SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 3, 422-440, 2025
DOI: 10.35377/saucis...1626239

RESEARCH ARTICLE

Investigating the Robustness of Text Mining Classification Algorithms: A Study of Algorithm and Expert Performance on Class-Inconsistent Data

Hüseyin Ataseven[1], Ömay Çokluk Bökeoğlu[2]

[1]Ankara University, Institute of Educational Sciences, Educational Measurement and Evaluation, Ankara, Türkiye, ror.org/01wntqw50
[2]Ankara University, Faculty of Educational Sciences, Department of Educational Measurement and Evaluation, Ankara, Türkiye, ror.org/01wntqw50

Corresponding author:
Hüseyin Ataseven
Ankara University,
Institute of Educational Sciences,
Educational Measurement and Evaluation,
Ankara, Türkiye
huseyinataseven@karabuk.edu.tr

Article History:
Received: 24.01.2025
Revised: 21.04.2025
Accepted: 16.07.2025
Published Online: 24.09.2025

ABSTRACT
This study compares the classification accuracy of text mining algorithms for foreign language proficiency exam items. The dataset included 2,868 items from ÜDS English tests (2006–2012) across Natural and Applied Sciences (n=956), Health Sciences (n=956), and Social Sciences (n=956). Algorithms tested were k-Nearest Neighbors (kNN), Naïve Bayes (NB), Naïve Bayes-Kernel (NB-K), Random Forest (RF), and Support Vector Machines (SVM). Binary classification accuracies ranged from 83.08% (NB) to 92.48% (SVM), while multiclass accuracies ranged from 71.93% (NB) to 84.96% (kNN). Expert analysis and cross-validation identified class-inconsistent items that negatively affected accuracy. Removing these items improved binary classification by 7.39%–9.83% and multiclass classification by 10.58%–17.89%. Among algorithms, kNN was least impacted by class-inconsistent data. These findings highlight the importance of addressing inconsistencies for improving algorithmic performance, with kNN showing robust results across scenarios.

Keywords: Text mining, Document classification, Class-inconsistent data, Robustness of classification algorithms

1. Introduction

Classification is a fundamental statistical and machine learning technique used to assign data points to predefined categories based on their characteristics. It involves the development of models that learn patterns from labeled datasets and apply this knowledge to classify new, unseen instances. The goal of classification is to maximize accuracy in predicting the correct category while minimizing errors, making it an essential tool for decision-making. From traditional methods like linear discriminant analysis and logistic regression to modern algorithms like decision trees, support vector machines, and neural networks, classification continues to evolve, addressing increasingly complex datasets [1], [2], [3]. Its applications span numerous fields, including healthcare, finance, education, and beyond, enabling the extraction of actionable insights from diverse types of data.

Early studies on classification focused on methods such as linear discriminant analysis (LDA) and logistic regression, which were primarily parametric and relied on assumptions about data distribution. LDA, introduced in the mid-20th century, was influential for its ability to separate classes using linear boundaries while maximizing variance between groups [4], [5], [6], [7]. Logistic regression provided a probabilistic framework to model binary outcomes, making it versatile for diverse applications. These methods emphasized mathematical simplicity and interpretability, setting the stage for advancements in non-parametric and machine learning-based approaches.

However, the classification process is inherently data-driven, meaning that the choice of method and its effectiveness depend heavily on the characteristics and quality of the data. Factors such as the number of features, the presence of noise, class imbalance, and the underlying distributions all influence model performance. Preprocessing steps like feature selection, normalization, and addressing missing values are crucial to ensure the data aligns with the assumptions and requirements of the chosen technique[2], [8], [9]. One particularly challenging issue is class-inconsistent data, where overlapping or ambiguous boundaries between classes can mislead the model during training. When categories are poorly defined, models struggle to distinguish between them, which in turn lowers classification accuracy and compromises overall reliability.

1.1. Text Mining

The exponential growth in textual data production has brought along various drawbacks as well as unique opportunities, which makes the ability to extract meaningful insights from this pile of unorganized data indispensable [10]. Text mining

Cite as: H. Ataseven and Ö. Çokluk Bökeoğlu. (2025). Investigating the robustness of text mining classification algorithms: A study of algorithm and expert performance on class-inconsistent data. *Sakarya University Journal of Computer and Information Sciences*. vol. 8, no. 3, pp. 422-440, 2025 doi: 10.35377/saucis...1626239

techniques make it possible to sift through this mound of unstructured text, revealing patterns, trends and precious knowledge hidden within by implementing information extraction, information retrieval, natural language processing, categorization, clustering, and classification techniques [11].

Text classification techniques, which offer a robust and systematic approach to categorizing, organizing, and understanding tremendous amounts of textual data, play a crucial role in many real-world applications. Targeted advertisements and content recommendations on the internet [12], [13], [14] , spam detection [15], [16], [17], sentiment analysis [18], [19], news categorization [20], [21], medical diagnosis [22], [23] and legal document analysis [24], [25] are just a few examples of these applications.

1.1.1. Class-Inconsistent Data

At the intersection of computers and education, these text classification techniques might be implemented using educational documents to shed light on the undiscovered patterns or to classify these documents into different labels. However, especially in text mining, where immense volumes of data are scraped using software, class-inconsistent data poses a significant challenge. The unstructured nature of textual data, combined with noise such as irrelevant information, misspellings, and ambiguous labels, makes this issue particularly pronounced. Addressing class inconsistency requires both necessary preprocessing techniques, such as tokenization, stemming, and embedding methods, and careful selection of algorithms. Among the commonly used methods, k-nearest neighbors (kNN), Naive Bayes, Naive Bayes Kernel, Random Forest, and Support Vector Machines are some of the most frequently used algorithms in the field. These algorithms, each with unique strengths, are critical in exploring strategies to improve classification performance in the presence of class-inconsistent data [1], [2], [8], [9].

The performance of classification algorithms in supervised learning largely depends on the quality and consistency of labeled data. In many real-world text mining tasks, especially those involving educational documents or domain-specific language, the labeling process can introduce inconsistencies. These inconsistencies, often referred to as label noise, class-inconsistent data, or ambiguous annotations, typically occur due to human error, subjective judgment, or unclear category boundaries. When the training data contain mislabeled instances, the models may struggle to learn accurate patterns, resulting in reduced classification performance.

The k-Nearest Neighbors (kNN) algorithm is known for its simplicity and strong performance in many text classification tasks. However, its reliance on local neighborhood information makes it particularly vulnerable to label noise. Mislabeled instances in the training data can distort the classification of nearby test samples, especially when the data is sparse or when a small value of k is used. Early work by Wilson [26] demonstrated that even a limited number of mislabeled items can negatively affect kNN's predictive accuracy. Zhu and Wu [27] further emphasized that noise in the local neighborhood can cause erroneous label propagation, particularly when the noisy instances dominate the surrounding region. Despite this sensitivity, the robustness of kNN can be improved through adjustments such as increasing the value of k or applying noise-handling techniques like instance editing or distance-weighted voting. Beretta and Santaniello [28] highlighted that imputation and density smoothing methods can help mitigate the effects of label noise, especially when combined with parameter tuning or pre-processing. These findings suggest that while kNN is inherently sensitive to class-inconsistent data, it can still perform well when supported by appropriate strategies.

Naïve Bayes classifiers are generally regarded as more resilient to label noise compared to instance-based methods. Their probabilistic foundation allows them to aggregate information across all features, which helps reduce the influence of individual mislabeled instances. This resilience is especially apparent when label noise is randomly distributed across classes. Zhang [29] pointed out that although the independence assumption of Naïve Bayes rarely holds in practice, the algorithm can still produce competitive results because its parameter estimates are relatively stable under moderate levels of noise. However, Naïve Bayes is not entirely immune to performance degradation. Frenay and Verleysen [30] observed that the algorithm becomes more vulnerable when the label noise is systematic or when it correlates with particular feature values. In such cases, the class priors and conditional probabilities can be skewed, leading to incorrect classifications. While its simplicity and efficiency make Naïve Bayes an attractive option, these characteristics also limit its ability to correct for complex noise patterns unless paired with additional filtering or preprocessing methods.

Naïve Bayes-Kernel (NB-K) extends the standard Naïve Bayes approach by applying kernel density estimation to model the feature distributions more accurately, particularly when dealing with continuous or non-normally distributed data. This added flexibility enhances classification performance in complex datasets but also introduces new sensitivities. Although direct research on the behavior of NB-K under label noise is limited, studies involving kernel-based methods suggest that mislabeling can distort the estimated densities, especially near class boundaries. Cesa-Bianchi et al. [31] found that kernel models, when exposed to label noise, often produce biased estimates that misrepresent the underlying structure of the data. These distortions can lead to unreliable classification results, particularly in high-dimensional feature spaces. To address this, researchers have proposed methods such as bandwidth adaptation and outlier filtering. Kim and Scott [32], for instance, introduced a robust kernel density estimation approach that adjusts the influence of suspected outliers and mislabeled samples. Such techniques can potentially improve NB-K's robustness, but their effectiveness in high-noise text classification scenarios remains an area that warrants further exploration.

Support Vector Machines (SVMs) are widely used for text classification due to their ability to construct optimal hyperplanes that separate classes with maximum margin. This margin-based learning helps prevent overfitting and contributes to strong generalization, especially in high-dimensional spaces. However, SVMs are not inherently robust to label noise. When mislabeling occurs near the decision boundary, it can lead to the selection of incorrect support vectors, which may distort the margin and reduce classification accuracy. Manwani and Sastry [33] demonstrated that SVMs show considerable sensitivity to noisy labels, particularly when these labels affect instances close to the margin. The misclassified support vectors can exert a disproportionate influence on the final decision boundary. To mitigate this issue, researchers have proposed robust variants of SVM that modify the loss function or apply reweighting strategies to reduce the impact of mislabeled instances. Biggio et al. [34], for example, explored ensemble-based and adversarially aware techniques that improve SVM's resilience in noisy and hostile environments. These methods offer promising results, but their computational complexity and dependence on additional tuning parameters can limit their practicality in routine text classification tasks.

Random Forest (RF) is often recognized for its natural robustness to label noise, largely due to its ensemble structure and the randomness introduced in both feature selection and data sampling. Unlike single-model classifiers, RF aggregates predictions from multiple decision trees trained on different bootstrapped subsets of the data. This aggregation dilutes the influence of mislabeled instances and helps stabilize predictions. Breiman [35] emphasized that RF's internal randomness reduces the likelihood that any single mislabeled data point will substantially influence the overall model. Empirical evidence supports this claim. Folleco et al. [36] observed that RF maintained higher classification accuracy than C4.5 and Naïve Bayes as class noise levels increased, particularly in software quality prediction tasks. More recently, Zhao et al. [37] showed that incorporating a noise transition matrix further improved RF's robustness in applications with systematic mislabeling. Additionally, Wilton et al. [38] found that RF consistently outperformed alternative classifiers in noisy settings across various domains. These results suggest that RF is a reliable choice for text classification tasks involving label uncertainty, offering a balance of accuracy and resilience without requiring extensive parameter tuning.

In this study, we tested these five most common algorithms, each of which offers a unique approach to handling classification tasks, making them widely applicable in text mining. By exploring their underlying principles and evaluating their performance, we aim to identify their strengths and weaknesses, particularly in the context of class-inconsistent data. The following section provides a short overview of text mining, document classification, examples of how each algorithm functions, and their potential for addressing classification challenges.

1.2. Classification Algorithms

The algorithms used in this study are k-Nearest Neighbors (k-NN), Naive Bayes (NB), Naive Bayes-Kernel (NB-K), Support Vector Machines (SVM), and Random Forest (RF). Although they fundamentally serve the same purpose, how they operate is quite different from one another [39]. While k-NN and SVM are based on the Vector Space Model, NB and NB-K rely on Bayesian probability principles, RF uses iterative decision trees to classify data. Below, each algorithm is presented along with its major features in brief.

1.2.1. k-Nearest Neighbors (k-NN)

K-Nearest Neighbors (k-NN) algorithm places different documents (exam items in our case) in certain distances to each other on the Vector Space Model depending on their attributes. When an unlabeled item needs to be classified, its distance to the other labelled items in the Vector Space Model is calculated, and a decision is made by using the "k" nearest neighboring items, where "k" represents the number of nearest items [40]. The working principle of k-NN algorithm classification process is presented in a three-step illustration in Figure 1.

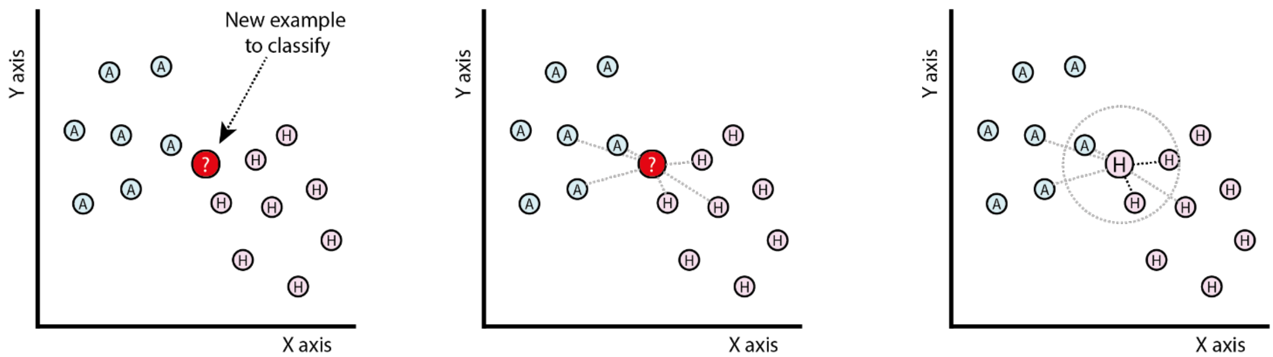

Figure 1. An example illustration of the k-NN algorithm

As seen in the third step, the unlabeled document is classified as "Class H" (k=3). Following the principle, "show me who your friends are, and I'll tell you who you are", the k-NN algorithm measures the distance to the three nearest documents and assigns the new document to the majority class. At this point, one might ask why the k value is chosen to be three instead of alternative values. For instance, the unlabeled document might be classified as "Class A" if k value is set to one. The answer lies within the training data because the decision is not made according to just one document observation. It is a question of the ideal “k” value that best fits the data, a process typically achieved through parameter optimization using computational software.

1.2.2. Naïve Bayes

Naive Bayes is a probabilistic classifier based on Bayes' theorem with the assumption of independence among predictors [41]. Despite its simplifying assumption, Naive Bayes often performs remarkably well in various applications, such as text classification and spam filtering. It calculates the probability of each class given a set of input features and selects the class with the highest probability. Naive Bayes is computationally efficient and requires a small amount of training data, making it an attractive choice for many machine learning tasks. The Naive Bayes classifier calculates the posterior probability using Bayes' theorem, as shown in Equation 1.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \tag{1}$$

1.2.3. Naïve Bayes-Kernel (NB-K)

Naive Bayes-Kernel is an extension of the traditional Naive Bayes classifier, incorporating kernel density estimation to handle continuous data more effectively. By utilizing kernels, it can estimate the probability densities of continuous features, thus relaxing the assumption of feature independence present in the standard Naive Bayes algorithm. This approach enables Naive Bayes-Kernel to handle a wider range of data types and achieve improved performance in scenarios where feature interdependencies are significant, such as in natural language processing and image recognition tasks.

Naïve Bayes- Kernel algorithm estimates the density function for each class separately using kernel density estimation. Each class's density function is calculated by placing kernels at the data points belonging to that class. The width of the kernels and the number of kernels can be adjusted to control the smoothness and accuracy of the density estimation. The number of kernels determines how finely the density function is estimated. More kernels generally result in a more accurate representation of the underlying distribution but may also lead to overfitting, especially in high-dimensional spaces. The width of the kernels (also known as the bandwidth) controls the smoothness of the estimated density function. A smaller bandwidth leads to a more peaked density estimate, capturing more details of the data distribution, but it may also result in overfitting. Conversely, a larger bandwidth results in a smoother estimate but may oversmooth the density, potentially hiding important features of the data [42].

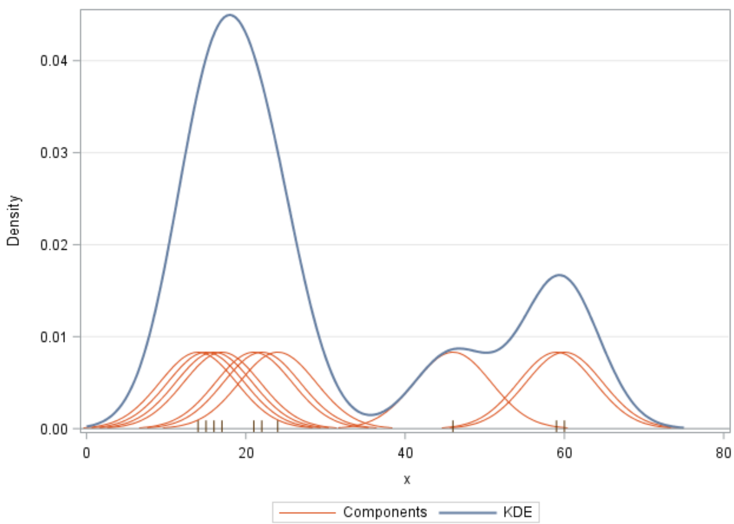

Figure 2. An example illustration of the Kernel Density Estimates
Source: https://blogs.sas.com/content/iml/2016/07/27/visualize-kernel-density-estimate.html

1.2.4. Support Vector Machines (SVM)

Support Vector Machines (SVM) are powerful supervised learning models used for classification and regression tasks. SVM aims to find the optimal hyperplane that best separates classes in high-dimensional space by maximizing the margin between data points [43]. SVMs are robust against overfitting and can handle large feature spaces efficiently, making them widely used in various fields such as text classification, image recognition, and bioinformatics. An illustration of classification process is presented in Figure 2.

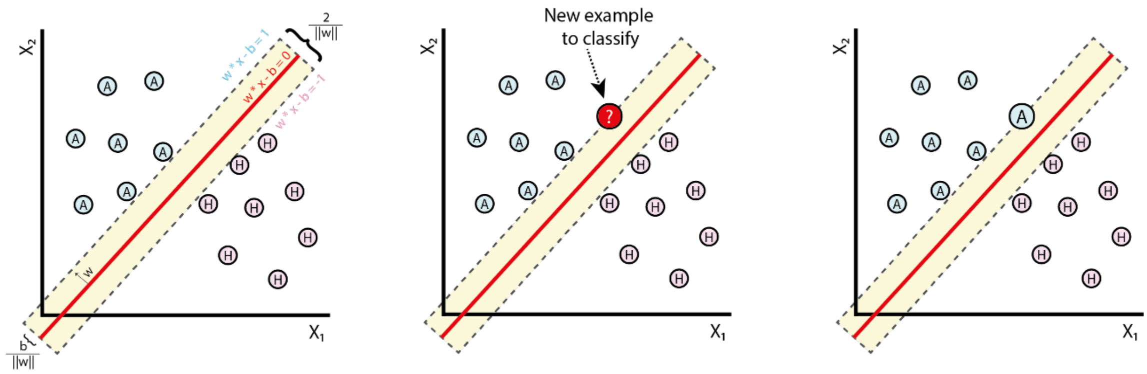

Figure 3. An example illustration of the SVM algorithm

As seen in the third step in Figure 2, the unlabeled document is classified as "Class A" because it is located on the “Class A” side of the classification line.

In SVM “w” is a vector perpendicular to the hyperplane and determines its orientation. It represents the weights assigned to each feature in the input data. The direction of “w” is crucial for defining the decision boundary. “x” represents the input data points in the feature space. Each data point is typically represented as a vector in a high-dimensional space, where each dimension corresponds to a feature. “b” is the bias term or intercept. It determines the offset of the hyperplane from the origin along the direction defined by “w”. The bias term allows the hyperplane to be translated to better separate the data points.

1.2.5. Random Forest (RF)

Random Forest (RF) is a versatile ensemble learning method used for classification and regression tasks [44]. It operates by constructing multiple decision trees during training and outputs the mode of the classes (classification) or the average prediction (regression) of the individual trees. RF reduces overfitting by combining predictions from many different trees, which makes it more reliable and less prone to variance. It is effective in handling high- dimensional data, non-linear relationships, and missing values, making it a popular choice in diverse fields including finance, healthcare, and ecology. An illustration demonstrating the working principles of Random Forest (RF) is presented in Figure 3.

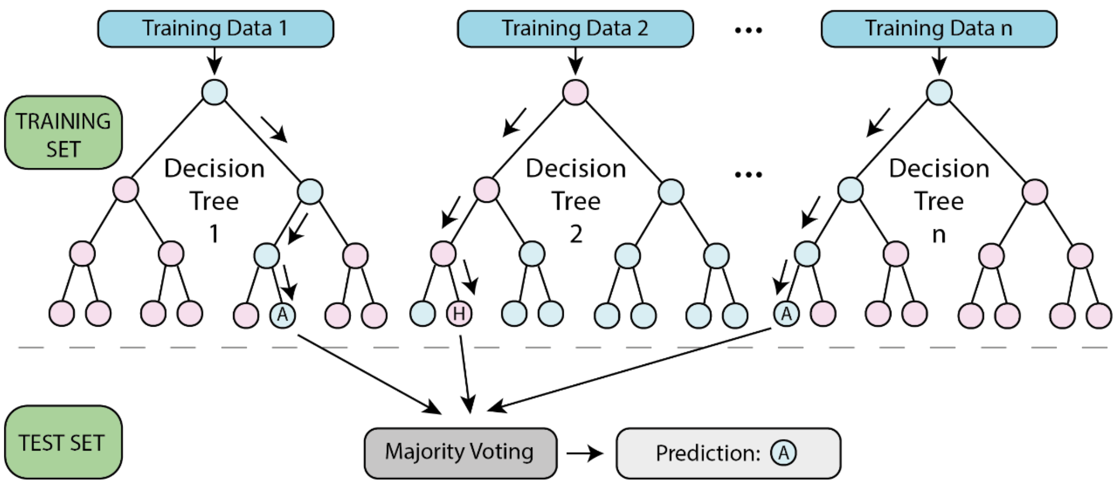

Figure 4. An example illustration of the RF algorithm

As can be seen in Figure 4, there are multiple decision trees in this algorithm. Each decision tree in the Random Forest (RF) algorithm independently classifies the training data, and the final classification is determined by a majority vote among all the trees.

Regardless of the text mining classification algorithm employed, the initial step involves transforming the raw text into an attributes matrix. This matrix comprises the frequencies of individual words present in the text. This procedure called pre-processing is presented below.

1.3. Pre-Processing Steps

Pre-processing in document classification involves several essential steps. *Tokenization* breaks text into individual words or tokens. *Case conversion* standardizes text to either lowercase or uppercase, reducing the complexity of word matching. *Stop words removal* eliminates common, non-discriminatory words like "and" or "the." *Minimum character filtering* removes tokens below a certain length threshold to focus on meaningful terms. Lastly, *stemming* reduces words to their root form, aiding in capturing semantic similarities. These steps collectively enhance the efficiency and accuracy of document classification algorithms by simplifying text representation and removing noise, thereby facilitating more effective pattern recognition and categorization [45]. An example of pre-processing steps is presented in Figure 5.

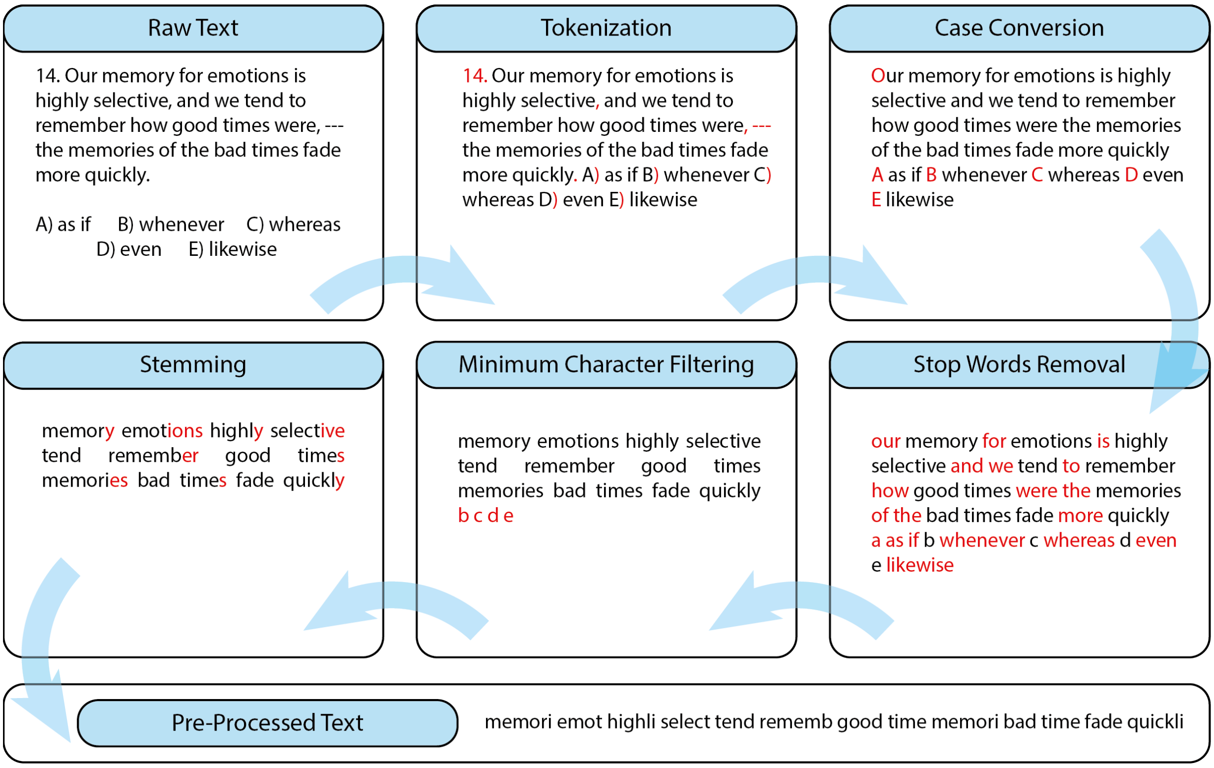

Figure 5. Pre-Processing Steps of Textual Classification

As can be seen in Figure 5, the first step in text pre-processing involves starting with the raw text, which contains all the original characters, punctuation, and potentially irrelevant elements. This serves as the initial input for the subsequent transformations. In the next step, tokenization is applied, where the text is split into individual tokens or words. During this process, punctuation marks such as hyphens and commas are removed, making the text easier to analyze at the word level.

After tokenization, case conversion is performed to standardize all tokens into lowercase. This eliminates inconsistencies caused by capitalization and ensures uniformity throughout the text. Following this, stop words are removed. These are commonly used words such as "and", "for", and "how", which do not carry significant meaning and are typically excluded to focus on more relevant terms.

Stemming is then applied to reduce words to their root forms, ensuring variations of the same word are grouped together. For example, both “memory” and "memories" are reduced to "memori", and "selective" becomes "select". This step helps simplify the text and reduces its dimensionality by omitting word formation suffixes, ensuring that all variations of a word are represented by a single common base form. Finally, minimum character filtering is performed, where very short tokens, such as single letters, are removed. These tokens often lack meaningful context and contribute little to the analysis. After all these steps, the text is cleaned, standardized, and ready for use in classification or other analytical tasks, ensuring that only the most informative features are retained. Then, a feature matrix is created for all the documents, allowing the entire textual dataset to be represented numerically in a unified structure.

1.4. Indicators of Classification Performance

Indicators of classification performance serve as crucial metrics for evaluating the effectiveness of machine learning models typically demonstrated with a *confusion matrix*. Common indicators include *accuracy*, which measures the proportion of correctly classified instances, *precision*, quantifying the ratio of true positives to all predicted positives, *recall*, indicating the ratio of true positives to all actual positives, and *F1-score*, which harmonizes precision and recall. These indicators provide comprehensive insights into the model's ability to correctly classify instances across different classes, guiding further refinement and optimization efforts [46].

1.4.1. Confusion Matrices

The distribution of errors among various classes or categories in a classification process is handled using numerical structures called confusion matrices, which were originally named contingency tables [47]. Confusion matrices display a cross-tabulation of true and predicted values of the classification, capturing the number of correctly and incorrectly classified items and the type of error. A sample confusion matrix of a binomial classification between Class A and Class H is presented in Table 1.

Table 1. Example Confusion Matrix

		TRUE		TOTAL
		A	H	
PREDICTED	A	125 (TP)	18 (FP)	143
	H	8 (FN)	115 (TN)	123
TOTAL		133	133	266

TP: True Positive, FP: False Positive, FN: False Negative, TN: True Negative

As seen in Table 1, in classification tasks, the terms true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) are used to evaluate the performance of a model. 125 items are classified correctly as Class A, and 115 items are classified correctly as Class H. However, 8 Class A items are misclassified as Class H, and 18 Class H items are misclassified as Class A.

1.4.2. Accuracy

Classification accuracy is a very general and straightforward metric reflecting the overall success of the classifiers. In essence, it is the ratio of the true positives (TP) and true negatives (TN) to the total number of items. The equation to calculate accuracy is presented in Equation 2.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

Based on the instances in Table 1, the accuracy value is calculated as (125+115) / 266 = 0.902. However, this overall metric doesn't reflect the false positive (FP) and false negative (FN) values. The distribution of these values on a confusion matrix might be equal between classes or skewed towards a specific class, increasing or decreasing Type 1 or Type 2 error. The indicators dealing with this situation are presented below.

1.4.3. Precision and Recall

Precision and recall are important metrics in text classification, assessing the effectiveness of the predictions of classifiers. Precision measures the accuracy of the positive predictions, which is essentially the ratio of true positive (TP) predictions to the total positive predictions made. Precision emphasizes the relevance of correctly identified instances. On the other hand, recall measures the ability of the model to identify all actual positives by calculating the ratio of true positive (TP) predictions to the total actual positives (TP and FN). Recall evaluates the classifier's ability to identify all relevant instances within the dataset. Achieving a balance between precision and recall ensures a classifier effectively identifies relevant information while minimizing false positives and negatives, optimizing classification performance. The equations to calculate precision and recall are presented in Equations 3 and 4.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

Based on the instances in Table 1, the precision value is calculated as 115/143= 0.874, and the recall value is calculated as 125/133= 0.940.

1.4.4. F1 Score

Another indicator used to evaluate the performance of a classification model is the F1 score. It is obtained by calculating the harmonic mean of precision and recall values. By balancing the trade-off between these two metrics, the F1 score provides a unified assessment of a classifier's ability to accurately identify positive instances. This value tends to be at the same level as the accuracy metric if the classifier has balanced FP and FN values. However, F1 score might be considered as a better alternative to accuracy while comparing imbalanced misclassifications. The equation to calculate F1 score is presented in Equation 5.

$$F1\,Score = 2.\left(\frac{precision\ x\ recall}{precision + recall}\right) \tag{5}$$

Based on the instances in Table 1, the F1 score is calculated as 2*((0.874*0.940)/(0.874+0.940)) = 0.906, which is very close to the previously calculated accuracy metric (0.902), indicating to a balanced classification performance.

1.4.5. Area Under the Curve – Receiver Operating Characteristic (AUC-ROC)

In binary classification tasks, the Area Under the Curve – Receiver Operating Characteristic (AUC-ROC) serves as an important performance metric. The ROC curve shows the relationship between the true positive rate and the false positive rate across different classification thresholds. The AUC value summarizes the overall ability of a model to distinguish between classes, with values closer to 1 indicating stronger performance. Unlike accuracy, which depends on a fixed threshold and may be affected by class imbalance, AUC-ROC provides a more robust and threshold-independent evaluation. Therefore, it is widely used to compare the effectiveness of classification algorithms, especially in studies involving imbalanced or domain-specific data [48].

1.5 Motivation of the Study and Research Questions

The motivation behind this study stems from the increasing need for efficient and accurate text classification algorithms in educational settings, particularly in the educational measurement domain for the classification of exam items. As online learning platforms gain prominence and educational materials become increasingly digitalized, there is a growing demand for automated systems capable of categorizing exam items quickly and reliably. A significant challenge in this context is the presence of class-inconsistent data, which can lead to misclassification and reduced performance of classification models. By comparing text classification algorithms (kNN, NB, NB-K, SVM, and RF), this study seeks to identify the most robust approach for handling class-inconsistent data, ensuring more accurate and reliable classification of exam items. Addressing this challenge is expected to enhance exam creation, grading, and overall educational measurement processes, contributing to improved educational outcomes and student learning experiences. With this motivation, the study aims to answer the following questions:

1- How do the selected algorithms perform in classifying the initial set of exam items?

2- How effectively do human experts classify exam items that contain class-inconsistencies?

3- How do the algorithms perform when class-inconsistent items are excluded from the dataset?

These questions aim to provide a comprehensive understanding of algorithmic and human performance in the context of text classification, particularly under the challenges posed by class-inconsistent data.

2. Method

The section focuses on the documents and data analysis conducted in this study. It begins by describing the characteristics and composition of the dataset. Then, it gives information about how the data was analyzed.

2.1. Documents

In Türkiye, ÖSYM (The Measuring, Selection and Placement Centre) is responsible for conducting national level university entrance examinations as well as several other large scale examinations including language proficiency exams for higher education students and graduates aspiring to pursue careers as academicians. ÜDS (Interuniversity Council Foreign Language Exam) was a former exam until 2012 and it was later replaced by a similar exam named as YÖKDİL (Higher Education

Institutions Foreign Language Exam). Due to the fact that the recent exam items aren't shared with the public in accordance with the new ÖSYM policies, it isn't possible to use YÖKDİL items in this study. The next best alternative is to use publicly shared ÜDS items (N=2868) from 42 exams conducted by ÖSYM between the years 2006 and 2012. ÜDS, having three domain categories as applied sciences, health sciences and social sciences, might be considered as the perfect choice for this document classification study.

2.2. Data Analysis

Previously conducted ÜDS exams are available for download on ÖSYM website (www.osym.gov.tr). All 42 exam PDFs were downloaded and transformed into a single dataset consisting of exam items and their domain category labels (applied sciences, health sciences and social sciences). Then, the items were classified into two and three categories using text mining classification algorithms on RapidMiner software. RapidMiner is a popular data mining software developed in 2001 at the University of Dortmund by Ingo Mierswa, Ralf Klinkenber and Simon Fisher under the name YALE and renamed as RapidMiner as of 2013. It is an effective data mining and text mining tool where many operations such as data pre-processing, model building, and prediction can be performed with its user-friendly interface (https://altair.com/altair-rapidminer). The flowchart regarding binomial and multinomial classification process has been presented in Figure 6.

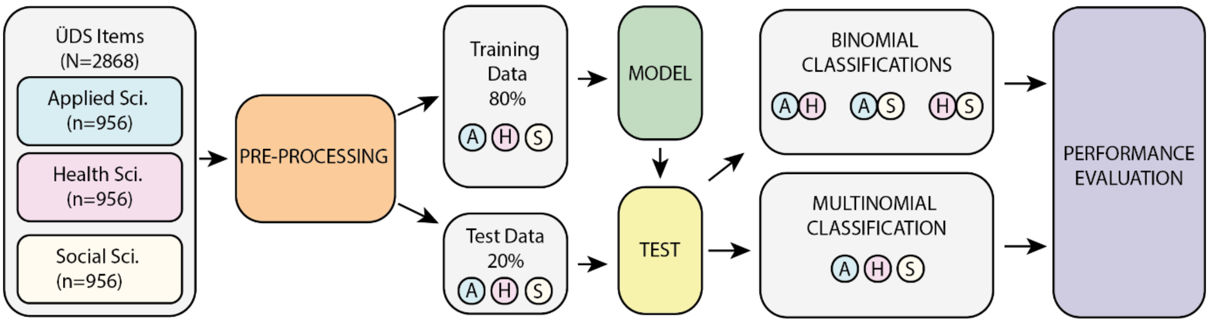

Figure 6. Binomial and Multinomial Classification Flowchart

As can be seen in Figure 6, ÜDS consists of English proficiency items (N=2868) in the field of applied sciences, health sciences and social sciences. Each domain has an equal number of items (n=956), which were randomly selected to be used as the training data (80%) and the test data (20%) after the pre-processing step. Then classification models were created using five text mining algorithms, which are k-Nearest Neighbors (k-NN), Naïve Bayes (NB), Naïve Bayes-Kernel (NB-K), Support Vector Machines (SVM), and Random Forest (RF). In the next step, binomial classifications between A-H, A-S, and H-S domain pairs, and multinomial classifications among A-H-S domains were tested, and the classification performance of the five algorithms were evaluated.

2.2.1. Pre-Processing

All data preprocessing procedures in this study were carried out using RapidMiner, a visual data science platform that does not require coding. The preprocessing workflow included the following sequential steps: (1) Tokenization, where the text data was divided into individual word units; (2) Case conversion, standardizing all text to lowercase to prevent case-related mismatches; (3) Stemming, which involved reducing words to their root forms to group inflected versions of the same term; (4) Minimum character filtering, where tokens shorter than two characters were removed; and (5) Stop-word removal, where frequently occurring but semantically weak words were excluded. These steps were applied consistently across all documents using RapidMiner's operator-based, drag-and-drop interface, ensuring a transparent and reproducible preprocessing flow.

2.2.2. Parameter Optimizations

To ensure optimal model performance and enhance reproducibility, key parameters for each algorithm were tuned using a 10-fold cross-validation process. For the k-Nearest Neighbors (kNN) algorithm, the optimal value of k was determined through iterative testing. In the Naïve Bayes Kernel (NB-K) algorithm, the number of kernels and the minimum bandwidth parameter were optimized. For the Random Forest (RF) model, hyperparameters including the number of trees (n_estimators) and the maximum depth (max_depth) were adjusted to achieve the best performance. Similarly, for Support Vector Machines (SVM), the convergence epsilon parameter and the appropriate kernel function were selected based on cross-validation results. This parameter optimization process aimed to maximize classification accuracy while maintaining model robustness.

The specific parameter values used in the classification processes for each algorithm and classification pair are as follows: For the k-Nearest Neighbors (kNN) algorithm, the optimal number of neighbors (k) was set as 96 for the A-H classification, 66 for A-S, 80 for H-S, and 85 for the A-H-S classification. For the Naïve Bayes Kernel (NB-K) algorithm, the number of kernels and minimum bandwidth parameters were respectively: 49 and 0.7 for A-H, 1 and 0.3 for A-S, 1 and 0.2 for H-S, and

1 and 0.2 for A-H-S. In the case of Random Forest (RF), the number of trees and maximum depth were optimized to 80 and 60 for A-H, 100 and 80 for A-S, 100 and 70 for H-S, and 80 and 60 for A-H-S. Finally, the Support Vector Machine (SVM) algorithm was run with optimized convergence epsilon values of 0.051 for A-H, 0.036 for A-S, 0.130 for H-S, and 0.008 for A-H-S

The time required to complete the parameter optimization process for each algorithm was also recorded and is presented in the relevant tables. For binary classifications, the average optimization durations were 29 minutes 17 seconds for kNN, 47 minutes 14 seconds for NB-K, 1 hour 5 minutes 39 seconds for Random Forest, and 7 minutes 1 second for SVM. Naïve Bayes requires no optimization time, as it operates without tunable parameters. For the multiclass classification, optimization times increased due to added complexity, with durations of 1 hour 18 minutes 45 seconds for kNN, 1 hour 49 minutes 52 seconds for NB-K, 5 hours 52 minutes 12 seconds for Random Forest, and 21 minutes 42 seconds for SVM. These differences reflect the varying computational demands of each algorithm. The total optimization time across all algorithms and tasks amounted to approximately 15 hours and 12 minutes.

2.2.3. Evaluation Metrics

To evaluate the classification performance of the algorithms, five commonly used metrics were employed: accuracy, precision, recall, F1-score, and AUC-ROC. These metrics were selected because they provide complementary insights into model performance. Accuracy measures the overall proportion of correctly classified items, while precision (the ratio of true positives to predicted positives) and recall (the ratio of true positives to actual positives) offer a more detailed understanding of performance at the class level. The F1-score, calculated as the harmonic mean of precision and recall, is particularly useful when a balance between these two measures is desired. In addition, AUC-ROC (Area Under the Curve – Receiver Operating Characteristic) provides a threshold-independent evaluation by measuring the model's ability to distinguish between classes across all decision thresholds. It is especially valuable in binary classification tasks and when class distributions are imbalanced. All metrics were computed based on the confusion matrices and probability scores generated by the classification models within RapidMiner.

In terms of parameter tuning, a 10-fold cross-validation method was used to optimize the hyperparameters of each algorithm. While RapidMiner does not explicitly refer to this process as Grid Search or Random Search, the platform enables users to define a range of values for each parameter, which are then systematically tested through looping structures. The combination yielding the best cross-validated performance was selected as optimal. This approach ensured both robustness and reproducibility in model training.

2.2.4. Significance of Difference

In this study, the significance of the difference in classification accuracy values obtained before and after the removal of class-inconsistent items was determined using the test of significance for the difference between two independent proportions [49]. This test is particularly appropriate because classification accuracy is expressed as a proportion (i.e., the ratio of correctly classified items to the total number of items), and the comparison involves two independent groups: one with class-inconsistent items and one without. Since the data points used in these two conditions differ, and accuracy is not a continuous variable but a derived proportion, traditional tests such as the paired t-test or ANOVA are not applicable. Moreover, the structure of classification accuracy does not allow for assumptions of normality or matched samples, further justifying the choice of this proportion-based test. Therefore, this method provides the most statistically valid approach to determining whether the observed improvements in accuracy are significant following the data refinement process. The equation for this test is provided in Equation 6.

$$\sigma_{\mathrm{P1\text{-}P2}} = \sqrt{\frac{P_1 Q_1}{N_1} + \frac{P_2 Q_2}{N_2}} \tag{6}$$

Equation 6 represents the standard error formula used to calculate the variability (σ) of the difference between two independent proportions, denoted as $P_1 - P_2$. In this equation, P_1 and P_2 refer to the proportions of correct classifications (i.e., accuracy) in the first and second datasets, respectively, while Q_1 and Q_2 are their complements (Q = 1 – P). N_1 and N_2 indicate the total number of items in each dataset. The equation computes the standard error of the difference between the two independent proportions, allowing to determine whether the observed difference in classification accuracy is statistically significant.

3. Results

3.1. Algorithm Performance on Initial Exam Items

The binomial classification performance metrics, including accuracy, precision, recall, and F1 score, for five machine learning algorithms—k-Nearest Neighbors (kNN), Naïve Bayes (NB), Naïve Bayes Kernel (NB-K), Random Forest (RF), and Support Vector Machine (SVM) applied to Applied Sciences and Health Sciences items are presented in Table 2.

Table 2. Binomial Classification Performance of the Algorithms (Applied Sciences – Health Sciences)

Domain	Classification	kNN	NB	NB-K	RF	SVM
	Accuracy (%)	90.23	83.08	**90.60**	86.84	86.84
Applied Sciences Items (n=133)	Precision (%)	**94.21**	81.43	93.55	84.51	87.12
	Recall (%)	85.71	85.71	87.22	**90.23**	86.47
	F1 Score (%)	89.76	83.52	**90.27**	87.27	86.79
Health Sciences Items (n=133)	Precision (%)	86.90	84.92	88.03	**89.52**	86.57
	Recall (%)	**94.74**	80.45	93.98	83.46	87.22
	F1 Score (%)	90.65	82.62	**90.91**	86.38	86.89

In evaluating the performance of five classification algorithms—k-Nearest Neighbors (kNN), Naïve Bayes (NB), Naïve Bayes Kernel (NB-K), Random Forest (RF), and Support Vector Machine (SVM)—across Applied Sciences and Health Sciences items, notable differences emerged in their effectiveness. Overall, NB-K and kNN demonstrated the highest accuracy (90.60% and 90.23%, respectively), making them the most reliable methods for this dataset. In the Applied Sciences domain, NB-K achieved the best F1 score (90.27%), indicating a strong balance between precision (93.55%) and recall (87.22%), while kNN followed closely with comparable metrics. Conversely, NB showed the weakest performance across all metrics, with an accuracy of 83.08% and an F1 score of 83.52% in this domain, likely due to its strong independence assumptions. In the Health Sciences domain, NB-K and kNN continued to perform robustly, with NB-K achieving the highest F1 score (90.91%) and kNN excelling in recall (94.74%). RF and SVM offered moderate performance across both domains, with RF excelling in precision for Health Sciences (89.52%) and recall for Applied Sciences (90.23%). The findings indicate that NB-K and kNN perform best in balancing precision and recall, whereas RF and SVM remain viable options with consistent, though slightly lower, performance. In contrast, NB may not be suitable for datasets with complex feature interdependencies. Figure 6 presents the ROC curves obtained from the binary classification of Applied Sciences and Health Sciences items, illustrating the true positive rate against the false positive rate for each algorithm.

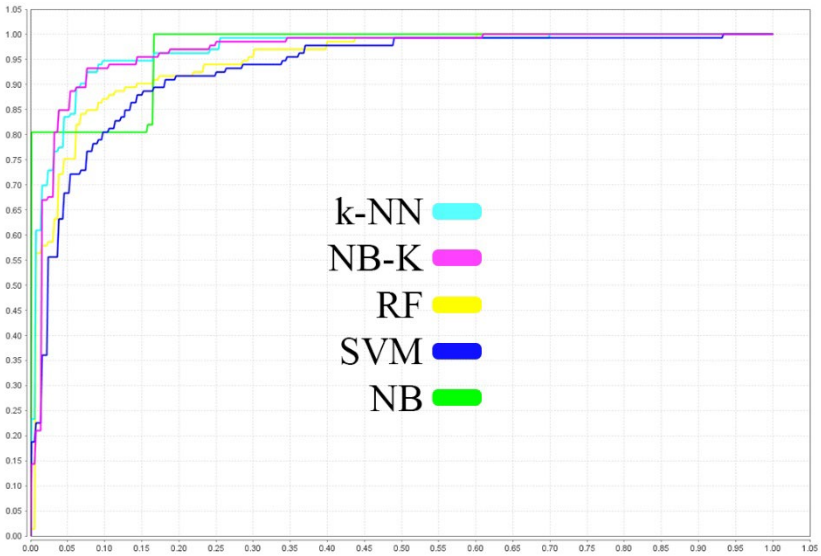

Figure 6. ROC Curves for Binomial Classification (Applied Sciences – Health Sciences)

As shown in Figure 6, the ROC curves for the binary classification between Applied Sciences and Health Sciences items demonstrate the discriminatory power of each algorithm. Based on the Area Under the Curve (AUC) values, which indicate overall classification performance regardless of decision thresholds, the ranking of algorithms from highest to lowest performance is as follows: kNN (AUC = 0.968), NB-K (AUC = 0.965), Random Forest (AUC = 0.951), SVM (AUC = 0.931), and Naïve Bayes (AUC = 0.778). These results confirm that kNN and NB-K exhibited superior performance in distinguishing between the two domains, while Naïve Bayes lagged behind in comparison.

The binomial classification performance metrics, including accuracy, precision, recall, and F1 score, for five machine learning algorithms—k-Nearest Neighbors (kNN), Naïve Bayes (NB), Naïve Bayes Kernel (NB-K), Random Forest (RF), and Support Vector Machine (SVM)—applied to Applied Sciences and Social Sciences items are presented in Table 3.

Table 3. Binomial Classification Performance of the Algorithms (Applied Sciences – Social Sciences)

Domain	Classification	kNN	NB	NB-K	RF	SVM
	Accuracy (%)	**89.10**	78.20	88.35	84.59	85.34
Applied Sciences Items (n=133)	Precision (%)	86.11	77.78	**91.80**	85.94	84.06
	Recall (%)	**93.23**	78.95	84.21	82.71	87.22
	F1 Score (%)	**89.53**	78.36	87.84	84.29	85.61
Social Sciences Items (n=133)	Precision (%)	**92.62**	78.63	85.42	83.33	86.72
	Recall (%)	84.96	77.44	**92.48**	86.47	83.46
	F1 Score (%)	88.62	78.03	**88.81**	84.87	85.06

The classification performance metrics indicate that NB-K and kNN consistently outperform other algorithms across both Applied Sciences and Social Sciences items, achieving high accuracy, precision, recall, and F1 scores. NB-K particularly excels in precision for Applied Sciences (91.80%) and recall for Social Sciences (92.48%), while kNN demonstrates balanced performance with the highest recall in Applied Sciences (93.23%) and precision in Social Sciences (92.62%). RF and SVM offer moderate performance, with RF showing strong recall in Social Sciences (86.47%) and SVM maintaining stable metrics across both domains. NB, however, lags significantly, with the lowest overall accuracy (78.20%) and weaker precision, recall, and F1 scores in both domains, reflecting its struggles with feature dependencies in the dataset. These results highlight NB-K and kNN as the most suitable algorithms for tasks requiring a balance between precision and recall, while RF and SVM remain viable alternatives for general classification tasks. Figure 7 displays the ROC curves for the binary classification between Applied Sciences and Social Sciences items, illustrating the performance of each algorithm in distinguishing between the two domains.

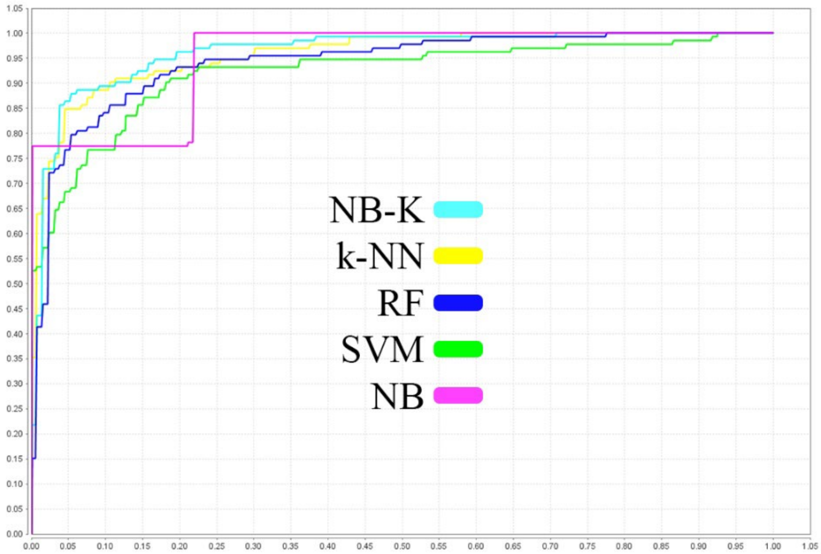

Figure 7. ROC Curves for Binomial Classification (Applied Sciences – Social Sciences)

As shown in Figure 7, the ROC curves illustrate the binary classification performance between Applied Sciences and Social Sciences items. Based on the Area Under the Curve (AUC) values, which provide a threshold-independent measure of classification accuracy, the algorithms ranked in performance as follows: NB-K (AUC = 0.962), kNN (AUC = 0.958), Random Forest (AUC = 0.933), SVM (AUC = 0.921), and Naïve Bayes (AUC = 0.703). These results indicate that NB-K and kNN were the most effective in distinguishing between the two domains, while Naïve Bayes again demonstrated relatively weaker performance.

The binomial classification performance metrics, including accuracy, precision, recall, and F1 score, for five machine learning algorithms—k-Nearest Neighbors (kNN), Naïve Bayes (NB), Naïve Bayes Kernel (NB-K), Random Forest (RF), and Support Vector Machine (SVM)—applied to Health Sciences and Social Sciences items are presented in Table 4.

Table 4. Binomial Classification Performance of the Algorithms (Health Sciences – Social Sciences)

Domain	Classification	kNN	NB	NB-K	RF	SVM
	Accuracy (%)	91.35	87.22	86.47	89.10	**92.48**
Health Sciences Items (n=133)	Precision (%)	89.29	87.22	95.33	**95.61**	93.13
	Recall (%)	**93.98**	87.22	76.69	81.95	91.73
	F1 Score (%)	91.58	87.22	85.00	88.25	**94.42**
Social Sciences Items (n=133)	Precision (%)	**93.65**	87.22	80.50	84.21	91.85
	Recall (%)	88.72	87.22	**96.24**	**96.24**	93.23
	F1 Score (%)	91.12	87.22	87.67	89.82	**92.53**

The classification metrics show that SVM achieves the highest overall accuracy (92.48%), followed closely by kNN (91.35%), highlighting their robustness across both domains. For Health Sciences, SVM demonstrates the best balance between precision (93.13%) and recall (91.73%), resulting in the highest F1 score (94.42%). RF also performs well in precision (95.61%) but shows lower recall (81.95%), slightly limiting its F1 score (88.25%). NB-K exhibits the highest precision (95.33%) but significantly lower recall (76.69%), leading to an F1 score (85.00%) that reflects its imbalance. In Social Sciences, RF and SVM excel in recall (96.24% and 93.23%, respectively), with SVM achieving the highest F1 score (92.53%). While kNN offers competitive precision (93.65%) and recall (88.72%), NB consistently underperforms across all metrics in both domains, with an accuracy of 87.22% and uniform scores for precision, recall, and F1, indicating potential limitations with its application to these datasets. Figure 8 presents the ROC curves for the binary classification between Health Sciences and Social Sciences items, illustrating the classification performance of each algorithm based on their ability to distinguish between these two domains.

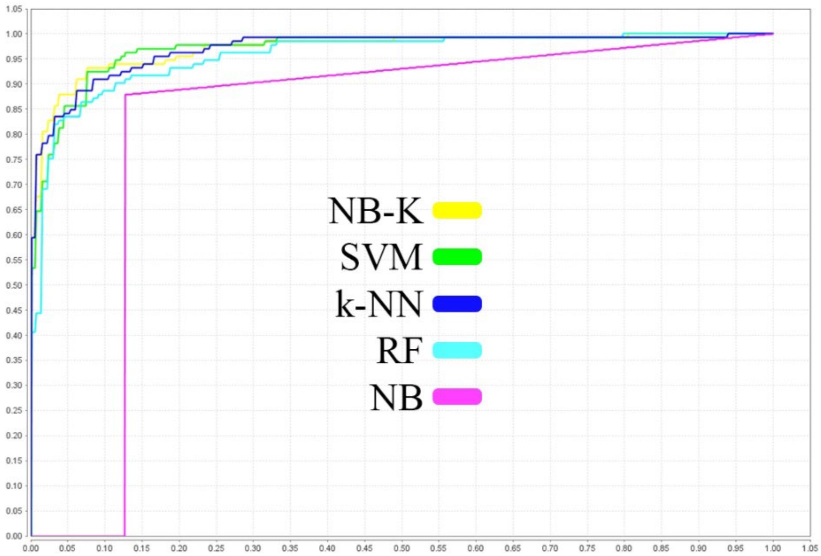

Figure 8. ROC Curves for Binomial Classification (Health Sciences – Social Sciences)

As shown in Figure 8, the ROC curves represent the binary classification performance between Health Sciences and Social Sciences items. According to the Area Under the Curve (AUC) values, which reflect the models' ability to distinguish between the two domains regardless of classification threshold, the ranking of algorithm performance is as follows: NB-K (AUC = 0.971), SVM (AUC = 0.968), kNN (AUC = 0.967), Random Forest (AUC = 0.961), and Naïve Bayes (AUC = 0.819). These results indicate that all models performed well, with NB-K achieving the highest classification capability for this particular binary task.

Overall, SVM and kNN emerge as the most effective algorithms, particularly for tasks requiring a balance of precision and recall, while NB-K and RF show strengths in precision but struggle with recall.

The multinomial classification performance metrics, including accuracy, precision, recall, and F1 score, for five machine learning algorithms—k-Nearest Neighbors (kNN), Naïve Bayes (NB), Naïve Bayes Kernel (NB-K), Random Forest (RF), and Support Vector Machine (SVM)—applied to Applied Sciences, Health Sciences, and Social Sciences items are presented in Table 5.

Table 5. Multinomial Classification Performance of the Algorithms (Applied Sciences -Health Sciences – Social Sciences)

Domain	Classification	kNN	NB	NB-K	RF	SVM
	Accuracy (%)	**84.96**	71.93	83.96	80.45	78.70
Applied Sciences Items (n=133)	Precision (%)	81.88	**95.49**	83.46	73.86	73.38
	Recall (%)	**84.96**	69.92	79.70	**84.96**	76.69
	F1 Score (%)	**83.39**	80.73	81.54	79.02	75.00
Health Sciences Items (n=133)	Precision (%)	83.45	79.39	**90.91**	85.25	82.61
	Recall (%)	**90.98**	78.20	82.71	78.20	85.71
	F1 Score (%)	**87.05**	78.79	86.62	81.57	84.13
Social Sciences Items (n=133)	Precision (%)	**90.52**	71.43	78.81	83.87	80.33
	Recall (%)	78.95	67.67	**89.47**	78.20	73.68
	F1 Score (%)	**84.34**	69.50	83.80	80.94	76.86

The classification results reveal that kNN and NB-K outperform other algorithms in terms of accuracy, with kNN achieving 84.96% and NB-K close behind at 83.96%. For Applied Sciences items, kNN demonstrates balanced performance across metrics, with an F1 score of 83.39%, while NB-K (81.54%) shows slightly lower F1 due to its reduced recall (79.70%). NB stands out for its high precision (95.49%) in this domain, but its overall effectiveness is limited by the lowest recall (69.92%), resulting in a moderate F1 score (80.73%). RF and SVM struggle with lower precision and recall, reflected in their F1 scores of 79.02% and 75.00%, respectively.

For Health Sciences items, NB-K achieves the highest F1 score (86.62%), driven by its strong precision (90.91%) and recall (82.71%). kNN follows closely with an F1 score of 87.05%, balancing high recall (90.98%) with slightly lower precision (83.45%). RF and SVM offer competitive performance, with F1 scores of 81.57% and 84.13%, respectively, while NB remains the weakest performer with an F1 score of 78.79%.

In Social Sciences, NB-K exhibits strong recall (89.47%) and achieves the highest F1 score (83.80%). However, kNN leads in precision (90.52%) while maintaining a respectable F1 score (84.34%), reflecting its reliability in identifying true positives. RF and SVM show moderate performance, with F1 scores of 80.94% and 76.86%, respectively, while NB once again underperforms with the lowest F1 score (69.50%), primarily due to its poor recall (67.67%).

Overall, kNN and NB-K emerge as the most effective algorithms, particularly for tasks requiring a balance of precision and recall. NB demonstrates strengths in precision but struggles with recall across all domains, limiting its overall effectiveness. RF and SVM provide moderate performance but are less consistent compared to kNN and NB-K.

3.2. Expert Classification Performance with Class-Inconsistent Items

After the inspection of the classification results, it was found that 51 items were consistently misclassified by three or more algorithms. The information about these items is presented in Table 6.

Table 6. Class-Inconsistent Item Exploration Results

	n	Item Number
Consistently Misclassified Items (Binomial Classifications)	12	214, 473, 483, 684, 931, 1491, 1566,1890, 2046, 2118, 2333, 3110
Consistently Misclassified Items (Multinomial Classifications)	22	27, 185, 492, 533, 572, 607, 652, 728, 773, 805, 810, 1334, 1501, 1730, 1765, 1774, 2032, 2045, 2112, 2171, 3030, 3268
Consistently Misclassified Items (Both Binomial and Multinomial Classifications)	17	249, 507, 523, 545, 676, 1214, 1284, 1444, 1470, 1731, 1776, 1873, 2016, 2092, 2249, 2467, 2629

These items were consistently misclassified in binomial classifications (n=12), in multinomial classifications (n=17), and both (n=22). In order to take a deeper look at these items, an expert opinion form was created. In this form, along with the consistently misclassified items (n=51), another set of correctly classified items (n=51) were chosen. The experts must read these items (N=102) and guess their domains just as the algorithms in the multinomial classification. Also, they must underline the keyword(s) which helped with their decision. The aim of this form was (1) to discriminate the reliable and accurate experts using 51 non-problematic items, (2) to use the judgement of these reliable and accurate experts on 51 problematic items, (3) to determine the keywords underlined by the experts.

In the next step, 13 English Language Teaching experts were asked to fill in the form. Their classification indicators related to 51 non-problematic items are presented in Table 7.

Table 7. Classification Performance of the Experts on 51 Non-problematic Items for Validation

Expert	Accuracy	F1 Score (Applied Sci.)	F1 Score (Health Sci.)	F1 Score (Social Sci.)	Mean F1 Score
E1	62.75%	25.01%	94.11%	59.10%	59.41%
E2	64.71%	42.86%	82.76%	66.67%	64.10%
E3	64.71%	25.00%	93.75%	65.22%	61.32%
E4	**94.12%**	**91.43%**	**100%**	**90.91%**	**94.22%**
E5	**92.16%**	**91.89%**	**97.14%**	**86.67%**	**91.90%**
E6	**88.24%**	**81.25%**	**100%**	**83.33%**	**88.19%**
E7	74.51%	68.57%	78.57%	76.92%	74.69%
E8	**86.27%**	**82.35%**	**93.75%**	**83.33%**	**86.48%**
E9	**92.16%**	**91.43%**	**97.14%**	**87.50%**	**92.02%**
E10	74.51%	%72.22	%71.43	%78.95	%74.20
E11	**84.31%**	**%81.08**	**%88.24**	**%83.87**	**%84.40**
E12	76.47%	%53.84	%97.14	%73.17	%74.72
E13	**86.27%**	**%82.35**	**%96.97**	**%80.00**	**%86.44**

Among these experts, seven of them (E4, E5, E6, E8, E9, E11, E13) were found to be reliable and accurate enough to proceed with the next step. Their classification indicators related to 51 consistently misclassified items are presented in Table 8.

Table 8. Classification Performance of the Experts on 51 Problematic Items

Expert	Accuracy	F1 Score (Applied Sci.)	F1 Score (Health Sci.)	F1 Score (Social Sci.)	Mean F1 Score
E4	50.98%	55.00%	53.33%	43.75%	50.69%
E5	52.92%	54.06%	63.16%	37.04%	51.42%
E6	49.02%	40.00%	61.54%	42.43%	47.99%
E8	47.06%	38.71%	50.00%	51.43%	46.71%
E9	49.02%	48.78%	59.46%	33.33%	47.19%
E11	45.10%	26.66%	61.90%	40.00%	42.85%
E13	39.22%	22.22%	41.02%	50.00%	37.75%

As can be seen in Table 8, even the most successful experts have difficulty in classifying these 51 problematic items correctly. This demonstrates that the low classification metrics are not stemming from the classifier but the data. After carefully inspecting these items one by one, it was decided that 28 items can be considered class-inconsistent.

Table 9. Significance of Differences in Binomial Classification Performance Before and After Removing Class-Inconsistent Items

		kNN	NB	NB-K	RF	SVM
Applied Sciences and Health Sciences Items	Before (n=266)	90.23%	83.08%	90.60%	86.84%	86.84%
	After (n=244)	98.66%	94.20%	98.66%	95.54%	97.32%
	Difference in %	8.43%	11.12%	8.06%	8.70%	10.85%
	Critical Value (p<.05)	3.22%	4.50%	3.17%	4.03%	3.80%
	Significance	Yes	Yes	Yes	Yes	Yes
Applied Sciences and Social Sciences Items		kNN	NB	NB-K	RF	SVM
	Before (n=266)	89.10%	78.20%	88.35%	84.59%	85.34%
	After (n=218)	98.62%	91.74%	96.33%	96.33%	95.87%
	Difference in %	9.52%	13.54%	7.98%	11.74%	10.53%
	Critical Value (p<.05)	3.39%	5.16%	3.84%	4.19%	4.19%
	Significance	Yes	Yes	Yes	Yes	Yes
Health Sciences and Social Sciences Items		kNN	NB	NB-K	RF	SVM
	Before (n=266)	91.35%	87.22%	86.47%	89.10%	92.48%
	After (n=226)	95.58%	92.04%	97.79%	95.13%	97.35%
	Difference in %	4.23%	4.82%	11.32%	6.03%	4.87%
	Critical Value (p<.05)	3.61%	4.47%	3.79%	3.92%	3.18%
	Significance	Yes	Yes	Yes	Yes	Yes

3.3. Algorithm Performance After Removing Class-Inconsistent Items

In addition to the 28 class-inconsistent items initially identified, an additional 297 items were classified as class-inconsistent following a comprehensive cross-validation of the dataset. Subsequently, these 325 class-inconsistent items were removed from the dataset to evaluate their impact on the classification performance of the algorithms. The significance of percentage differences in classification accuracies was assessed using the test of significance for the difference between two proportions. The binomial and multinomial classification differences and their significance are presented in Table 9 and 10.

Table 10. Significance of Differences in Multinomial Classification Performance Before and After Removing Class-Inconsistent Items

		kNN	NB	NB-K	RF	SVM
Applied Sciences, Health Sciences and Social Sciences Items	Before (n=399)	84.96%	71.93%	83.96%	80.45%	78.70%
	After (n=334)	95.81%	89.82%	96.11%	91.92%	94.01%
	Difference in %	10.85%	17.89%	12.15%	11.47%	15.31%
	Critical Value (p<.05)	3.37%	4.46%	3.41%	3.96%	3.89%
	Significance	Yes	Yes	Yes	Yes	Yes

Tables 9 and 10 highlight the significant improvements in classification performance across different algorithm types (kNN, NB, NB-K, RF, and SVM) after the removal of class-inconsistent items for various combinations of item categories. Before removal, accuracy scores varied widely, with Naive Bayes often exhibiting lower performance compared to other algorithms. After removal, all algorithms showed notable accuracy gains, ranging from 4.23% to 17.89%, with Naive Bayes and SVM achieving the largest improvements.

The differences in performance were consistently statistically significant, with all the improvements observed exceeding the critical values at the 0.05 significance level. These findings underscore the detrimental impact of class-inconsistent items on classification accuracy and the universal benefit of addressing such inconsistencies across algorithms and item categories. The removal of these items not only enhanced overall accuracy but also highlighted the robustness of specific algorithms like Naive Bayes Kernel and SVM in coping with clean, consistent data.

4. Discussion

The present study investigated the classification performance of five widely used machine learning algorithms, k-Nearest Neighbors (kNN), Naïve Bayes (NB), Naïve Bayes Kernel (NB-K), Random Forest (RF), and Support Vector Machines (SVM) on domain-labeled exam items, with a particular focus on their robustness to class-inconsistent data. The findings suggest that all classifiers exhibit notable performance improvements when class-inconsistent instances are removed. This emphasizes the critical role of data quality in supervised learning, echoing prior research highlighting the detrimental effects of label noise on algorithmic accuracy[27], [30]

Among the algorithms, kNN demonstrated the highest level of robustness, with minimal performance degradation in the presence of noisy data. This aligns with Beretta and Santaniello's [28] findings, which suggest that instance-based learners like kNN can mitigate label noise through neighborhood aggregation, especially when proper parameter tuning is applied. Notably, kNN maintained strong classification performance across all domains even before the removal of problematic items, reinforcing its utility in real-world scenarios where noisy labels are inevitable.

On the other hand, Naïve Bayes exhibited the weakest performance, particularly under noisy conditions, despite its probabilistic foundations. While NB is known for its resilience to randomly distributed noise [29], it struggles when label inconsistencies correlate with specific feature distributions. NB-K, which incorporates kernel density estimation, outperformed standard NB in almost all settings, indicating that relaxing the feature independence assumption improves performance under moderate noise. However, NB-K's susceptibility to distorted density estimates under systematic mislabeling [31] was partially evident in this study, particularly when recall was considered.

SVM and RF demonstrated strong performance after class-inconsistent data were excluded, consistent with prior research suggesting that margin-based and ensemble methods benefit from cleaner datasets [33], [35]. Nevertheless, their sensitivity to mislabeled support vectors and overfitting in smaller feature subsets underlines the need for careful preprocessing and validation. The results suggest that although these algorithms can achieve high accuracy, they require more controlled conditions compared to kNN, particularly in educational text classification tasks.

Finally, the expert evaluation revealed that even human raters struggled to consistently label class-inconsistent items, with accuracy levels often dropping below 50%. This underscores the inherent ambiguity of some exam items, suggesting that classification errors may originate from the data itself rather than algorithmic shortcomings. Consequently, identifying and filtering out such ambiguous instances is essential not only for model optimization but also for improving assessment reliability in educational contexts.

5. Conclusion

On an overall basis, kNN demonstrates notable robustness in handling class-inconsistent data, maintaining over 84% accuracy even before the removal of problematic items. This suggests that kNN is relatively unaffected by inconsistencies and can classify effectively in noisy datasets. In contrast, while Random Forest also shows resilience to class inconsistency, its optimization and processing time are significantly higher, making it less practical for large-scale or time-sensitive applications. To sum up, given that class-inconsistent data may be inevitable in many real-world scenarios, kNN emerges as a more efficient and reliable choice for such tasks.

In addition to its robustness, kNN's computational simplicity and straightforward implementation make it an appealing choice for a wide range of classification tasks, especially in educational settings where ease of deployment is a priority. Because it maintains strong performance even in the presence of noisy data, kNN proves to be a reliable choice in practical, less controlled settings. Moreover, kNN's performance can often be enhanced further through parameter tuning, such as optimizing the value of "k," allowing for adaptability across various dataset characteristics.

While other algorithms like Naive Bayes Kernel and Support Vector Machines demonstrate substantial improvements in performance after addressing class-inconsistent data, their sensitivity to feature selection and preprocessing requirements can pose additional challenges. These algorithms may require more extensive preparation to achieve comparable results, highlighting the practical advantages of kNN in scenarios with limited resources or time constraints. By balancing simplicity, adaptability, and robustness, kNN solidifies its position as a versatile and dependable option for text classification tasks.

References

[1] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, “Assessing the accuracy of prediction algorithms for classification: an overview,” *Bioinformatics*, vol. 16, no. 5, pp. 412–424, May 2000, doi: 10.1093/bioinformatics/16.5.412.

[2] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, “Text classification algorithms: A survey,” *Information*, vol. 10, no. 4, Art. no. 150, Apr. 2019, doi: 10.3390/info10040150.

[3] J. Riggs and T. Lalonde, *Handbook for Applied Modeling: Non-Gaussian and Correlated Data*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[4] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, 3rd ed. Hoboken, NJ, USA: Wiley-Interscience, 2003.

[5] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, Sep. 1936.

[6] S. Har-Peled, D. Roth, and D. Zimak, “Constraint classification for multiclass classification and ranking,” in *Proc. 16th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, Dec. 2002, pp. 809–816.

[7] N. Matloff, *Statistical Regression and Classification: From Linear Models to Machine Learning*. Boca Raton, FL, USA: CRC Press, 2017.

[8] E. Apostolova and R. A. Kreek, “Training and prediction data discrepancies: Challenges of text classification with noisy, historical data,” in *Proc. 2018 EMNLP Workshop W-NUT: 4th Workshop on Noisy User-Generated Text*, Brussels, Belgium, Nov. 2018, pp. 104–109. doi: 10.18653/v1/W18-6114.

[9] J. Wainer, “Comparison of 14 different families of classification algorithms on 115 binary datasets,” *arXiv preprint*, Jun. 2016, [Online]. Available: https://arxiv.org/abs/1606.00930v1

[10] A. Hotho, A. Nürnberger, and G. Paaß, “A brief survey of text mining,” *J. Lang. Technol. Comput. Linguist.*, vol. 20, no. 1, pp. 19–62, Jul. 2005, doi: 10.21248/JLCL.20.2005.68.

[11] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008. doi: 10.1017/CBO9780511809071.

[12] T. Graepel, J. Q. Candela, T. Borchert, and R. Herbrich, “Web-scale Bayesian click-through rate prediction for sponsored search advertising in Microsoft’s Bing search engine,” 2010, [Online]. Available: https://discovery.ucl.ac.uk/id/eprint/1395202/

[13] M. Richardson, E. Dominowska, and R. Ragno, “Predicting clicks: Estimating the click-through rate for new ads,” in *Proc. 16th Int. World Wide Web Conf. (WWW)*, Banff, AB, Canada, 2007, pp. 521–530. doi: 10.1145/1242572.1242643.

[14] Y. Yang, Y. Yang, B. Jansen, and M. Lalmas, “Computational advertising: A paradigm shift for advertising and marketing?,” *IEEE Intell. Syst.*, vol. 32, no. 3, pp. 3–6, May 2017, doi: 10.1109/MIS.2017.58.

[15] J. Burrell, “How the machine ‘thinks’: Understanding opacity in machine learning algorithms,” *Big Data Soc.*, vol.

3, no. 1, Jan. 2016, doi: 10.1177/2053951715622512.

[16] N. Jindal and B. Liu, "Opinion spam and analysis," in *Proc. 2008 Int. Conf. Web Search and Data Mining (WSDM)*, Palo Alto, CA, USA, Feb. 2008, pp. 219–229. doi: 10.1145/1341531.1341560.

[17] E. P. Lim, V. A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, "Detecting product review spammers using rating behaviors," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manag. (CIKM)*, Toronto, ON, Canada, Oct. 2010, pp. 939–948. doi: 10.1145/1871437.1871557.

[18] S. Redhu, "Sentiment analysis using text mining: A review," *Int. J. Data Sci. Technol.*, vol. 4, no. 2, p. 49, 2018, doi: 10.11648/J.IJDST.20180402.12.

[19] C. Zucco, B. Calabrese, G. Agapito, P. H. Guzzi, and M. Cannataro, "Sentiment analysis for mining texts and social networks data: Methods and tools," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 10, no. 1, p. e1333, Jan. 2020, doi: 10.1002/widm.1333.

[20] A. Dhar, H. Mukherjee, K. Roy, K. C. Santosh, and N. S. Dash, "Hybrid approach for text categorization: A case study with Bangla news article," *J. Inf. Sci.*, vol. 49, no. 3, pp. 762–777, Jun. 2023, doi: 10.1177/01655515211027770.

[21] H. Gomes, M. de Castro Neto, and R. Henriques, "Text mining: Sentiment analysis on news classification," in *Proc. 8th Iberian Conf. Inf. Syst. Technol. (CISTI)*, Lisboa, Portugal, Jun. 2013, pp. 1–6.

[22] M. Jamaluddin and A. D. Wibawa, "Patient diagnosis classification based on electronic medical record using text mining and support vector machine," in *Proc. 2021 Int. Seminar Appl. Technol. Inf. Commun. (iSemantic)*, Semarang, Indonesia, Sep. 2021, pp. 243–248. doi: 10.1109/isemantic52711.2021.9573178.

[23] C. Luque, J. M. Luna, M. Luque, and S. Ventura, "An advanced review on text mining in medicine," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 9, no. 3, p. e1302, May 2019, doi: 10.1002/widm.1302.

[24] Y. L. Chen, Y. H. Liu, and W. L. Ho, "A text mining approach to assist the general public in the retrieval of legal documents," *J. Am. Soc. Inf. Sci. Technol.*, vol. 64, no. 2, pp. 280–290, Feb. 2013, doi: 10.1002/asi.22767.

[25] K. Berezka, O. Kovalchuk, S. Banakh, S. Zlyvko, and R. Hrechaniuk, "A binary logistic regression model for support decision making in criminal justice," *Folia Oecon. Stetin.*, vol. 22, no. 1, pp. 1–17, Jun. 2022, doi: 10.2478/foli-2022-0001.

[26] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Trans. Syst., Man, Cybern.*, vol. 2, no. 3, pp. 408–421, 1972, doi: 10.1109/tsmc.1972.4309137.

[27] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study," *Artif. Intell. Rev.*, vol. 22, no. 3, pp. 177–210, Nov. 2004, doi: 10.1007/s10462-004-0751-8.

[28] L. Beretta and A. Santaniello, "Nearest neighbor imputation algorithms: a critical evaluation," *BMC Med. Inform. Decis. Mak.*, vol. 16, Jul. 2016, doi: 10.1186/s12911-016-0318-z.

[29] H. Zhang, "The optimality of naive Bayes," in *Proc. 17th Int. Florida Artif. Intell. Res. Soc. Conf. (FLAIRS)*, Miami Beach, FL, USA, May 2004, pp. 562–567. [Online]. Available: https://cdn.aaai.org/FLAIRS/2004/Flairs04-097.pdf

[30] B. Frénay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 845–869, May 2014, doi: 10.1109/tnnls.2013.2292894.

[31] N. Cesa-Bianchi, S. Shalev Shwartz, and O. Shamir, "Online learning of noisy data," *IEEE Trans. Inf. Theory*, vol. 57, no. 12, pp. 7907–7931, Dec. 2013, doi: 10.1109/tit.2011.2164053.

[32] J. Kim and C. D. Scott, "Robust kernel density estimation," *J. Mach. Learn. Res.*, vol. 13, pp. 2529–2565, 2012, [Online]. Available: https://www.jmlr.org/papers/volume13/kim12b/kim12b.pdf

[33] N. Manwani and P. S. Sastry, "Noise tolerance under risk minimization," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 1146–1151, Jun. 2013, doi: 10.1109/tsmcb.2012.2223460.

[34] B. Biggio, G. Fumera, and F. Roli, "Multiple classifier systems for robust classifier design in adversarial environments," *Int. J. Mach. Learn. Cybern.*, vol. 1, no. 1–4, pp. 27–41, Dec. 2010, doi: 10.1007/s13042-010-0007-7.

[35] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/a:1010933404324.

[36] A. Folleco, T. M. Khoshgoftaar, J. Van Hulse, and L. Bullard, "Software quality modeling: The impact of class noise on the random forest classifier," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Hong Kong, Jun. 2008, pp. 3853–3859. doi: 10.1109/cec.2008.4631321.

[37] J. Zhao, M. Kang, and Z. Han, "Robustness of classification algorithm in the face of label noise," *EAI Endorsed Trans IoT*, vol. 9, no. 1, p. e5, Jun. 2023, doi: 10.4108/eetiot.v9i1.3270.

[38] J. Wilton and N. Ye, "Robust loss functions for training decision trees with noisy labels," in *Proc. AAAI Conf. Artif. Intell.*, Mar. 2024, pp. 15859–15867. doi: 10.1609/aaai.v38i14.29516.

[39] A. Srivastava and M. Sahami, *Text mining: Classification, clustering, and applications*. Boca Raton, FL, USA: CRC Press, 2009.

[40] B. W. Silverman and M. C. Jones, "E. Fix and J. L. Hodges (1951): An important contribution to nonparametric discriminant analysis and density estimation: Commentary on Fix and Hodges (1951)," *Int. Stat. Rev.*, vol. 57, no. 3, p. 233, Dec. 1989, doi: 10.2307/1403796.

[41] E. Alpaydın, *Introduction to Machine Learning*, 4th ed. Cambridge, MA, USA: MIT Press, 2020.

[42] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Stat.*, vol. 33, no. 3, pp. 1065–1076, Sep. 1962, doi: 10.1214/aoms/1177704472.

[43] A. Christmann and I. Steinwart, "Support vector machines," in *Support Vector Machines: Theory and Applications*, L. T. Yang, Ed., Boston, MA, USA: Springer, 2005, pp. 93–123.

[44] A. Cutler, D. R. Cutler, and J. R. Stevens, "Random forests," in *Ensemble Machine Learning: Methods and Applications*, C. Zhang and Y. Ma, Eds., Boston, MA, USA: Springer, 2012, pp. 157–175. doi: 10.1007/978-1-4419-9326-7_5.

[45] S. Vijayarani, M. J. Ilamathi, and M. Nithya, "Preprocessing techniques for text mining—an overview," *Int. J. Comput. Sci. Commun. Netw.*, vol. 5, no. 1, pp. 7–16, Feb. 2015, [Online]. Available: https://www.researchgate.net/publication/339529230_Preprocessing_Techniques_for_Text_Mining_-_An_Overview

[46] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Min. Knowl. Manag. Process*, vol. 5, no. 2, pp. 1–11, Mar. 2015, doi: 10.5121/ijdkp.2015.5201.

[47] K. Pearson, "Mathematical contributions to the theory of evolution. XII—On a generalised theory of alternative inheritance, with special reference to Mendel's laws," *Proc. R. Soc. Lond.*, vol. 72, no. 477–486, pp. 505–509, Jan. 1904, doi: 10.1098/rspl.1903.0081.

[48] J. Fan, S. Upadhye, and A. Worster, "Understanding receiver operating characteristic (ROC) curves," *Can. J. Emerg. Med.*, vol. 8, no. 1, pp. 19–20, Jan. 2006, doi: 10.1017/s1481803500013336.

[49] İ. Akhun, "İki yüzde arasındaki farkın manidarlığının test edilmesi," *Ankara Univ. Egit. Bilim. Fak. Derg.*, vol. 15, no. 1, pp. 240–259, Jan. 1982, doi: 10.1501/egifak_0000000817.

Article Information Form

Authors Contributions

Hüseyin Ataseven: Conceptualization, Writing – original draft, Visualisation, Validation, Methodology, Data analysis .

Ömay Çokluk Bökeoğlu: Conceptualization, Writing – review & editing, Methodology, Investigation &Supervision.

Conflict of Interest Notice

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

Ethical Approval and Informed Consent

Ethical approval for this study was obtained from Ankara University, ensuring adherence to research and publication ethics throughout the study. The approval was granted by the Social Sciences Ethics Committee, with reference number 52, on 06/02/2023. The authors affirm that all scientific and ethical principles were upheld during the preparation and execution of this research. Furthermore, all sources and studies referenced have been appropriately cited in the bibliography, ensuring transparency and academic integrity.

Artificial Intelligence Statement

Artificial intelligence tools were utilized solely for language editing purposes. All ideas, arguments, and information presented in this article were conceived and written by the authors, who bear full responsibility for the content.

Plagiarism Statement

This article has been scanned by iThenticate ™.