ISSN 1870-4069

Classification of Offensive Comments on the Web Using SVM

Claudio Eduardo Gómez Cabrera[1], Abdiel Reyes Vera[2]

[1,2] Instituto Politécnico Nacional,
Escuela Superior de Ingeniería Mecánica y Eléctrica,
Mexico

[2] Instituto Politécnico Nacional,
Escuela Superior de Cómputo,
Mexico

{claedgomcab, abdielreyes81}@gmail.com

Abstract. This paper presents a web application for the classification of offensive comments using Support Vector Machines (SVM). A corpus generated by web scraping of video games and movies review platforms was used. Natural Language Processing (NLP) techniques such as tokenization and TF-IDF were implemented for data representation. The results show that the proposed model improves the identification of offensive content with high accuracy, providing an efficient solution to improve moderation on digital platforms.

Keywords: Text classification, NLP, SVM, web scraping.

1 Introduction

The growth of digital platforms has facilitated global communication, but it has also generated an increase in offensive comments, hate speech, and online harassment. Moderating this content is a challenge, as it must balance freedom of expression with the creation of safe spaces for users. Although manual moderation is still used, it is a slow and costly process, requiring automated solutions based on artificial intelligence.

This paper proposes the use of Support Vector Machines (SVM) for the classification of offensive comments, using Natural Language Processing (NLP) techniques. For this purpose, data were collected from platforms such as Steam and Metacritic through web scraping, which were preprocessed and manually labeled. The performance of SVM was compared with other classification models, demonstrating its effectiveness in detecting offensive content with high accuracy. Despite the rise of deep learning architectures such as Transformers, traditional models like SVM remain highly competitive, especially when computational resources or data availability are limited. Their lower complexity and faster training times make them ideal for integration into lightweight applications such as browser extensions.

In addition, a web extension was developed to implement real-time moderation within browsers. This tool operates independently of platform infrastructure, offering a scalable and accessible solution for content moderation. This article explores the theoretical foundations of text classification, the method- ology employed, the results

obtained, and the implications of this solution for improving moderation in digital environments.

2 Theoretical Framework

In order to develop an artificial intelligence-based offensive comment moderation system, it is essential to understand the concepts underlying this solution. This section addresses the principles of automatic learning, the use of Support Vector Machines (SVM) in text classification, and the Natural Language Processing (NLP) techniques applied in data preprocessing.

2.1 Machine Learning

Machine Learning allows models to identify patterns in data without being explicitly programmed. It is divided into several categories, with supervised learning being the most relevant in this work, as the SVM model is trained on previously labeled data.

Supervised Learning This approach is based on learning a function that relates the input data to their respective output labels. For this purpose, the data are divided into two sets: training, used to fit the model, and testing, used to evaluate its performance. (El Naqa & Murphy, 2015)

2.2 Natural Language Processing

NLP is a branch of artificial intelligence focused on the interaction between computers and human language.

In this work, techniques such as tokenization, lemmatization and TF-IDF vector representation were employed to transform comments into a format processable by classification models.

2.3 Term Frequency - Inverse Document Frequency (TF-IDF)

To represent the comments numerically, we used TF-IDF (Term Frequency - Inverse Document Frequency), a technique that measures the importance of a word in a set of documents.

This methodology made it possible to highlight terms characteristic of offensive comments and to minimize the impact of common words:

- **Term Frequency (TF)**: Represents how many times a word appears in a comment.
- **Inverse Document Frequency (IDF)**: Penalizes words that are too common throughout the corpus, reducing their weight in the classification.

The use of TF-IDF helped the SVM model to detect patterns in offensive language without the need for a predefined list of forbidden words, since insults tend to be repeated and acquire greater weight within the corpus.

2.4 Support Vector Machines (SVM)

Support Vector Machines (SVM) are supervised learning models used for classification and regression. Their goal is to find a hyperplane that maximizes the separation between different classes, optimizing the accuracy of the model.

Maximum Margin Concept: The margin is the distance between the separator hyperplane and the nearest points of each class (support vectors). Maximizing this margin improves the model's ability to generalize and minimize classification errors in unseen data. (Jakkula, 2006).

2.5 Web Scraping and Legal Considerations

Web scraping is an automated technique used to extract information from web- sites. While it facilitates the collection of data for training NLP models, its use must adhere to international legal and ethical standards. Organizations such as the United Nations (UN) and the United Nations Educational, Scientific and Cultural Organization (UNESCO) emphasize the importance of protecting per- sonal data and respecting digital rights.

To minimize legal and ethical risks, it is recommended to verify whether a platform provides an official API and to consult its terms of service before applying scraping techniques (Kinsta, 2025).

What data can be scraped: It is possible to extract any type of data from different web pages. Many websites allow scrapers to access their data, but not all do so for free. The way in which different types of data can be obtained depends on the website. If there is an API (Application Programming Interface) to obtain the necessary data, it is recommended to use them.

However, this practice is not always well-received, and some sites may im- pose limitations. There are some tricks you can implement to circumvent these limitations, taking care not to violate the website's rules or breach copyright or personal data protection laws (Kinsta, 2025).

Privacy and Data Protection Risks: A common misconception is that scraping public websites is always legally permissible. However, data protection laws apply regardless of whether the information is publicly accessible. For instance, under the General Data Protection Regulation (GDPR), personal data must be processed lawfully, fairly, and transparently. Even public data, if linked to identifiable individuals, falls under the scope of protection.

Personal data includes names, email addresses, IP addresses, device identifiers, and other information that can be associated with an individual: Improper collection or use of such data without consent may lead to significant legal consequences. Therefore, it is crucial to ensure that any scraping activity respects data privacy frameworks and ethical guidelines to avoid breaching legal obligations. (EDJ-XTECH-LAW-SCHOOL, 2023)

3 State of the Art

Automatic moderation of offensive comments has been extensively studied within Natural Language Processing (NLP) and artificial intelligence. There are multiple approaches,

ranging from traditional methods such as Support Vector Ma- chines (SVM) to advanced models such as Transformers and Convolutional Neural Networks (CNN).

3.1 Hate Speech and Offensive Language Detection

One of the main challenges in classifying offensive comments is to identify the context and intent of the language. In (Rodríguez & Pérez, 2023), we experimented with the Pysentimiento model based on Transformers, obtaining good results in the classification of emotions and polarity in Spanish. It was shown that neural models can better capture language semantics compared to traditional techniques such as TF-IDF + SVM.

On the other hand, (Jiménez & Rojas, 2022) evaluated the performance of SVM and CNNs in detecting hate on Twitter. The results showed that SVM, combined with TF-IDF, can achieve good results, although CNNs outperformed them due to their ability to capture more complex semantic relationships.

3.2 Models Based on Transformers

Transformers, such as BERT, have significantly improved the detection of offensive language in social networks. In (Basile & et al., 2019), BERT was compared with SVM, noting that BERT achieves higher accuracy and F1-score due to its ability to interpret context. However, these models require more computational power and large volumes of data for training.

A hybrid approach proposed in (Park, Shin, & Lee, 2020) combined BERT and CNN to improve the classification of offensive language, capturing both textual features of the text and local patterns of offensive keywords. Although this technique achieved better results, its high computational cost represents a limitation.

3.3 Comparison of Methods and SVM Justification

Although advanced models have demonstrated superior performance, traditional methods such as SVM remain a viable alternative in resource-limited scenarios. In this work, SVM with TF-IDF was chosen due to:

- **Size of the data set:** The corpus used consists of 4049 comments, which is insufficient for models such as BERT, which require large volumes of data.
- **Computational efficiency:** Transformers and CNNs require more processing power, making them difficult to implement in real time. SVM is a lighter and more efficient alternative.
- **Good performance with TF-IDF:** SVM achieved an accuracy of 84.09% in this study, demonstrating its feasibility in detecting offending comments with small datasets.

Given these factors, SVM represents a balance between accuracy, efficiency and ease of implementation, being suitable for automated content moderation in a web extension.

4 Methodology

To evaluate the effectiveness of Support Vector Machines (SVM) in the classification of offensive comments, a structured process was followed that included model collection, preprocessing, training and evaluation. Data were obtained by web scraping from platforms such as Steam and Metacritic, applying Natural Language Processing (NLP) techniques for cleaning and conversion to numerical representations with TF-IDF.

The SVM model was trained with a previously labeled corpus and compared with other classification algorithms, such as KNN and Random Forest. Finally, a web extension was implemented to apply the model in real time within the browsers. The main steps of the process are detailed below.

4.1 Data Collection

Web scraping was used to extract comments from review platforms, saving them in a CSV file for further processing with the Pandas library in Python. Comments were classified into two categories: 1 (Non-offensive) and 0 (Offensive). A comment was considered not offensive if it did not contain:

- Offensive words.
- Sexual Apology.
- Ambiguous content.
- Excessively short comments (two words or less).

These criteria were established after analyzing the recurrent patterns in offensive comments. In total, 4049 comments were collected, of which 3240 were used for model training and the rest for the testing phase.

4.2 Data Labeling Process

To label the dataset, three domain experts manually annotated each comment as offensive or non-offensive. The labeling criteria were based on the presence of profanity, hate speech, discrimination, or personal attacks. In cases of dis- agreement, a consensus was reached through discussion. This manual approach ensured the reliability of the labeled data used for training and testing the classification models.

If some data do not meet any of the criteria established by the consensus, the data in question are discarded to improve the quality of the overall data and not contribute useless data to the model, thus improving optimal classification in the training of the model.

4.3 Data Preprocessing

Before training the models, the text data was preprocessed through several steps: conversion to lowercase, removal of punctuation and special characters, elimination of stopwords, and tokenization:

Table 1. Accuracy of the evaluated models.

Model	Accuracy	Description
KNN	66%	Calculate the distance between the point to be classified and its nearest neighbors.
RF	72%	Create multiple decision trees and take the most common classification among them.
SVM	84%	Find an optimal hyperplane that maxi- mizes the margin between categories.

- **Tokenization:** Tokenization was used at the word level to divide the comments into individual units, eliminating punctuation marks and separating each term. This allowed structuring the text so that it could be analyzed by the classification model.
- **Elimination of Stopwords:** Stopwords are frequent words such as "the", "of", "and", which do not contribute value in the classification. They were eliminated using predefined lists in Spanish and English, reducing the dimensionality of the data and improving the efficiency of the model.
- **Text Normalization:** To ensure a uniform treatment of words, the following techniques were applied:
- **Lowercase conversion:** Prevents identical words from being treated as distinct terms (e.g., "Video game" and "video game").
- **Elimination of special characters and emojis:** Symbols and emojis were discarded as they did not provide useful information for classification.
- **Filtering Ambiguous Content:** Extremely short comments (one or two words) were eliminated because they did not provide significant information for classification. Also discarded were those composed only of special characters or graphic patterns used to generate figures with inappropriate content.

4.4 Model Training

The SVM model was trained with a manually labeled corpus, using TF-IDF for text representation. Its performance was compared with KNN and Random Forest, obtaining better results.

As shown in Table 1, SVM obtained the best performance. The regularization parameter "C", which balances margin maximization and classification error minimization, was optimized. To determine its optimal value, tests were performed with different values, observing their impact on accuracy and recall.

It was found that low values of "C" favored a wide margin with more errors, while high values reduced error tolerance, generating overfitting. After several iterations, the value offering the best balance was selected.

Table 2. Comparison of Metrics.

Model	Class	Accuracy	Recall	F1-Score
KNN (66.50%)	0 (Offensive)	0.84	0.70	0.76
	1 (Not Offensive)	0.61	0.62	0.61
Random Forest (72.00%)	0 (Offensive)	0.75	0.80	0.77
	1 (Not Offensive)	0.70	0.70	0.70
SVM (84.09%)	0 (Offensive)	0.94	0.80	0.87
	1 (Not Offensive)	0.70	0.90	0.79

4.5 Hyperparameter Tuning

Each classifier was trained using a specific set of hyperparameters. The Sup- port Vector Machine (SVM) and Random Forest (RF) classifiers were manually configured based on preliminary experimentation, while K-Nearest Neighbors (KNN) was used with default parameters due to its low observed performance.

- **Support Vector Machine (SVM):** A radial basis function (RBF) kernel was used, with a regularization parameter $C = 1$ and kernel coefficient $\gamma = 0.2$. This configuration was chosen to handle non-linear relationships within the TF-IDF-transformed feature space.
- **Random Forest (RF):** The number of decision trees was increased to 300 to enhance model stability and reduce variance. Other parameters, such as maximum depth and minimum samples per leaf, were kept at their default values.
- **K-Nearest Neighbors (KNN):** The classifier was applied with default settings, including $k = 5$ and Euclidean distance. No tuning was performed, as initial tests showed low classification accuracy, making further optimization less relevant.

4.6 Justification for the Choice of SVM

For the classification of offensive comments, different models were evaluated with metrics such as accuracy, precision, recall, and F1-score. SVM proved to be the best option.

Comparison with Other Models Evaluated KNN and Random Forest were evaluated, with inferior results:

- **KNN**: Although it had good accuracy in the offensive class (0.84), its performance in the non-offensive class was low (0.61), with an overall accuracy of 66.50%.
- **Random Forest**: Although more balanced, its accuracy was only 72.00%, with an F1-score of 0.70 in the non-offensive class.

SVM outperformed both models in accuracy and classification of non-offensive comments, being the best choice for this task.

Reasons for SVM Selection SVM was chosen for the following reasons:

- **High accuracy:** Achieved 84.09%, surpassing other models.
- **Accuracy in non-offensive comments:** Obtained an accuracy of 0.94, minimizing

false positives.

- **Efficiency in detecting offensive comments:** A recall of 0.90 indicates a low number of false negatives.
- **Effective generalization:** Thanks to the maximum margin principle, it avoids overfitting and improves classification on new data.
- **High dimensionality handling:** SVM works efficiently with TF-IDF, find- ing optimal hyperplanes in multidimensional spaces.
- **Computational efficiency:** It requires fewer resources compared to neural networks, facilitating its implementation in a web extension.

4.7 Hyperparameter Tuning

The K-Nearest Neighbors (KNN) classifier was tested with different values of *k* ranging from 3 to 10, and the best results were achieved with $k = 5$. For the Random Forest (RF) classifier, we used 100 trees with a maximum depth of 10. No grid search or cross-validation was performed for hyperparameter tuning in this initial version, but it is considered as a future improvement.

4.8 Use of TF-IDF

To represent the comments numerically, TF-IDF was used. This technique al- lowed us to evaluate the importance of each word within the corpus, assigning greater weight to terms that were distinctive of offensive comments.
The use of TF-IDF improved the classification due to:

- **Repetition of offensive words**: Insults and aggressive expressions are often recurrent in offensive comments. TF-IDF assigns greater weight to these terms, facilitating their identification.
- **Differentiation of key terms**: Offensive expressions have a wide distribution. The model is able to detect them without predefined lists of prohibited words.
- **Reduction of the impact of common terms:** TF-IDF balances the weight of frequent words, preventing the model from being biased towards overused terms in both types of comments.

Unlike a simple Bag of Words, TF-IDF allowed not only to count terms, but also to weight them according to their relevance in the classification. Thanks to this representation, SVM identified more accurately offensive linguistic patterns, improving the performance of the system.

4.9 Web Extension Development

To implement automated moderation of offensive comments, a web extension was developed that interacts with a Python API. The extension does not directly extract the comments, but sends the URL of the page to the API, which performs web scraping, classifies the comments with SVM and returns only the non offensive ones for display.

Interaction with Web Pages and Content Modification**.** The web extension

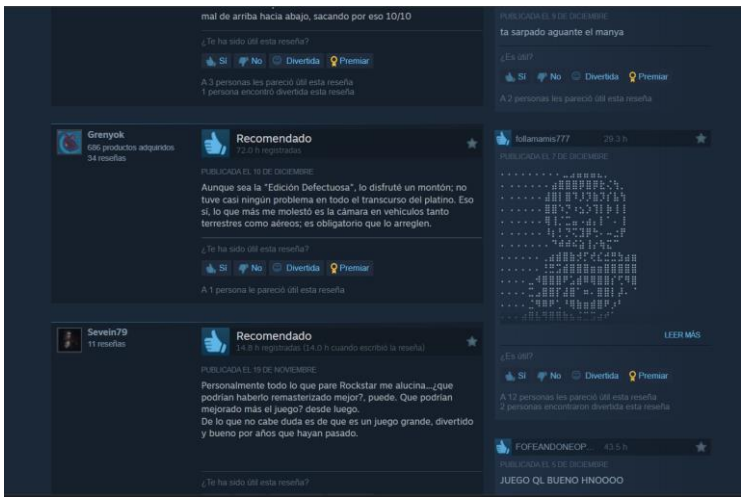

Fig. 1. Inappropriate comments in the Steam comments section.

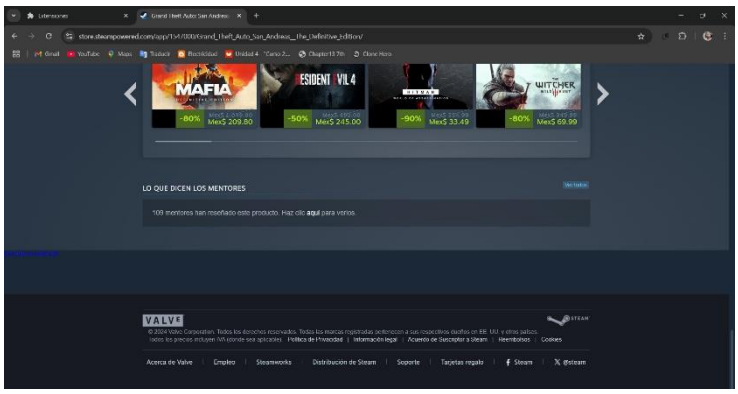

Fig. 2. Hidden inappropriate comments.

dynamically modifies the structure of the visited pages to ensure a safe environment and a clean presentation. It works as follows:

- **Sending the URL to the API**: The extension captures the URL of the page with comments and sends it to the API for analysis.
- **Extraction and classification**: The API applies web scraping to obtain the comments and uses SVM to classify them as offensive or non-offensive.
- **Display of filtered comments**: The API returns only non-offensive comments, and the extension modifies the page in real time to display them, hiding the offensive ones without affecting the visual structure.
- **Dynamic modification of the content**: The extension replaces the section original feedback with the filtered ones, ensuring a seamless experience.

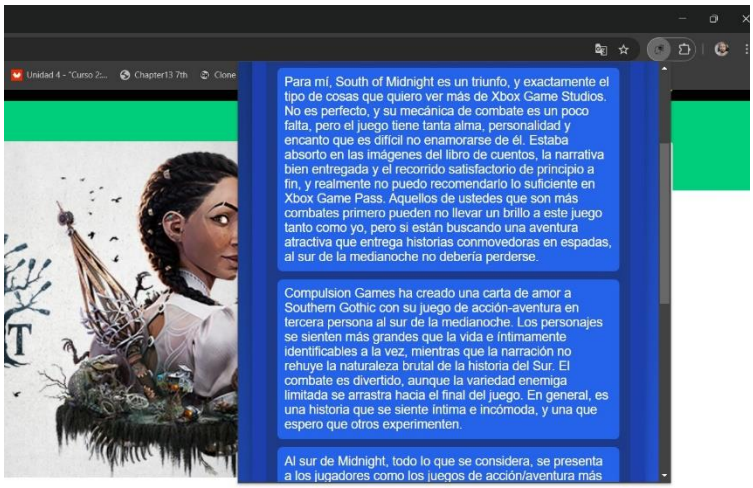

Fig. 3. Classified comments in the extension.

In Fig. 1, inappropriate content is displayed in the comments section. There- fore, the dynamic modification provided by the web extension was used to hide the comments section on the various pages to which the rating applies and pro- vide a space free of inappropriate content.

Once the web extension has modified the web page's source code, the comments section is hidden as shown in Fig.2 to prevent the user from encountering such content and leaving them only the option of viewing the appropriate comments with the web extension.

Communication between the Extension and the API The communication process between the web extension and the API follows these steps:

- **Capture and send URL**: The extension detects the visited page and sends the URL to the API via HTTP request.
- **Extraction and classification**: The API identifies the structure of the page, extracts the comments and processes them with the SVM model.
- **Return of filtered comments**: The API responds with the comments classified as appropriate Fig.3.

Impact on Security and User Experience. The extension ensures a secure environment by displaying only API-filtered comments, preventing exposure to inappropriate content without manual intervention. In addition, it visually restructures the page to avoid empty spaces or clutter in the original layout, providing a smooth user experience without alterations to the functionality of the visited website.

Beyond content moderation, the implementation of the web extension rep- resents a scalable and adaptable solution for different platforms. Its integration with an API makes it possible to continuously update and improve the classification criteria without the need to modify the extension's code. This facilitates the implementation of future improvements, such as the incorporation of new machine learning models or the adaptation to changes in the structure of the analyzed web pages.

As can be seen in Fig.3, the filtered comments are displayed in the web extension once the previously mentioned communication is completed and the classified comments are returned within the API.

5 Results

The results show that the SVM model achieved an accuracy of 84.09% in classifying offensive comments, outperforming KNN (66.50%) and Random Forest (72%). This confirms that Support Vector Machines are suitable for this task by balancing accuracy and generalization.

5.1 Model Performance

In the Table 2, shows that the SVM model achieved an F1-score of 0.87 for non offensive comments, indicating a high predictive ability in this category. For offensive comments, the model achieved a recall of 0.90, which means that it was able to identify most of the inappropriate comments with a low margin of error. In contrast, KNN showed difficulties in classifying non-offensive comments, with an F1-score of 0.61, while Random Forest obtained a moderate improvement at 0.70. This suggests that SVM not only outperforms these models in overall accuracy, but also reduces false positives and false negatives, crucial aspects for automated moderation.

5.2 Efficiency in Web Extension

During testing, the web extension ran on multiple platforms, allowing offensive comments to be filtered without affecting the structure of the pages. Integration with the API enabled seamless real-time moderation, ensuring that only comments classified as non-offensive were displayed.

Despite the good results, challenges were identified in classifying very short comments, as in some cases words with high TF-IDF weights led to incorrect classifications. To reduce these errors, a bag of words was created containing problematic terms identified in the analysis. These words were assigned a lower weight within the model, allowing to adjust its impact on the classification and improving accuracy in cases where the context was limited.

However, the system showed high effectiveness in detecting offensive language, providing a viable solution for online content moderation.

6 Conclusions

The effectiveness of the Support Vector Machines (SVM) based model for comment moderation in digital platforms was demonstrated. Through the use of Natural Language Processing (NLP) and text representation using TF-IDF, we were able to train a model capable of identifying offensive comments with high accuracy, providing an effective tool to improve security in digital environments. The use of web scraping was fundamental in data collection, allowing us to obtain a representative corpus of real

comments from platforms such as Steam and Metacritic. This technique facilitated the construction of a labeled dataset, essential for the training and evaluation of the model. Despite the challenges presented by web scraping, such as the restrictions imposed by some platforms, its implementation proved to be a viable alternative for obtaining data in text classification projects.

In addition, the integration of the model into a web extension allowed for real time implementation within browsers. This represents a practical and accessible solution for moderating comments on various platforms, without the need to modify the infrastructure of the websites. The ability of web extensions to interact with HTML code and communicate with an external API proved to be a key factor in the implementation of the offensive comment filtering system.

Despite the rise of models based on Transformers and profound neural net- works, the choice of SVM was appropriate for several reasons:

- **Lower computational cost**: SVM does not require the high computational power demanded by more complex models, which facilitates its implementation in resource-constrained environments.
- **Increased interpretability**: Compared to deep networks, SVM models allow a better understanding of the model's decisions, which is beneficial in content moderation applications.
- **Efficiency on small datasets**: With a corpus of 4049 comments, SVM achieved competitive accuracy without requiring large volumes of data for training.

In conclusion, this project presents an effective and scalable solution for the detection and moderation of offensive comments on the web. Its implementation in browsers through web extensions makes it an accessible and easy to integrate tool, with the potential to improve the user experience in digital environments.

References

1. Basile, V.: Um-iu@ling at Semeval-2019 Task 6: Identifying Offensive Tweets Using Bert and SVMs. ArXiv doi: 10.48550/arXiv.1904.03450.
2. Google Developers: Chrome extensions overview (2023) https:// developer.chrome.com/docs/extensions.
3. EDJ-XTECH-LAW-SCHOOL (2023). Los riesgos legales del web scraping: Privacidad, protección de datos y malos usos. Retrieved from https:// www.edjxtechlawschool.com/post/los-riesgos-legales-del-web-scraping-privacidad-protecci%C3%B3n-de-datos-y-malos-usos
4. Electronic Frontier Foundation: Automated Content Moderation: Challenges and Recommendations (2020) https://www.eff.org/issues/automated-content-moderation
5. El Naqa, I., Murphy, M.J. What is Machine Learning? In: Machine Learning in Radiation Oncology: Theory and Applications, pp. 3–11 (2015) doi: 10.1007/978-3-319-18305-3_1.
6. Microsoft Foundation: Browser Extensions – Introduction (2023) https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/ WebExtensions.

7. Hardeniya, N., Perkins, J., Chopra, D.: Natural Language Processing: Python and NLTK. Packt Publishing Ltd (2016)
8. Jakkula, V.: Tutorial on Support Vector Machine (SVM) (2006) https://www.public.asu.edu/~jakkula/tutorialsvm.pdf.
9. Jiménez, M., Rojas, C.: Comparison of Models for Automatic Hate Speech Detection on Twitter. https://www.kerwa.ucr.ac.cr/items/340cc182-c780-47a0-ad7b-ec4495f2dbd0
10. Kinsta: ¿Qué es el web scraping? Cómo extraer legalmente el contenido de la web (2025) https://kinsta.com/es/base-de-conocimiento/que-es-web-scraping.
11. Park, J., Shin, J., Lee, S.G.: KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media. arXiv (2020) doi: 10.48550/arXiv.2007.13184.
12. ProxyElite.: Aspectos legales del web scraping: Lo que necesita saber para evitar infringir la ley (2022) https://proxyelite.info/es/legal-aspects-of-web-scraping-what-you-need-to-know-to-avoid-breaking-the-law/.
13. Rodríguez, A., Pérez, L.: Aggression and Hate in Spanish Text Messages: Identification Using Transformers (2023) https://laccei.org/LACCEI2023-BuenosAires/papers/Contribution_1077_a.pdf
14. Spärck Jones, K.: A statistical Interpretation of Term Specificity and its Application in Retrieval. Journal of Documentation, 28(1), pp. 11−21 (1972)
15. UNESCO.: La gobernanza de internet y la protección de derechos humanos (2021) https://unesdoc.unesco.org/ark:/48223/ pf0000377231.