 2015, Madureira et al. 2021].

Based on this scenario, this article proposes and evaluates the use of *machine learning* models, following the experimental process described in [Colaço Júnior et al. 2022, Colaço Júnior 2025], to investigate their effectiveness and efficiency in text classification. The main objective is to analyze *machine learning* models through a controlled (in vitro) experiment, evaluating them against the results of a human-annotated dataset with respect to accuracy, precision, recall, and F1-score metrics. The focus is on the classification of health-related news articles with indications of irregularities, from the perspective of

Data Scientists and Auditors of the Brazilian Unified Health System (SUS), in the context of public healthcare audits conducted by the National Department of SUS Auditing (AudSUS).

The article is structured as follows: Section 2 presents a concise literature review on the use of *machine learning* for the identification of fraud and corruption, with emphasis on public administration. Section 3 details the dataset employed and the evaluation metrics adopted. In Section 4, the experiment's objective, planning, research questions, dependent and independent variables, study object selection, experimental design, and instrumentation are specified. Section 5 describes the procedures for data preparation, execution, and validation. Subsequently, Section 6 presents the results obtained and discusses threats to validity. Finally, Section 7 provides concluding remarks and outlines directions for future research.

2. Related Work

Automatic detection of corruption and fraud has become a crucial area of research in public management and computer science, driven by the growing volume of data and the need to enhance oversight [Amaral and Rodrigues 2020]. The modernization of public administration requires techniques that enable the identification of patterns to uncover or prevent acts of misconduct [Amaral and Rodrigues 2020]. Various methodologies based on Artificial Intelligence (AI), Machine Learning (ML), and text mining have been proposed to address this challenge in different contexts, such as government audits, public procurement, and the financial sector.

Machine learning-based methods are widely employed to predict and measure corruption, either by exploiting predictive variables in tabular data or by identifying anomalies [Lima and Delen 2020, Ash et al. 2020]. ML approaches can be categorized as supervised or unsupervised. Supervised methods use samples of pre-labeled records (fraudulent and non-fraudulent) to build models that classify new observations [Joudaki et al. 2015].

Among supervised methods, Random Forest (RF), an ensemble algorithm, has proven to be one of the most accurate classification methods for predicting perceptions of corruption in a transnational multiclass setting (achieving 85.77% accuracy, outperforming Support Vector Machines and Artificial Neural Networks) [Lima and Delen 2020]. Tree-based classifiers, such as the Gradient Boosted Classifier (an ensemble of decision trees), have been applied to predict the presence of corruption in local public finances using budgetary data with high accuracy (76%) [Ash et al. 2020]. Gradient Boosting has also shown the best performance in predicting petty corruption intent among law enforcement officers, with accuracy above 90% [Masrom et al. 2023], and achieved high accuracy (71%) in classifying banks involved in corruption scandals in the banking sector [Damiano et al. 2025]. In public procurement fraud detection, ensemble methods (such as Random Forest) have outperformed other ML models in detecting anomalies, collusion, and other types of fraud [Schneider dos Santos et al. 2025].

Other methods like Logistic Regression and Support Vector Machines (SVM) are widely used for predicting corruption indices in public procurement and detecting fraud in healthcare insurance [Rabuzin and Modrušan 2019, Joudaki et al. 2015, Kose et al. 2015]. Logistic Regression has also been employed to assess corruption risk among public officials through imbalanced learning approaches [Vasconcelos et al. 2021].

Artificial Neural Networks (ANN), including the Multilayer Perceptron (MLP) and Deep Learning architectures (such as LSTM, BiLSTM, and CNN), are applied in several domains, including fraud detection in healthcare [Joudaki et al. 2015, Kose et al. 2015] and news classification for corruption risk management [Weichselbraun et al. 2020]. These advanced Deep Learning architectures have outperformed classical ML approaches (such as Naive Bayes and SVM) in text classification tasks [Weichselbraun et al. 2020].

Unsupervised methods are valuable for uncovering hidden patterns in data without the need for prior labels, making them essential for anomaly and outlier detection [Amaral and Rodrigues 2020, Joudaki et al. 2015]. Techniques such as Expectation-Maximization (EM) and K-Means are employed to cluster actors or data with similar behaviors [Kose et al. 2015, Joudaki et al. 2015]. In the context of public procurement, clustering techniques have shown superior results in detecting favoritism [Schneider dos Santos et al. 2025].

For anomaly detection, algorithms such as Isolation Forest (IF), based on binary trees, have proven efficient, particularly in public procurement, due to their reduced runtime and memory requirements [Schneider dos Santos et al. 2025]. PAACDA was proposed for comprehensive corruption detection in numerical tabular data. It employs a graph-based approach (Adamic Adar) to detect outliers and missing or modified values, achieving high precision (99.04% for linear data and 96.35% for clustered data) [Bannur et al. 2023]. Interactive Machine Learning (IML) incorporates expert knowledge directly into the model-building process, making it vital in dynamic fraud ecosystems where corruption patterns evolve rapidly [Kose et al. 2015].

Text mining, also known as Knowledge Discovery in Text (KDT), refers to the process of uncovering potentially useful knowledge from unstructured databases, which is indispensable given that much of the relevant data exists in textual form [Amaral and Rodrigues 2020].

Latent Dirichlet Allocation (LDA) is a topic modeling algorithm that applies a Bayesian approach to segment collections of documents by dominant topics. LDA has been successfully applied to government audit data (over 65,000 purchase records) to segment items by semantic proximity [Amaral and Rodrigues 2020]. In public administration studies, LDA has been adopted to map research trends on corruption [Caputo et al. 2025].

Textual analysis combined with machine learning algorithms has been applied to detect corruption scandals in banks by analyzing financial reports [Damiano et al. 2025].

An innovative approach leverages advanced language models, such as the Large Language Model (LLM) FinBERT (a BERT adaptation for the financial domain), in conjunction with dictionary-based methods to extract and analyze the tone (positive, negative, and litigious) of governance-related disclosures. This is used as a predictive tool to detect corruption scandals before they become public [Damiano et al. 2025].

Natural Language Processing (NLP) is employed to classify media documents (such as news articles) according to their corruption risk, using algorithms such as Naive Bayes, SVM, and Deep Learning architectures (e.g., CNN, LSTM) [Weichselbraun et al. 2020]. NLP is also applied to extract information from procurement documents and detect signs of corruption [Rabuzin and Modrušan 2019, Schneider dos Santos et al. 2025].

3. Materials and Method

This is an experimental study, following the steps presented by [Colaço Júnior et al. 2022, Colaço Júnior 2025] for evaluating the results of machine learning models applied to text classification, using health-related news articles with indications of irregularity, and assessing the quality of summaries using the Accuracy, Precision, Recall, and F1 metrics.

The description of the database used, the process of selecting news articles with indications of irregularity, and the evaluation metrics of the results are presented in Subsections 3.1 and 3.2, respectively.

Experiment replication is a key feature of any scientific field. In the software domain, it is therefore necessary to apply methods that can be replicated and evaluated, in order to prevent new methods, techniques, languages, and tools from being suggested, published, or marketed without experimentation and validation [Travassos et al. 2020]. Finally, the experiment definition and planning, along with all its stages, are described in detail in Section 4.

3.1. Database

The database consists of a collection of 154,407 news articles gathered from the internet. It was constructed by [Fontes et al. 2023] in three stages: a proof of concept, an exploratory study, and the final database construction.

To identify health-related news articles with audit indications, a three-step keyword-based selection strategy was executed. In the first stage, using the entire database, news articles containing the keyword **"saúde"** in their **content** (corpus) were classified as "Health," while all others were classified as "Generic News." The generic news articles were excluded from the subsequent stages. In this stage, 9,516 articles were classified as "Health," while the remaining 144,891 articles were excluded as generic news.

In the second stage, keywords indicating irregularity were applied to the set of articles previously classified as "Health," dividing the group into those that did or did not contain such keywords, respectively "Generic Health" and "Health Irregularity." Articles containing **at least one keyword** in the title and/or abstract (headline) were classified as "Irregularity News." In this stage, 6,239 articles with indications of irregularity were identified, while the remaining 3,277 were classified as "Generic Health." Table 1 presents the keywords used.

In the third stage, articles in the "Irregularity News" subgroup were independently evaluated by two annotators. In case of disagreement, the final decision was made by five additional evaluators. The evaluation process involved reading the title and abstract of all articles to confirm their classification as "Health Irregularity," "Generic Health," or "Generic News."

Finally, after the third stage, 421 health-related news articles with indications of irregularity were identified. Figure 1 illustrates the news selection process. The raw and annotated dataset is publicly available on the Zenodo Platform [Guimarães et al. 2025].

3.2. Evaluation Metrics

The following evaluation metrics were used to assess the classification results: Accuracy, Precision, Recall, and F1-Score [Zhu et al. 2010]. Table 2 describes the definition and

Table 1. Keywords used to identify signs of irregularities.

Keywords
abuso, abuso de poder, acordo ilegal, acusaç, acusação, apropriação indébita, auditoria, aumento orçamento, bilhões, cartel, coação, compra, compras públicas, conluio, contrato, contratos, corrupto, corrupção, corte orçamento, crime, crime organizado, criminoso, deflagrou, denuncia, denúncia, desassistência, desfalque, desonestidade, desonesto, desperdício, desvio, desvios, disfarce, documento alterado, dolo, enganar, engano, enganos, enganoso, enriquecimento, enriquecimento ilícito, escândalo, esquema, evasão, falcatrua, falsa declaração, falsificado, falsificador, falsificação, falso, falta, falta de equipamentos, falta equipamento, fiscalização, forjar, fraudador, fraudar, fraude, fraude em contratos, fraude em licitações, fraude financeira, fraude licitação, fraudulento, fugir, golpe, ilegal, ilusão, ilícito, indicativo, indício, infração, investiga, investigação, irregular, irregularidade, irregularidade administrativa, irregularidade de gestão, irregularidade financeira, irregularidades, lavagem, lavagem de dinheiro, licitação, mandado, manipulado, manipulador, manipulação, manipulação de dados, maquiar, mentira, milhões, má conduta, negligência dolosa, ocultar, ocultação, PF, peculato, perjúrio, plano, plano de saúde, plano saúde, prevaricação, propina, recurso, relatório falso, rombo, roubo, sem autorização, sem consentimento, sobrefaturamento, sonegar, sonegação, suborno, sugestão, superfaturamento, suspeit, suspeita, suspeito, transação, transação suspeita, transgressão, transparência, uso indevido, uso indevido de recursos, plano saude, uso irregular, venda.

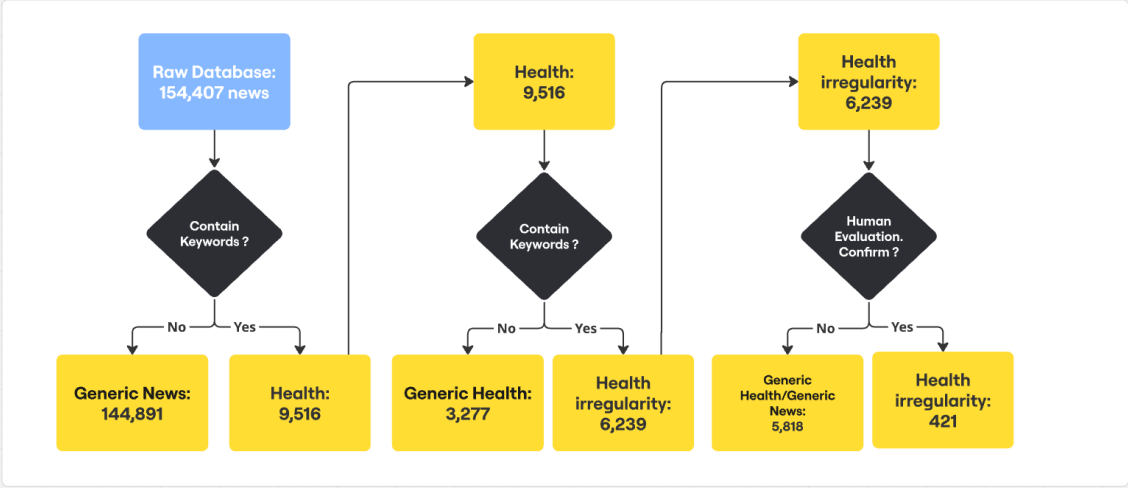

Figure 1. Database classification process.

calculation of each metric. These metrics are based on the frequencies of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), which represent:

- **True Positive (TP)**: The total number of positive instances (news articles) that were correctly classified as positive.
- **True Negative (TN)**: The total number of negative instances (news articles) that were correctly classified as negative.
- **False Positive (FP)**: The total number of negative instances (news articles) that were incorrectly classified as positive.
- **False Negative (FN)**: The total number of positive instances (news articles) that were incorrectly classified as negative.

4. Experimental Definition

This section presents the objective of the experimental evaluation, the planning, the research questions, the independent variables, the dependent variables, and the hypotheses.

Table 2. Evaluation Metrics for Classifiers

Metric	Description	Formula
Accuracy	Represents the percentage of instances (news articles) that were correctly classified, considering both true positives and true negatives.	accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$
Precision	Measures the proportion of instances predicted as positive that actually belong to the positive class, i.e., the reliability of positive predictions.	precision = $\frac{TP}{TP+FP}$
Recall	Also called sensitivity, measures the proportion of positive instances that were correctly identified by the model.	recall = $\frac{TP}{TP+FN}$
F1-score	Harmonic mean between precision and recall, aiming to balance the two indicators; particularly useful in scenarios with imbalanced classes.	F1-score = $2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision}+\text{recall}}$

4.1. Objective

To formalize the objective of this study, the *Goal Question Metric* (GQM) model proposed by [Basili and Weiss 1984] was adopted. This study aims to **analyze** machine learning models through a controlled experiment (*in vitro*), **with the purpose of** evaluating them against the results of a human-classified dataset, **with respect to** the metrics of accuracy, precision, recall, and F1-score, **in relation to** the classification of health-related news with indications of irregularity, **from the perspective of** Data Scientists and Auditors of the Brazilian Unified Health System (SUS), **in the context of** public health audits conducted by the National Department of SUS Auditing (AudSUS).

4.2. Planning

The experiment was conducted in a controlled *in vitro* environment, using the dataset described in Subsection 3.1 and machine learning models to classify the news articles. The filtered dataset contains 6,239 news articles.

4.3. Context Selection

Despite technological advances, many processes in the public sector still rely on manual searches for knowledge construction. This scenario is also observed at the National Department of Auditing of the Brazilian Unified Health System (AudSUS), responsible for monitoring and auditing SUS. The auditing activities carried out by this agency play a crucial role in the management and proper utilization of public resources; however, the process is highly resource-intensive due to the large demand, as auditors must oversee all SUS areas while also addressing internal and external demands from the Ministry of Health [Fontes et al. 2023]. In this context, the proposed experiment aims to support the analytical phase of the auditing process, which is responsible for planning and preparing the team for the operational phase, by collecting information related to the audit objectives.

4.4. Research Question

To guide the experiment and fulfill the study's objective, the following research questions were formulated:

- RQ1: Which of the selected algorithms are the most effective?
- RQ2: Among the selected algorithms, which are the most efficient?

To address the research questions, the following theoretical hypotheses described in Table 3 were created.

Table 3. Research questions and associated hypotheses

RQ	Null Hypothesis (H_0)	Alternative Hypothesis (H_1)
RQ1	The algorithms have the same effectiveness.	The algorithms do not have the same effectiveness.
RQ2	The models have the same efficiency.	The models do not have the same efficiency.

4.5. Dependent Variables

The dependent variables, or output variables, were the classified news articles, from which the metrics Accuracy, Precision, Recall, and F1-score can be derived.

4.6. Independent Variables

In this experiment, the independent variables, or input variables, are: the annotated (classified) dataset of news articles with indications of irregularity; and the algorithms used for the classification task: *AdaBoost, CatBoost (CatBoostClassifier), GradientBoosting, KNN, LightGBM (LGBMClassifier), Logistic Regression, Naive Bayes (Multinomial), RandomForest, Support Vector Classification (SVC), XGBoost (XGBClassifier).*

4.7. Objects Selection

Following the context described in 4.3, the objects of this experiment are health-related news articles with indications of irregularity, as described in Section 3.1. For the sample size calculation, a finite population of 154,407 news articles, i.e., the total number of articles in the complete dataset, was considered. It is noteworthy that the final sample exceeds the estimated number according to Eq. 2. For the sample calculation, a 95% confidence level (value of $Z = 1{,}96$), a tolerable sampling error of 5% ($e = 0{,}05$), and an expected proportion of 50% ($p = 0{,}5$) were considered, which maximizes the variability of the sample and ensures a more conservative size.

The sample size calculation for a finite population was performed in two stages: first, the sample for an infinite population (n) was estimated using Eq. 1, and then the adjustment for a finite population ($n_{adjusted}$) was applied according to Eq. 2, resulting in approximately 383.21 samples, as shown in Eq. 4. Finally, all manually classified samples from the dataset were used, totaling 6,239, comprising 421 news articles in the class "Health Irregularity" and the remaining in the class "Generic Health," which exceeds the minimum requirement of 384 articles for a representative sample.

$$n = \frac{Z^2 \cdot p \cdot (1-p)}{e^2} \tag{1}$$

$$n_{adjusted} = \frac{n}{1 + \left(\frac{n-1}{N}\right)} \tag{2}$$

$$n = \frac{1{,}96^2 \cdot 0{,}5 \cdot (1-0{,}5)}{0{,}05^2} = \frac{3{,}8416 \cdot 0{,}25}{0{,}0025} = 384{,}16 \tag{3}$$

$$n_{adjusted} = \frac{384{,}16}{1 + \left(\frac{384{,}16-1}{154407}\right)} \approx 383{,}21 \tag{4}$$

4.8. Experiment Design

The classification was carried out in 35 rounds using the 9 selected algorithms. For each round, a combination of hyperparameter optimization methods (default, random, or bayes) with training methods (hold out or cross validation) was employed, resulting in 6 combinations of classifiers ('default' and 'hold out', 'default' and 'cross validation', 'random' and 'hold out', 'random' and 'cross validation', 'bayes' and 'hold out', or 'bayes' and 'cross validation'). This totaled 54 models, each executed 35 times for the selected algorithms.

Before executing the classification pipeline, a preprocessing stage was necessary in which Term Frequency-Inverse Document Frequency (TF-IDF) was applied, as described by [Salton and Buckley 1988], to account for the relative importance of words in the news headlines and create the attributes (features) or independent variables of the dataset. After data preparation, the execution pipeline was run.

4.9. Instrumentation

The following materials and resources were used:

- Google Sheets;
- Annotated dataset with reference summaries (3.1);
- Google Colab [1];
- Python programming language (3.11.13)[2];
- Python libraries: catboost (1.2.8), lightgbm (4.6.0), matplotlib (3.10.5), numpy (2.2.6), openpyxl (3.1.4), pandas (2.3.1), pyarrow (21.0.0), scikit-learn (1.7.1), scikit-optimize (0.10.2), seaborn (0.13.2), sentence-transformers (5.1.0), tqdm (4.67.1), xgboost (3.0.4), and uv (0.8.14);
- Computational resources from the High-Performance Computing Center (NPAD) at the Federal University of Rio Grande do Norte (UFRN).

5. Experiment Operation

This section describes the experiment preparation process, execution, and evaluation of results.

5.1. Experiment Preparation

The execution environment on the NPAD supercomputer was prepared using the *uv* library, which was employed to create the virtual environment and install the necessary libraries described in Subsection 4.9 for the classification task; subsequently, the dataset was uploaded to the environment.

To ensure a systematic execution process, a classification *pipeline* was developed. This *pipeline* consists of a script that performs the steps for each candidate model described in Subsection 4.6 and for each of the 6 combinations of classifier types described in Subsection 4.8. Algorithm 1 details the execution pipeline. To guarantee reproducibility, the random parameter (random_state) was set to 42 and the test size was fixed at 30% of

[1]https://colab.google/

[2]https://www.python.org/

the dataset. Table 4 describes the search parameters used in hyperparameter optimization. The implementation code of the experimentation pipeline are publicy available on github[3].

As a pilot study, the *pipeline* was initially tested in 5 rounds with 25 samples to verify its operation. Necessary adjustments and potential failures were corrected during this preliminary stage. Subsequently, the full process was executed for all methods.

Algorithm 1 Classification Models Experimentation Pipeline

Define **data**: $X \leftarrow$ features, $y \leftarrow$ labels
Define candidate **models**
Define **search spaces** for hyperparameters
Select **text representation** (*TF-IDF*)
Initialize lists for **results** and **metrics**
for each **model** in models **do**
  for each iteration $r = 1$ **to** n_round **do**
    if training strategy = **Hold-out then**
      Split data into train and test
    else if training strategy = **Cross-validation then**
      Define validation folds
    end if
    if hyperparameter tuning = **Random** or **Bayes then**
      Perform search over hyperparameter space
      Select best model and parameters
    else
      Train model with default parameters
    end if
    Generate predictions on the test set
    Store predictions, probabilities, and parameters
  end for
end for
Aggregate results across all iterations
Compute evaluation metrics (*accuracy*, *precision*, *recall*, *F1*, *ROC-AUC*)
Return final tables of **results** and **metrics** =0

Table 4. Search spaces for model hyperparameters

Model	Search parameters
LogReg	$C \in \{0.01, 0.1, 1, 10\}$
RandomForest	$n\_estimators \in \{100, 200, 500\}$, $max\_depth \in \{\text{None}, 10, 20\}$
SVC	$C \in \{0.1, 1, 10\}$
NaiveBayesMultinomial	$\alpha \in \{0.1, 1, 10\}$, $fit\_prior \in \{\text{True}, \text{False}\}$
KNN	$n\_neighbors \in \{3, 5, 7\}$
GradientBoosting	$n\_estimators \in \{100, 200\}$, $learning\_rate \in \{0.05, 0.1\}$, $max\_depth \in \{3, 6, 10\}$
AdaBoost	$n\_estimators \in \{50, 100, 200\}$, $learning\_rate \in \{0.05, 0.1\}$
XGBoost	$n\_estimators \in \{100, 200\}$, $learning\_rate \in \{0.05, 0.1\}$, $max\_depth \in \{3, 6, 10\}$
LightGBM	$n\_estimators \in \{100, 200\}$, $learning\_rate \in \{0.05, 0.1\}$, $max\_depth \in \{3, 6, 10\}$

5.2. Experiment Execution

The execution of the experiment consisted of two stages: pipeline execution (1) and results evaluation.

[3]https://github.com/k3ybladewielder/guimaraes2026experimental

The pipeline was designed to perform preprocessing systematically: lowercasing and TF-IDF. All words are converted to lowercase and, subsequently, through TF-IDF, the texts are transformed into vectors containing the total words, filled with the corresponding value of their relative importance within the corpus.

Afterward, the models are trained using 70% of the dataset as the training set and 30% as the test set. Each model is trained using evaluation type parameters (hold-out or cross-validation) and hyperparameter optimization strategies (default, random search, or Bayesian search) across 35 rounds.

The hold-out evaluation type employs a fixed sample for training and the remainder for testing. In contrast, the cross-validation method splits the dataset into 5 stratified subsets, such that in each round a different subset is used as the test set while the others are used for training. This process ensures greater robustness in model evaluation, as all examples in the dataset are used for both training and testing across different iterations.

Hyperparameter optimization was performed in three ways:

- **Default**: the model is trained directly with its default parameters.
- **Random Search**: random samples of hyperparameter combinations are evaluated to identify better configurations.
- **Bayesian Search**: employs an iterative process based on Bayesian optimization, selecting new parameter combinations more intelligently by considering previous results.

Each round generates predictions, class-associated probabilities, and records the parameters used, enabling comparison of models and optimization strategies.

5.3. Data Evaluation

Four (4) types of statistical tests were employed for data analysis, interpretation, and validation: Anderson-Darling (AD Test), Kolmogorov-Smirnov (KS Test), paired t-test, and Wilcoxon Signed-Rank Test (paired). The Anderson-Darling and Kolmogorov-Smirnov tests were applied to verify the normality of the data. For the evaluation of paired models that showed evidence of normality, the t-test was used, whereas the Wilcoxon Signed-Rank Test was applied to compare the medians of the metrics in cases without evidence of normality.

6. Results

This section describes the process of Data Analysis and Interpretation, Threats to Validity, Conclusions, and Future Work.

6.1. Data Analysis and Interpretation

To address the research questions presented in 4.4, the execution stage was carried out, and classification results were obtained for the defined evaluation metrics. Figure 2 visually represents the performance by model, ordered by F1 Score.

The models SVC (cross-validation random), Random Forest (cross-validation default), and Random Forest (cross-validation random) achieved the highest accuracies. Meanwhile, Gradient Boosting (cross-validation default), Gradient Boosting (cross-validation bayes), and Gradient Boosting (cross-validation random) achieved the best precision. Regarding recall, the Random Forest (cross-validation random), Random Forest

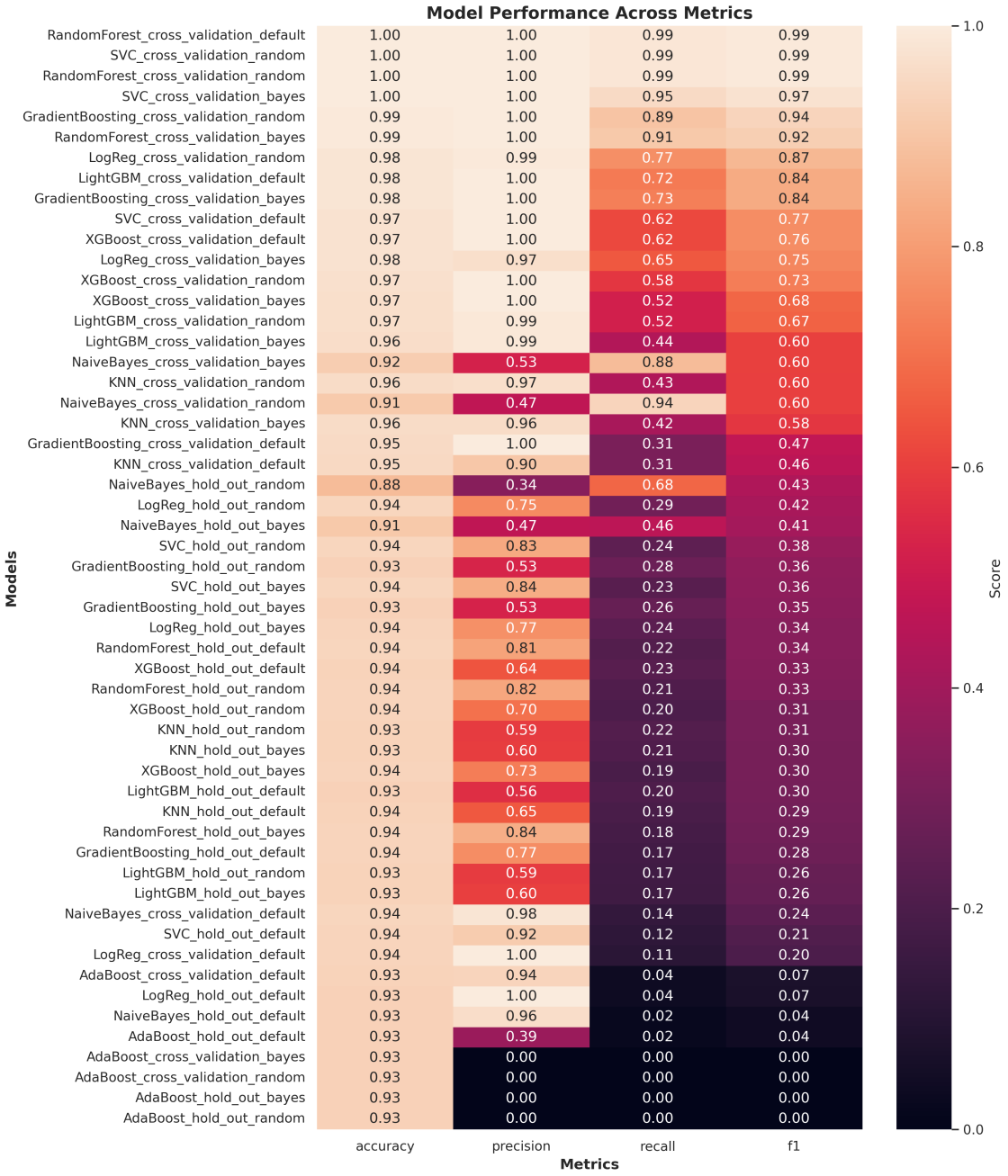

Figure 2. Heatmap of Evaluation Metrics per Model. Sorted by F1 Score.

(cross-validation default), and SVC (cross-validation default) models stood out. Finally, Random Forest (cross-validation default), SVC (cross-validation random), and Random Forest (cross-validation random) presented the best average results for F1-score. Overall, with the exception of precision, Random Forest (cross-validation random), Random Forest (cross-validation default), and SVC (cross-validation random) stood out across all metrics.

To compare how good the algorithms are relative to one another, conclusive statistical evidence is required. For this purpose, the Anderson-Darling (AD Test) and Kolmogorov-Smirnov (KS Test) were applied. Evidence indicates that some models and configurations follow a normal distribution, as shown in Table 5.

Table 5. AD Statistic test results and rejection at the 5% level for different models and metrics. Critical value of 0.719 and N of 35

Model Card	AD_Statistic				AD_Reject_at_5%			
	accuracy	f1	precision	recall	accuracy	f1	precision	recall
AdaBoost_cross_validation_bayes	0.000000	0.000000	0.000000	0.000000	False	False	False	False
AdaBoost_cross_validation_default	0.000000	0.000000	34.863135	34.863135	False	False	True	True
AdaBoost_cross_validation_random	0.000000	0.000000	0.000000	0.000000	False	False	False	False
AdaBoost_hold_out_bayes	0.199108	0.000000	0.000000	0.000000	False	False	False	False
AdaBoost_hold_out_default	0.298551	1.953894	1.861945	2.102234	False	True	True	True
AdaBoost_hold_out_random	0.199108	0.000000	0.000000	0.000000	False	False	False	False
GradientBoosting_cross_validation_bayes	0.858945	0.810436	0.000000	0.858945	True	True	False	True
GradientBoosting_cross_validation_default	0.444682	0.445671	0.000000	0.444682	False	False	False	False
GradientBoosting_cross_validation_random	4.275143	4.802213	0.000000	4.275143	True	True	False	True
GradientBoosting_hold_out_bayes	0.197463	0.233263	0.319014	0.284222	False	False	False	False
GradientBoosting_hold_out_default	0.554966	0.258988	0.187784	0.280080	False	False	False	False
GradientBoosting_hold_out_random	0.506313	0.153878	0.508408	0.368565	False	False	False	False
KNN_cross_validation_bayes	10.434122	10.434122	10.434122	10.434122	True	True	True	True
KNN_cross_validation_default	34.863135	0.000000	34.863135	34.863135	True	False	True	True
KNN_cross_validation_random	34.863135	34.863135	34.863135	34.863135	True	True	True	True
KNN_hold_out_bayes	0.191272	0.316983	0.420982	0.293760	False	False	False	False
KNN_hold_out_default	0.638736	0.154533	0.320086	0.248038	False	False	False	False
KNN_hold_out_random	0.161830	0.291198	0.572434	0.341261	False	False	False	False
LightGBM_cross_validation_bayes	0.311995	0.305245	2.518436	0.323909	False	False	True	False
LightGBM_cross_validation_default	34.863135	34.863135	34.863135	0.000000	True	True	True	False
LightGBM_cross_validation_random	2.995580	3.060265	8.790676	2.976609	True	True	True	True
LightGBM_hold_out_bayes	0.267097	0.578166	0.586358	0.478697	False	False	False	False
LightGBM_hold_out_default	0.446770	0.373120	0.246842	0.330400	False	False	False	False
LightGBM_hold_out_random	0.246480	0.583085	0.452593	0.474505	False	False	False	False
LogReg_cross_validation_bayes	9.628126	9.503945	12.799091	9.629308	True	True	True	True
LogReg_cross_validation_default	0.000000	34.863135	0.000000	34.863135	False	True	False	True
LogReg_cross_validation_random	34.863135	0.000000	34.863135	0.000000	True	False	True	False
LogReg_hold_out_bayes	0.330636	4.456116	2.654289	3.697330	False	True	True	True
LogReg_hold_out_default	0.193064	0.507881	0.000000	0.536271	False	False	False	False
LogReg_hold_out_random	0.145412	0.666314	0.269488	0.860037	False	False	False	True
NaiveBayes_cross_validation_bayes	7.197509	5.721915	7.722218	5.587578	True	True	True	True
NaiveBayes_cross_validation_default	0.000000	0.000000	34.863135	34.863135	False	False	True	True
NaiveBayes_cross_validation_random	10.434122	10.434122	10.434122	10.434122	True	True	True	True
NaiveBayes_hold_out_bayes	1.926970	2.942268	1.034293	1.929615	True	True	True	True
NaiveBayes_hold_out_default	0.223302	0.366367	11.949051	0.362413	False	False	True	False
NaiveBayes_hold_out_random	3.922155	0.408525	4.972972	4.544995	True	False	True	True
RandomForest_cross_validation_bayes	11.744037	11.745579	7.687040	11.649086	True	True	True	True
RandomForest_cross_validation_default	0.000000	8.634957	8.634957	8.634957	False	True	True	True
RandomForest_cross_validation_random	12.437239	11.606703	6.341815	5.808617	True	True	True	True
RandomForest_hold_out_bayes	0.409875	2.689163	0.619997	2.089862	False	True	False	True
RandomForest_hold_out_default	0.436902	0.310446	0.271578	0.233527	False	False	False	False
RandomForest_hold_out_random	0.342452	2.056600	0.395991	1.408790	False	True	False	True
SVC_cross_validation_bayes	11.781270	11.781270	0.000000	11.781270	True	True	False	True
SVC_cross_validation_default	34.863135	0.000000	0.000000	34.863135	True	False	False	True
SVC_cross_validation_random	0.000000	0.000000	0.000000	34.863135	False	False	False	True
SVC_hold_out_bayes	0.329687	1.043737	0.360275	0.848211	False	True	False	True
SVC_hold_out_default	0.348676	0.376290	0.562678	0.465908	False	False	False	False
SVC_hold_out_random	0.409029	0.187146	0.366474	0.413657	False	False	False	False
XGBoost_cross_validation_bayes	0.526720	0.679656	7.630495	0.523515	False	False	True	False
XGBoost_cross_validation_default	0.000000	34.863135	34.863135	0.000000	False	True	True	False
XGBoost_cross_validation_random	4.071622	3.792327	6.011999	4.077454	True	True	True	True
XGBoost_hold_out_bayes	0.349994	0.160523	0.346832	0.215693	False	False	False	False
XGBoost_hold_out_default	0.246101	0.554307	0.181937	0.425533	False	False	False	False
XGBoost_hold_out_random	0.371639	0.613486	0.389416	0.434931	False	False	False	False

The results were divided into two groups: Group A, where there is evidence that the results follow a normal distribution, and Group B, where there is no evidence of normality. To evaluate the performance of paired models in Group A, the paired t-test was used, while the Wilcoxon Signed-Rank Test was applied for Group B. Table 6 describes the number of times the evaluated model outperformed the others. Figure 3 illustrates the total score per model.

To analyze the efficiency of the algorithms, the focus was placed on identifying

Table 6. Total score of the 15 best models by metric and overall sum.

Model	Accuracy	F1	Precision	Recall	ROC AUC	Total
RandomForest_cross_validation_default	51	51	42	49	53	246
RandomForest_cross_validation_random	49	50	40	49	51	239
SVC_cross_validation_random	51	50	45	48	45	239
RandomForest_cross_validation_bayes	49	49	40	47	51	236
SVC_cross_validation_bayes	49	49	45	47	46	236
GradientBoosting_cross_validation_random	48	48	45	46	45	232
SVC_cross_validation_default	43	43	45	40	49	220
LogReg_cross_validation_random	46	46	34	44	48	218
GradientBoosting_cross_validation_bayes	44	44	45	41	43	217
LogReg_cross_validation_bayes	45	45	34	42	45	211
LightGBM_cross_validation_default	44	44	39	41	43	211
XGBoost_cross_validation_default	42	42	38	39	36	197
XGBoost_cross_validation_random	41	41	40	38	36	196
XGBoost_cross_validation_bayes	39	39	40	35	35	188
LightGBM_cross_validation_random	39	39	34	35	35	182
KNN_cross_validation_random	37	36	32	33	39	177
GradientBoosting_cross_validation_default	35	33	45	30	30	173
LightGBM_cross_validation_bayes	37	34	34	32	33	170
KNN_cross_validation_bayes	36	34	31	32	37	170
KNN_cross_validation_default	34	32	28	30	32	156

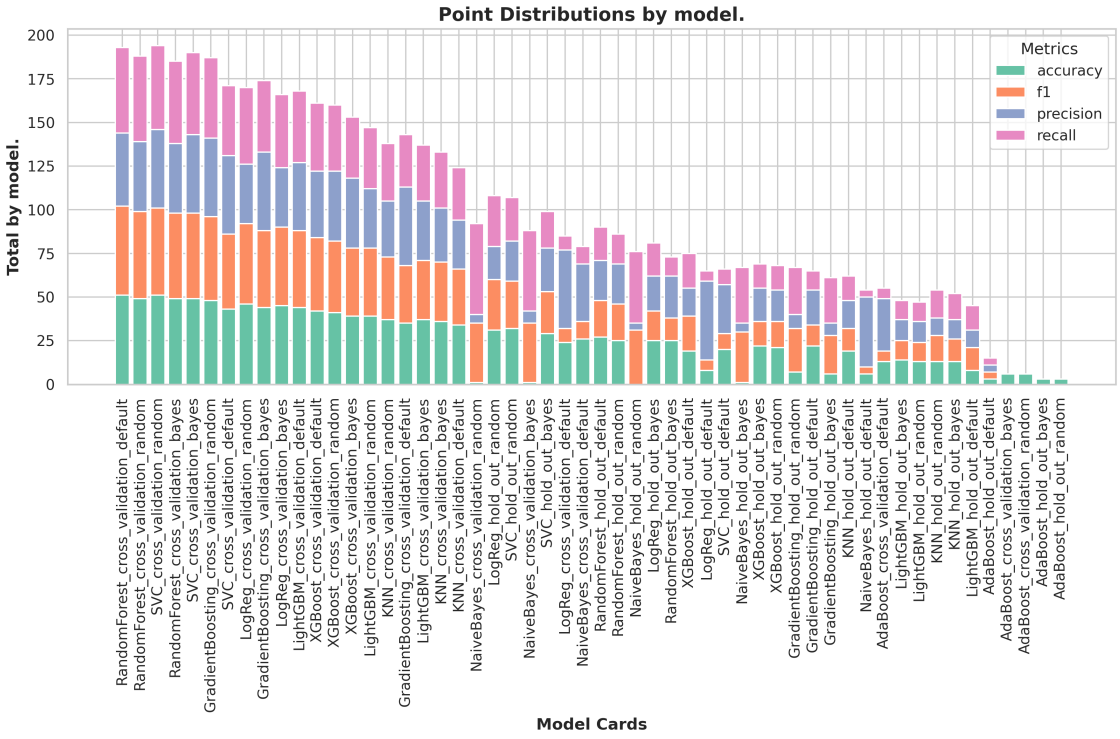

Figure 3. Point distributions by model.

the order of magnitude of basic operation counts as the primary indicator of algorithmic efficiency [Levitin 2012]. To compare and rank these orders of magnitude, Big-O notation was applied. Table 7 describes the selected models and their complexity for training and prediction. The basic operations were identified with the following notations:

- n = number of dataset samples;
- d = number of attributes (dimensionality);
- I = number of optimizer iterations;

- T = number of trees (for ensembles and boosting);
- D = average tree depth;
- C = number of classes;
- s = number of support vectors (in kernelized SVC).

Table 7. Summary of classification models, complexities, and memory usage.

Model	Description	Training	Prediction	Memory Usage
Logistic Regression	Maximize probability via logistic regression	$O(n \cdot d \cdot I)$	$O(d)$	$O(d)$
RandomForest	Ensemble de T trees trained on subsets of the dataset	$O(T \cdot n \cdot d \cdot \log n)$	$O(T \cdot D)$	$O(T \cdot n)$
GradientBoosting	Sequential trees correcting errors from previous ones	$O(T \cdot n \cdot d \cdot \log n)$	$O(T \cdot D)$	$O(T \cdot n)$
AdaBoost	T weighted combined weak classifiers	$O(T \cdot n \cdot d)$	$O(T \cdot D)$	$O(T \cdot n)$
C-SVC	Multiclass SVM via one-vs-one with kernel	$O(n^2 \cdot d + n^3)$ per classifier	$O(s \cdot d)$	$O(n^2)$
Naive Bayes (MultinomialNB)	Frequency count by class, assumes independence	$O(n \cdot d)$	$O(d)$	$O(d \cdot C)$
KNeighborsClassifier	Lazy learning, classify by proximity	$O(n \cdot d)$	$O(n \cdot d)$ (ou $O(d \cdot \log n)$ com KD-tree)	$O(n \cdot d)$
XGBClassifier	Optimized boosting with histograms	$O(T \cdot n \cdot d)$	$O(T \cdot D)$	$O(T \cdot n)$
LGBMClassifier	Efficient leaf-wise boosting on large datasets	$O(T \cdot (n + d))$	$O(T \cdot D)$	$O(T \cdot n)$

Among the machine learning algorithms, those with the shortest training time are Naive Bayes and KNN, both with linear complexity in relation to the number of samples and attributes. The former performs only frequency counting, while the latter is limited to data storage. Next, Logistic Regression also exhibits linear complexity, but its cost depends on the number of optimizer iterations. More recent *boosting* methods, such as XGBoost, LightGBM, and CatBoost, demonstrate higher efficiency in tree construction compared to traditional AdaBoost. Random Forest and Gradient Boosting from *scikit-learn*, however, require higher computational cost, with complexity of $O(nd \log n)$ per tree. Finally, C-SVC (libsvm) is the most computationally expensive, since quadratic optimization grows between $O(n^2 d)$ and $O(n^3)$, making its application infeasible for large-scale datasets.

In the prediction process, the most efficient algorithms are Naive Bayes and Logistic Regression, both with linear cost in relation to the number of attributes per sample ($O(d)$). Following them are the tree-based ensembles AdaBoost, RandomForest, GradientBoosting, XGBoost, LightGBM, and CatBoost, whose cost is proportional to the number of trees multiplied by their average depth ($O(TD)$), which proves efficient in practical scenarios. C-SVC (libsvm) achieves intermediate performance, as its cost depends on the number of support vectors ($O(sd)$), which may vary according to the dataset. KNN, in contrast, is the least efficient, as it requires comparing the test sample with all stored points ($O(nd)$), which becomes impractical for large datasets.

Regarding memory consumption, the lightest algorithms are Naive Bayes and Logistic Regression, which only require the storage of frequencies or a weight vector, with cost $O(dC)$ or $O(d)$. KNN demands higher memory usage, as it requires storing the entire dataset ($O(nd)$). Ensembles (AdaBoost, RandomForest, GradientBoosting, XGBoost, LightGBM, and CatBoost) require even more memory, since all trees must be

stored ($O(Tn)$). The highest cost, however, is observed in C-SVC (libsvm), which needs to store the kernel matrix of size $n \times n$, resulting in quadratic cost ($O(n^2)$), a factor that limits its applicability for large datasets.

From the auditor's perspective, the choice of the most suitable models depends on factors such as factuality, urgency, availability of computational resources, inference time, and type of application (streaming or batch processing). In scenarios that demand higher precision in identifying news with indications of irregularities, as evaluated by metrics such as *accuracy*, *precision*, *recall*, and *F1-score*, it is preferable to employ models that exhibit stronger performance across these metrics, such as *Random Forest* (cross-validation default), *Random Forest* (cross-validation random), and *SVC* (cross-validation random), which outperformed across all evaluation metrics in the experiment. In situations requiring greater urgency and agility in planning, models with lower theoretical complexity in training, prediction, and memory usage, such as *Naive Bayes*, *Logistic Regression*, or *LightGBM*, are recommended.

6.2. Threats to Validity

For the evaluation of the experiment, it is necessary to consider factors that may influence the results, characterized as threats to internal and external validity.

- **Internal Validity**: The news classification process was conducted by two annotators. Since this is a manual and intensive activity, classification errors may occur. To mitigate this risk, a third evaluator intervened in cases of disagreement between the annotators regarding the categorization of the news.
- **External Validity**: Considering the nature of the data (news articles) and the fact that only the titles (*headlines*) were used in this experiment, the low linguistic variability could hinder the classification task. To mitigate this issue, the experimental objects were selected in a robust and representative manner. Furthermore, robust training methods were applied, such as stratified cross-validation and the use of an independent test set in each iteration.

7. Conclusion and future work

The audit process is generally characterized as costly, time-consuming, and demanding in terms of substantial human and material resources. In this regard, it is necessary to implement solutions and techniques that enable the automation of the analysis of corruption complaints. This process is usually divided into two stages: in the first, the aim is to identify elements and evidence of corruption, such as suppliers, contracts, employees, clients, and other stakeholders, assessing the plausibility and consistency of complaints and indications of fraud; in the second stage, the investigation itself is carried out.

To build the knowledge required for audit activities, it is essential to collect information related to the audit's objectives. At this stage, various sources are consulted, including web pages. To support the information collection process, *webscraping* techniques can be applied to extract large-scale data from health-related websites. Furthermore, to assist in analyzing this large volume of data, NLP techniques such as text summarization can be employed, significantly reducing the time and resources needed for analyzing and gathering evidence of possible irregularities.

In this context, with the objective of supporting, improving, and optimizing the collection of relevant information that may assist in combating irregularities, this study presents the results of applying 54 machine learning classification models to a set of health-related news articles with indications of irregularity. The aim was to evaluate whether such methods could contribute to the audit process by directly identifying news articles with indications of irregularity, as well as determining which models are most efficient and effective for this task.

In this controlled **in vitro** experiment, using a curated dataset of 6,239 news samples, 54 machine learning classification models were executed, repeated across 35 rounds to robustly measure model effectiveness and efficiency. Effectiveness was assessed using performance metrics Accuracy, Precision, Recall, and F1-Score [Zhu et al. 2010]. Efficiency was measured using asymptotic analysis (Big-O) to identify the complexity of training, prediction, and memory usage.

Regarding effectiveness, the models SVC (cross-validation random), Random Forest (cross-validation default), and Random Forest (cross-validation random) presented the best accuracies. Gradient Boosting (cross-validation default), Gradient Boosting (cross-validation bayes), and Gradient Boosting (cross-validation random) achieved the best precision scores. For recall, Random Forest (cross-validation random), Random Forest (cross-validation default), and SVC (cross-validation default) stood out. Finally, Random Forest (cross-validation default), SVC (cross-validation random), and Random Forest (cross-validation random) obtained the best average results for F1-score. Overall, with the exception of precision, Random Forest (cross-validation random), Random Forest (cross-validation default), and SVC (cross-validation random) stood out across all metrics.

Based on these results, normality was analyzed through Anderson-Darling (AD) and Kolmogorov-Smirnov (KS) tests. In pairwise analysis, results with evidence of normality were evaluated using the paired t-test, while the Wilcoxon Signed-Rank Test was employed to compare metric medians in cases without evidence of normality. The outcomes of these tests are presented in this study. In the pairwise comparisons, Random Forest (cross-validation random), Random Forest (cross-validation default), and SVC (cross-validation random) ranked as the first, second, and third best-performing models overall.

With respect to efficiency, the algorithms with the shortest training time are Naive Bayes and KNN, both with linear complexity in relation to the number of samples and attributes. In the prediction process, the most efficient algorithms are Naive Bayes and Logistic Regression, both with linear cost relative to the number of attributes per sample ($O(d)$). Regarding memory usage, the lightest algorithms are Naive Bayes and Logistic Regression, which only require storing frequencies or a weight vector, with cost $O(dC)$ or $O(d)$.

From the auditor's perspective, the choice of the most appropriate models depends on factors such as factuality, urgency, availability of computational resources, inference time, and the type of application (streaming or batch processing). In scenarios requiring higher precision in identifying news articles with indications of irregularities, as evaluated by metrics such as *accuracy*, *precision*, *recall*, and *F1-score*, it is preferable to employ models that demonstrate stronger performance across these metrics, such as *Random Forest*

(cross-validation default), *Random Forest* (cross-validation random), and *SVC* (cross-validation random), which outperformed across all evaluation metrics of the experiment. In situations requiring greater urgency and planning agility, models with lower theoretical complexity in training, prediction, and memory usage, such as *Naive Bayes*, *Logistic Regression*, or *LightGBM*, are recommended.

For future work, aiming to optimize and automate the audit process, the following methods can be investigated:

- **Topic Modeling**: Employ techniques such as Latent Dirichlet Allocation (LDA), Non-Negative Matrix Factorization (NMF), or transformer-based models to identify latent patterns and semantic categories in the analyzed documents, enabling automatic thematic clustering of information, trend detection, and prioritization of higher-risk areas during auditing;
- **Automatic Text Summarization**: Apply text summarization methods, such as language models, to reduce information overload while maintaining the quality of summaries of the dataset used in the analytical phase, thereby reducing the intensive demand for human resources.

Acknowledgments

During the preparation of this work, the author(s) used the generative artificial intelligence tool ChatGPT [ChatGPT 2025] only for spell-checking and to review the methods in asymptotic complexity classification. After using this tool/service, the author(s) reviewed and edited the content as needed and assume full responsibility for the content of the published article.

References

Amaral, J. and Rodrigues, J. (2020). Alocação de tópicos latentes — um modelo para segmentação de dados de auditoria do governo de pe. *Revista de Engenharia e Pesquisa Aplicada*, 5(1):40–49.

Ash, E., Galletta, S., and Giommoni, T. (2020). A machine learning approach to analyzing corruption in local public finances. Working Paper 06/2020, ETH Zurich, Center for Law & Economics. Open Access. In Copyright - Non-Commercial Use Permitted.

Bannur, C., Bhat, C., Singh, K., Kulkarni, S. A., and Doddamani, M. (2023). Paacda: Comprehensive data corruption detection algorithm. *IEEE Access*, 11:24908–24934.

Basili, V. R. and Weiss, D. M. (1984). A methodology for collecting valid software engineering data. *IEEE Transactions on Software Engineering*, SE-10(6):728–738.

Benjelloun, F.-Z., Benjelloun, F.-Z., Lahcen, A. A., Lahcen, A. A., Lahcen, A. A., Belfkih, S., and Belfkih, S. (2015). An overview of big data opportunities, applications and tools. *null*.

Caputo, F., Ligorio, L., and Venturelli, A. (2025). Framing research on corruption and public administration in management studies: research trends and future directions. *Journal of Global Responsibility*.

ChatGPT (2025). Chatgpt. Disponível em: https://chat.openai.com/. Acesso em: janeiro de 2025.

Colaço Júnior, M. (2025). *IA para a Galera Toda: Agentes e Inovação Experimental Sem Código*. Amazon Publishing.

Colaço Júnior, M., Cruz, R., Araújo, L., Bliacheriene, A., and Nunes, F. (2022). Evaluation of a process for the experimental development of data mining, ai and data science applications aligned with the strategic planning. *Journal of Information Systems and Technology Management*, 19.

Damiano, R., Polizzi, S., Scannella, E., and Valenza, G. (2025). Corruption detection through textual analysis: Evidence from eurozone banks. *Business Ethics, the Environment & Responsibility*, 0:1–21. Open Access, Creative Commons Attribution License.

do Amaral, J. A. A., Amaral, J. A., Rodrigues, J. B., Rodrigues, J. B., and Rodrigues, J. B. (2020). Alocacao de topicos latentes — um modelo para segmentacao de dados de auditoria do governo de pe. *null*.

Fontes, R. S., Júnior, M. C., Prado, H., Nely, A., Araújo, J., de Paiva, J. C., and de Medeiros Valentim, R. A. (2023). Sussurro - detecção na web de eventos auditáveis que representam riscos à saúde pública. *Anais Estendidos do XXIII Simpósio Brasileiro de Computação Aplicada à Saúde (SBCAS 2023)*.

Guimarães, A., Almeida, S., Colaço Junior, M., Fontes, R., and Ferreira de Araújo, G. G. (2025). Health related news dataset.

Jiang, Y., Li, J., Wong, D., and Kan, H. Y. (2023). Natural language processing adoption in governments and future research directions: A systematic review. *Applied Sciences*.

Joudaki, H., Rashidian, A., Minaei-Bidgoli, B., Mahmoodi, M., Geraili, B., Nasiri, M., and Arab, M. (2015). Using data mining to detect health care fraud and abuse: A review of literature. *Global Journal of Health Science*, 7(1):194–202. Open Access under Creative Commons Attribution 4.0 License.

Kose, I., Gokturk, M., and Kilic, K. (2015). An interactive machine-learning-based electronic fraud and abuse detection system in healthcare insurance. *Applied Soft Computing*, 36:283–299.

Levitin, A. (2012). *Introduction to the Design & Analysis of Algorithms*. Pearson, Boston, MA, USA, 3rd edition. Includes bibliographical references and index.

Lima, M. S. M. and Delen, D. (2020). Predicting and explaining corruption across countries: A machine learning approach. *Government Information Quarterly*, 37(1):101407.

Mackey, T. K., Mackey, T. K., Vian, T., Vian, T., Köhler, J. C., and Kohler, J. C. (2018). The sustainable development goals as a framework to combat health-sector corruption. *Bulletin of The World Health Organization*.

Madureira, L., Popovič, A., and Castelli, M. (2021). Competitive intelligence: A unified view and modular definition. *Technological Forecasting & Social Change*, 173:121086. Received 22 December 2020; Received in revised form 26 July 2021; Accepted 28 July 2021; Available online 9 August 2021; ©2021 Elsevier Inc. All rights reserved.

Masrom, S., Abdul Rahman, R., Salleh, N. A., Pitaloka, E., Md Nor, M. A., and Zakaria, N. B. (2023). Machine learning prediction of petty corruption intention among law enforcement officers. *Indonesian Journal of Electrical Engineering and Computer*

Science (IJEECS), 30(3):1634–1642. Open Access, Creative Commons Attribution-ShareAlike 4.0 International License.

Paula, T. D., Amaral, A. D., Victor, A., Sales, L. A., Moreira, R., Meirelles, T., and Basso, R. (2024). Automated admissibility of complaints about fraud and corruption. In Gamallo, P., Claro, D., Teixeira, A., Real, L., Garcia, M., Oliveira, H. G., and Amaro, R., editors, *Proceedings of the 16th International Conference on Computational Processing of Portuguese - Vol. 1*, pages 610–613, Santiago de Compostela, Galicia/Spain. Association for Computational Lingustics.

Rabuzin, K. and Modrušan, N. (2019). Prediction of public procurement corruption indices using machine learning methods. In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2019) - KMIS*, pages 333–340. INSTICC, SciTePress.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Sanchez-Gomez, J. M., Vega-Rodríguez, M. A., and Pérez, C. J. (2022). A multi-objective memetic algorithm for query-oriented text summarization: Medicine texts as a case study. *Expert Systems with Applications*, 198:116769.

Schneider dos Santos, E., Machado dos Santos, M., Castro, M., et al. (2025). Detection of fraud in public procurement using data-driven methods: a systematic mapping study. *EPJ Data Science*, 14:52.

Travassos, G. H., Gurov, D., and Amaral, E. (2020). Introdução à engenharia de software. Relatório, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil. Experimental.

Vasconcelos, M. O., Chaim, R. M., and Cavique, L. (2021). Imbalanced learning in assessing the risk of corruption in public administration. In Marreiros, G., Melo, F. S., Lau, N., Lopes Cardoso, H., and Reis, L. P., editors, *Progress in Artificial Intelligence*, pages 510–523, Cham. Springer International Publishing.

Weichselbraun, A., Hörler, S., Hauser, C., and Havelka, A. (2020). Classifying news media coverage for corruption risks management with deep learning and web intelligence. In *Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics*, WIMS 2020, page 54–62, New York, NY, USA. Association for Computing Machinery.

Zhu, W., Zeng, N., and Wang, N. (2010). Sensitivity, specificity, accuracy, associated confidence interval and roc analysis with practical sas implementations. In *NorthEast SAS Users Group, Health Care and Life Sciences*.