WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

Reviewlytics: Fake Review Detection and Customer Feedback Analytics Using Machine Learning

Mrs. Kokila G[1], Janushri K[2], Jayasri K[3], Harini KR[4], Kaviya G[5]

[1]Associate Professor, Department of Artificial Intelligence and Data Science, Sri Shakthi Institute of Engineering and Technology, TamilNadu, Coimbatore-641062, India

[2]Student, Department of Artificial Intelligence and Data Science, Sri Shakthi Institute of Engineering and Technology, TamilNadu, Coimbatore-641062, India

[3]Student, Department of Artificial Intelligence and Data Science, Sri Shakthi Institute of Engineering and Technology, TamilNadu, Coimbatore-641062, India

[4]Student, Department of Artificial Intelligence and Data Science, Sri Shakthi Institute of Engineering and Technology, TamilNadu, Coimbatore-641062, India

[5]Student, Department of Artificial Intelligence and Data Science, Sri Shakthi Institute of Engineering and Technology, TamilNadu, Coimbatore-641062, India

ABSTRACT

Online product reviews shape buying decisions, yet some are written to mislead. This paper presents Reviewlytics, a web-based platform that identifies potentially suspicious product reviews and summarises customer feedback for electronics such as headphones, televisions, laptops, mobile phones, speakers, wearables and tablets. Review text is cleaned and converted to TF-IDF features, and a Logistic Regression classifier assigns each review a suspicion score and a Genuine or Potentially Suspicious assessment. A local VADER lexicon scores sentiment separately, so tone is never mistaken for authenticity. A React dashboard, backed by a FastAPI service and a SQLite history, supports single-review checks, bulk CSV or Excel analysis, product-wise classification charts and review-trend charts. The system was exercised on a synthetic demonstration dataset of 5,000 reviews. Its outputs are model-generated assessments, not proof that a review is fake, and results on synthetic data do not represent real-world performance.

Keywords: Fake Review Detection, Sentiment Analysis, TF-IDF, Logistic Regression, Customer Feedback Analytics, FastAPI, React.

1. Introduction

Customers rely on written reviews and star ratings when choosing electronics. Because reviews influence sales, some are written to mislead, either to promote a product or to harm a competitor, and reading thousands of them by hand is not practical. Machine learning can learn textual patterns that tend to separate suspicious reviews from ordinary ones, but a classifier alone is not enough: sellers and analysts also need to see how feedback is spread across products and time, and how customers feel about each product.

Reviewlytics combines these needs in one application built only from free tools: React, FastAPI, scikit-learn, VADER and SQLite. This work provides (i) single-review checks and bulk CSV or Excel analysis, (ii) a Genuine or Potentially Suspicious assessment with a suspicion score, (iii) sentiment reported separately from authenticity, and (iv) product-wise classification and trend charts with a saved history. The outputs are model-generated assessments, not proof that a review is fake, and they should support human judgement rather than replace it.

2. Literature Survey

Early work on opinion spam studied Amazon product reviews and treated duplicated or near-duplicated reviews as a clear sign of spam [1]. Later studies built text classifiers using word n-grams and linear models such as support vector machines, and showed that wording alone can separate deceptive from truthful hotel reviews when the labels come from a controlled, crowdsourced collection [2]. Research on filtered reviews from a commercial platform found that text-only classifiers perform worse on real data than on crowdsourced data, and that behavioural signals such as reviewer activity patterns helped more [3].

For feature extraction, TF-IDF weighting [4] remains a simple and strong baseline for text classification and is available in scikit-learn [5]. Transformer models such as BERT [6] can learn richer representations but need more data and computing power. Sentiment analysis is often used alongside spam detection; VADER is a rule-based lexicon tuned for short social-media style text that runs locally and needs no training [7].

Existing studies mostly focus on the classifier, and few provide an end-to-end tool that combines classification, sentiment, bulk upload, product-level charts and a saved history. Reviewlytics addresses that gap with a deliberately simple and free pipeline. It takes the first finding above as its starting point, that review text carries useful signals of deception, and treats the second, that text-only models lose accuracy on real data, as a limit that it must state openly.

3. Methodology

The system follows a pipeline from raw review text to a saved, visualised assessment (Fig. 1). Training happens once during setup; the backend then loads the saved model at start-up instead of retraining on every request.

A. Data Loading and Validation

Pandas reads the Reviews sheet of the Excel file, or an uploaded CSV or XLSX file. Required columns are checked, and alternative names such as text for review_text can be mapped. Ratings are checked to lie between 1 and 5, dates are parsed, and missing values and duplicates are counted and reported rather than deleted silently.

B. Text Preprocessing and Feature Extraction

Review text is lower-cased, trimmed and cleaned of extra whitespace. A TF-IDF vectoriser then turns each review into a weighted word vector, and the fitted vectoriser is saved with Joblib.

C. Classification

A Logistic Regression classifier is trained on labelled reviews with a train/test split. It outputs a probability, which is shown as the suspicion score, and the fitted classifier is saved with Joblib. A linear model on TF-IDF features is fast, needs no paid service and can be explained through its most influential words.

D. Evaluation

Accuracy, precision, recall, F1-score and a confusion matrix are computed on the held-out test set and labelled as test-split results.

E. Sentiment Analysis

VADER computes a compound score for each review. Scores of +0.05 or above are Positive, scores of −0.05 or below are Negative, and the rest are Neutral, which are the thresholds recommended by the VADER authors [7]. Sentiment and authenticity are kept apart: a positive review is not automatically genuine, and a negative review is not automatically fake.

F. Uncertainty Rule

Reviews whose suspicion score falls inside a configurable band around the decision threshold may be shown as Uncertain. If no band is set, the Uncertain series is left out of the charts.

G. Software Architecture

The frontend uses React, Vite, Tailwind CSS, Recharts and Axios. The backend uses FastAPI with Pydantic validation and CORS enabled for the local development server. Predictions and sentiment results are saved in SQLite so that earlier analyses can be revisited.

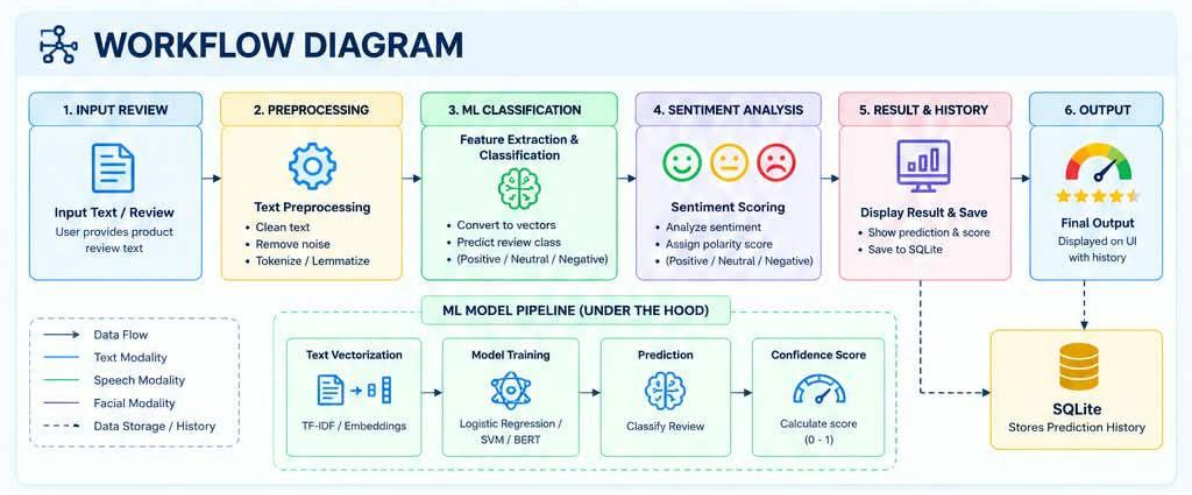


Fig. 1 – Workflow of the Reviewlytics system.

4. Results and Discussion

A. Demonstration Dataset

The supplied Excel file is a synthetic demonstration dataset. Its Read Me sheet states that it was generated for this project, was not collected from real customers, and that its labels are illustrative rather than expert-verified. Table 1 summarises it. The dataset is meant for testing validation, charts, filters, upload and history. Inspecting it also showed that its 5,000 rows are built from only 150 repeated template sentences, and no template appears under both labels.

Table 1. Summary of the demonstration dataset

Property	Value
Review rows	5,000
Products / categories / brands	12 / 7 / 8
Review dates	1 Jan 2025 to 28 Oct 2026
Labels	3,806 Genuine (76.1%), 1,194 Fake (23.9%)
Unique review texts	150 (100 Genuine, 50 Fake)
Rows sharing a text with another row	4,850
Mean rating	4.51 for Fake, 3.92 for Genuine
Sentiment of Fake-labelled rows	All 1,194 Positive
Missing values in text, rating, date	None

B. Baseline Evaluation

As a baseline, TF-IDF features (unigrams and bigrams) with Logistic Regression were trained twice (Table 2). One run used a random stratified 80/20 split. The other kept all rows with the same text on the same side of the split, so no test sentence was seen in training.

Table 2. Baseline evaluation on the synthetic dataset (test-split results)

Split	Test rows	Accuracy	Precision (Fake)	Recall (Fake)	F1 (Fake)	Confusion matrix errors
Random 80/20	1,000	1.000	1.000	1.000	1.000	0
Grouped by text	986	1.000	1.000	1.000	1.000	0

A perfect score does not mean the model is perfect. The Fake and Genuine templates differ in obvious wording, so even a simple model separates them. These figures show that the training and evaluation code works. They say nothing about accuracy on real customer reviews and must not be reported as real-world performance.

C. Sample Output

Fig. 2 shows the single-review analysis screen. For a review that is all praise and no detail, the system displays a Potentially Suspicious label, a suspicion score of 95.1%, a separate Positive sentiment score of 0.826, the contributing phrases and a plain-language explanation that restates the model-generated nature of the result. By design, the classification chart counts predicted labels per product or category, and the trend chart groups predicted labels by real review dates. Both read the model's predictions, never the original label column.

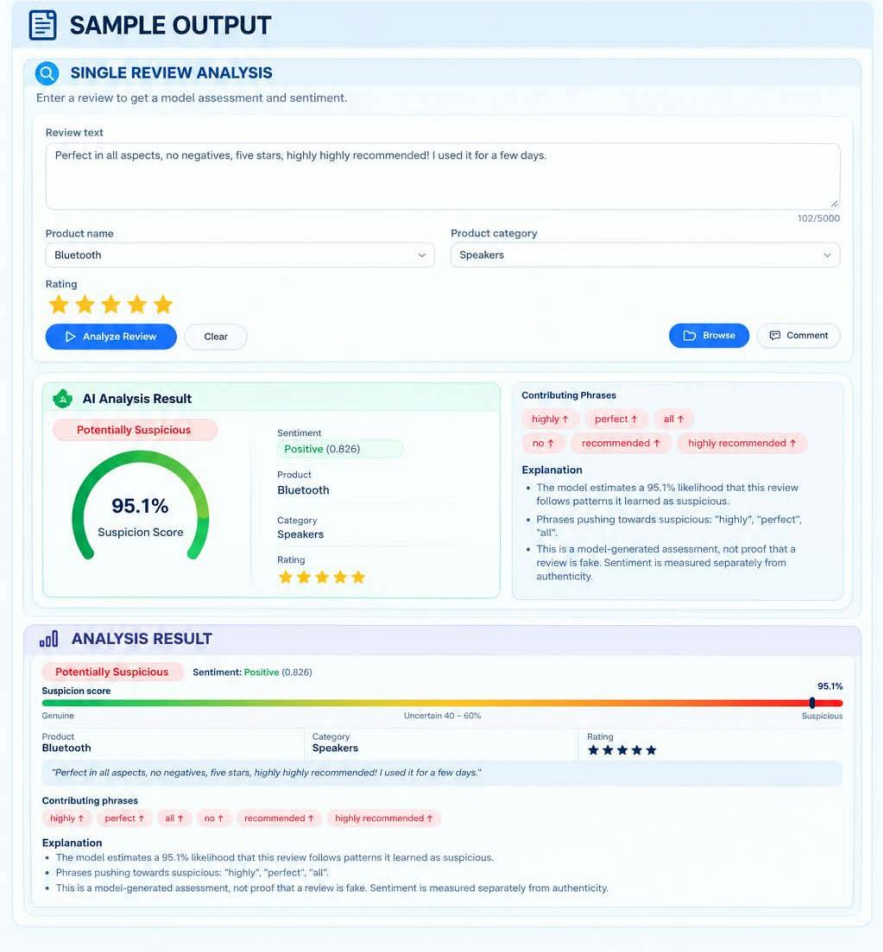


Fig. 2 – Sample output of the single-review analysis.

D. Discussion and Limitations

Reviewlytics shows that a free, local stack can cover the full path from a review file to a saved, visual assessment. The main limitation is the data. The demonstration set has 150 distinct sentences and labels that follow simple template patterns, so it cannot show how the model behaves on real reviews, which are varied and in which skilled fake writers imitate genuine style. The model also reads text only. It does not see reviewer history, posting bursts or purchase verification, which are known to help in practice [3].

There is also an ethical side. A suspicion score can harm an honest reviewer or a seller if it is treated as a verdict. For this reason the interface says potentially suspicious, shows the score, and leaves the final decision to a person. Sentiment is reported separately so that a positive tone is never read as evidence of fakery. Finally, metrics from the synthetic set confirm only that the code runs; a final evaluation needs a properly sourced and documented labelled dataset, which the system supports by allowing the demonstration file to be replaced.

5. Conclusion and Future Scope

Reviewlytics is a practical, free platform for analysing customer reviews of electronics. It joins TF-IDF and Logistic Regression classification, VADER sentiment analysis, product-wise and time-based charts, bulk file upload and a persistent history in one application. Its assessments are model-generated and should guide human review, not replace it. Because the demonstration dataset is synthetic and template-based, its results only show that the pipeline works.

Future work includes training and evaluating on a real, documented and labelled review dataset with an independent test set, adding behavioural features such as reviewer history, posting bursts and verified-purchase status, comparing transformer models such as BERT with the linear baseline, supporting more product categories, languages and larger files, improving the phrase-level explanations, and adding model monitoring with retraining when new labelled data arrives.

REFERENCES

1. N. Jindal and B. Liu, “Opinion Spam and Analysis,” Proc. International Conference on Web Search and Data Mining (WSDM), pp. 219–230, 2008.

2. M. Ott, Y. Choi, C. Cardie and J. T. Hancock, “Finding Deceptive Opinion Spam by Any Stretch of the Imagination,” Proc. Annual Meeting of the Association for Computational Linguistics (ACL), pp. 309–319, 2011.

3. A. Mukherjee, V. Venkataraman, B. Liu and N. Glance, “What Yelp Fake Review Filter Might Be Doing?,” Proc. International AAAI Conference on Web and Social Media (ICWSM), pp. 409–418, 2013.

4. G. Salton and C. Buckley, “Term-Weighting Approaches in Automatic Text Retrieval,” Information Processing & Management, vol. 24, no. 5, pp. 513–523, 1988.

5. F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

6. J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” Proc. NAACL-HLT, pp. 4171–4186, 2019.

7. C. J. Hutto and E. Gilbert, “VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text,” Proc. International AAAI Conference on Web and Social Media (ICWSM), pp. 216–225, 2014.