Truth or Trickery: Detection of Fake Reviews Using Natural Language Processing

Ismail Peracha
California State University, Fullerton
ismailperacha@csu.fullerton.edu

Helen Truong
California State University, Fullerton
htruong122@csu.fullerton.edu

Kanika Sood
California State University, Fullerton
kasood@csu.fullerton.edu

***Abstract*—The prevalence of fake reviews on online platforms poses a serious threat to consumer trust and decision making. This work presents a machine learning approach to classify online reviews as genuine or misleading. We aggregate and preprocess two distinct datasets consisting of authentic and fake reviews, applying feature engineering techniques such as TF-IDF vectorization, dimensionality reduction via Truncated SVD, and outlier filtering. Two models are developed: one using the full feature set and another using a reduced set selected through mutual information. Both models are trained using logistic regression, GXBoost, Naive Bayes, Random Forest, Support Vector Machine, and Decision Tree. Their performance is evaluated based on accuracy and other classification metrics. The results demonstrate that reduced feature selection can maintain comparable accuracy while improving efficiency. This study highlights the effectiveness of combining preprocessing, dimensionality reduction, and machine learning to detect fraudulent review content.**

I. Introduction

Online reviews have become a key factor in shaping consumer decisions. Whether it is purchasing a product, booking a hotel, or choosing a restaurant, users often rely heavily on the opinions of others. However, this reliance has also opened the door to deception. Fake or manipulated reviews are increasingly common, designed to promote low-quality products or unfairly discredit competitors. These reviews erode consumer trust and pose challenges for businesses and platforms striving to maintain integrity.

Detecting fake reviews is not a trivial task. Deceptive content is often designed to appear natural, making it difficult to identify using traditional keyword-based filtering. As a result, recent efforts in machine learning and Natural Language Processing (NLP) have focused on building intelligent systems that can learn to distinguish between authentic and fraudulent reviews based on patterns in text.

In this work, we combine two publicly available datasets of real and fake reviews to build a robust classification model. We apply standard NLP techniques such as TF-IDF vectorization to convert text into numerical form, and then use dimensionality reduction and feature selection to streamline the data for machine learning. Multiple classifiers, including Random Forest, Support Vector Machine (SVM), Naive Bayes, and Decision Tree, are trained and compared in both full and reduced feature sets.

This work aims to evaluate the effectiveness of various models in identifying fake reviews and to understand how preprocessing decisions impact their performance. In doing so, we contribute to the growing field of automatic review authenticity detection, an area that holds practical value for both consumers and businesses. For example, platforms like Amazon could integrate this machine learning pipeline into their review moderation system. As reviews are submitted, they could be vectorized using TF-IDF, processed through a trained classifier, and flagged for further review if identified as potentially deceptive. This enables the real-time detection of fake reviews, helping to maintain consumer trust, reduce manipulation of product rankings, and support more reliable recommendation systems.

II. Background

As online reviews increasingly shape how people shop, the rise of fake or deceptive reviews has become a serious issue. These reviews can mislead buyers, boost ratings unfairly, or harm competitors. That is why the detection of fake reviews has become a growing focus in fields such as machine learning and Natural Language Processing (NLP).

A. Natural Language Processing

Natural Language Processing (NLP) is a branch of Artificial Intelligence that helps computers understand and work with human language. For this work, it allows us to break down reviews into meaningful parts and turn them into something that a machine can analyze. We use techniques like tokenization and TF-IDF vectorization, which helps highlight the most important words in a review by showing how unique they are compared to other reviews. In this way, our models can learn to spot patterns that might hint at whether a review is real or fake.

B. Sentiment Analysis

Sentiment analysis looks at the tone or emotion in a piece of text-whether it is positive, negative, or neutral. Fake reviews often have exaggerated praise or vague language, which can make them stand out. Although our work does not rely solely on sentiment, we use it to support feature selection and help identify overly emotional or suspicious writing that might indicate deception.

C. Text Vectorization via TF-IDF

To convert the raw review text into a numerical format for our machine learning models, we employed Term Frequency-Inverse Document Frequency (TF-IDF) vectorization. This technique measures how important a word is in a single review compared to all other reviews, allowing the models to focus on meaningful words instead of common ones.

We used the Scikit-learn TfidfVectorizer to turn the review text into numbers that the machine learning model can understand. We limit it to the 1,000 most important words and remove common words like 'the', 'and' and 'was' that don't have much meaning. This made the data cleaner and more focused. The final TF-IDF results were then used to train both full and reduced-feature models.

D. Dimensionality Reduction with Truncated SVD

We applied Truncated Singular Value Decomposition (Truncated SVD) to reduce the number of features while keeping the most important information. High-dimensional data can lead to slower training times and may cause models to overfit, especially when features are redundant.

We used Scikit-learn Truncated SVD to reduce the TF-IDF to 100 components. This allowed us to keep the structure and patterns of the data while making the dataset smaller and easier to process. Dimensionality reduction also helped us visualize feature relationships and improved model efficiency without greatly affecting accuracy.

E. Feature Selection via Mutual Information

With the use of feature selection, it improved model performance and reduced overfitting for our model. Specifically, we used mutual information to identify and keep the most relevant features that have the highest dependency with the target labels.

Using Scikit-learn SelectKBest with mutual-info-classif, we selected the top thirty features from the full set. Mutual information measures how much knowing the value of a feature reduces uncertainty about the class label. This can capture more complex and non-linear relationships between input features and the output.

This step helped simplify the data by keeping only the most informative features, removing noise, and redundancy. We then train models using this reduced set to compare their performance against models trained on the full feature set.

III. Approach

To effectively detect fake reviews, we developed a complete machine learning pipeline that involved careful dataset preparation, feature engineering, and model selection. Our approach began with combining and preprocessing two publicly available datasets to create a robust training set. We then extracted features using TF-IDF and refined them using dimensionality reduction and feature selection techniques. Finally, we trained and compared the performance of four different classifiers—Random Forest, Support Vector Machine, Naive Bayes, and Decision Tree—on both full and reduced feature sets. Each model was evaluated using multiple performance metrics, including accuracy, precision, recall, F1 score, and ROC-AUC, to provide a more comprehensive assessment of their effectiveness.

A. Product Review Dataset

We used two publicly available datasets from Kaggle[1] and the UCI Machine Learning Repository[2] to construct our training and testing. The first dataset originates from UCI Amazon Commerce Reviews Set. This dataset includes customer reviews across various Amazon product categories and provides associated metadata, such as review titles and class labels. To prepare this dataset for analysis, we download the original .rar archive[2] , extracted its contents, and converted the .arff file into .csv format using the Weka ARFF[3]. This preprocessing step allowed us to integrate the data into our model. The second dataset, sourced from Kaggle, consists of user-generated product reviews labeled as either genuine or deceptive. These reviews are constructed and labeled to reflect common patterns observed in fake content, making it suitable for our work.

To make our data more balanced, we combined both datasets into one. This helped our models learn from different types of reviews, making them better at spotting fake ones in a wider range of situations.

B. Features

The core features used in our work comes from the text content of product reviews. Since raw text isn't directly suitable for machine learning models, we transformed these reviews into numerical data using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization. This technique helps represent each review based on how important each word is across the entire dataset, filtering out common words like “the” or “and” and emphasizing more distinctive ones.

After applying TF-IDF, each review was converted into a high-dimensional vector. To make the data more manageable[4], we used Truncated SVD to reduce these vectors to 100 components, capturing the most important patterns while simplifying the feature space. To address this, we used two techniques to refine our dataset: Dimensionality Reduction with Truncated SVD: This compressed all of our TF-IDF features into just 100 components while preserving most of the variation in the data. It made training faster and helped reduce noise. Feature Selection with Mutual Information: For certain experiments, we selected the top 30 most informative features based on mutual information with the target label. This helped identify which terms were most useful for distinguishing real reviews from fake ones. Together, these steps ensured that our models learned from meaningful, compressed representations of the reviews, improving both accuracy and efficiency.

C. Classifiers

To evaluate our ability to detect fake reviews, we built four different supervised ML classifiers:

- **Random Forest**: A method that builds multiple decision trees and combines their outputs to improve generalization. It is particularly effective for high-dimensional data and resistant to overfitting, making it ideal for TF-IDF representations.

- **SVM**: A powerful method that seeks an optimal boundary between classes. SVMs are known to perform well in text classification tasks due to their robustness in handling sparse, high-dimensional data.
- **Naive Bayes**: A probabilistic model based on Bayes' Theorem with strong, or naive, independence assumptions. It is simple and fast, and often used as a strong baseline for text classification tasks.
- **Decision Tree**: A rule-based model that splits data based on feature thresholds. Although it can be interpretable, it is prone to overfitting, especially when working with many sparse features such as those from TF-IDF.
- **Logistic Regression**: A linear model that estimates probabilities using a logistic function. Despite its simplicity, it is effective in high-dimensional spaces and serves as a strong baseline for binary classification problems such as fake vs. real reviews.
- **XGBoost (Extreme Gradient Boosting)**: An optimized gradient boosting algorithm that builds models sequentially and focuses on correcting the errors of previous models. Known for its performance and scalability, XGBoost often yields state-of-the-art results in classification tasks, especially with structured or tabular data.

Each of these classifiers was trained twice. Once using all 100 components produced by Truncated SVD (full feature set), and once using a reduced set of 30 features selected via mutual information. This allowed us to compare how well each model performs with a complete versus a compressed representation of the review data.

IV. Experiments & Results

A. Experimental Procedure

We used a combined dataset of real and fake product reviews, consisting of over 41,000 samples labeled as genuine or deceptive. After cleaning and removing duplicates, we vectorized the review text using TF-IDF, a common approach in NLP to weigh word importance in a corpus [5]. To reduce dimensionality, we applied Truncated SVD, following practices shown to improve model efficiency in text-based classification. The target label was encoded using LabelEncoder from scikit-learn, and the dataset was split into training and testing sets using an 80-20 ratio. We used this setup to train several machine learning classifiers.

B. Learner Comparison

To assess the fake review classification task, we evaluated four machine learning models: Random Forest, Support Vector Machine (SVM), Naive Bayes, and Decision Tree Classifier. These models are selected due to their strong track record in natural language processing and text classification tasks [6].

Each model is trained using a feature set derived from TF-IDF vectorization, with dimensionality reduction performed via Truncated SVD. Additionally, we applied mutual information-based feature selection to create reduced versions of the dataset. This allowed us to evaluate whether a smaller, more focused set of features could yield comparable performance.

The accuracy results for each classifier on the test set were as follows:

Classifier	Accuracy	Precision	Recall	F1 Score	ROC-AUC
XGBoost	78.67%	77.28%	80.86%	79.03%	86.98%
Logistic Regression	76.01%	74.23%	79.27%	76.67%	83.61%
Random Forest	76.99%	76.54%	77.45%	76.99%	85.06%
SVM	71.07%	67.49%	80.66%	73.49%	77.28%
Naive Bayes	70.75%	69.33%	73.82%	71.51%	76.14%
Decision Tree	66.31%	66.09%	66.21%	66.15%	77.28%

TABLE I
ACCURACY OF EACH CLASSIFIER ON THE TEST SET.

These results show that XGBoost achieved the highest accuracy and overall performance across all evaluation metrics, demonstrating its strength in handling structured, high-dimensional data. Logistic Regression and Random Forest also performed well, offering a strong balance of precision and recall. In contrast, Naive Bayes and Decision Tree classifiers underperformed relative to the others, likely due to their simplistic modeling assumptions and sensitivity to reduced feature sets. Although SVM achieved high recall, its lower precision contributed to a comparatively lower F1 score.

C. Analysis of Best Performance

Among all the models tested, **XGBoost** delivered the best overall performance. It achieved the highest scores across multiple metrics, including an accuracy of 78.67%, a precision of 77.28%, a recall of 80.86%, an F1 score of 79.03%, and a ROC-AUC of 86.98%. This well-rounded performance indicates that XGBoost is highly effective in balancing false positives and false negatives, making it especially suitable for the detection of fake reviews.

XGBoost's strength lies in its gradient boosting framework, which builds decision trees sequentially and prioritizes the correction of previous errors. This iterative learning process enables it to capture complex patterns in the data more effectively than single estimators or even traditional ensembles like Random Forest. Its robustness against overfitting, particularly in high-dimensional spaces created by TF-IDF vectorization, makes it a powerful choice for text classification tasks.

Furthermore, the model benefits from preprocessing steps such as mutual information-based feature selection and dimensionality reduction, which help remove noise and enhance learning efficiency. These results align with existing literature showing that gradient-boosted models often outperform other classifiers in tasks involving sparse, structured data like textual reviews[7][8].

D. Analysis of Worst Performance

The Decision Tree classifier yielded the lowest accuracy at 66.7%, likely due to its tendency to overfit on training data. Especially when working with high-dimensional, feature spaces generated by TF-IDF.

Its poor performance may also be due to the reduced feature set obtained through mutual information-based feature selection. Although this technique improves efficiency by retaining only the most informative individual features, it may inadvertently exclude valuable feature combinations that decision trees rely on to form effective splits[9]. This limitation is particularly relevant to our work, which involves high-dimensional text data. Where individual words (features) may appear insignificant in isolation but gain predictive power when combined with others.

E. Tuning Parameters

To improve model performance and maintain computational efficiency, we apply several key adjustments during preprocessing and training.

- **TF-IDF Vectorization**: We limit the vocabulary to the top 1,000 words and remove common English stop words[10]. This focuses the model on informative terms and reduces noise from filler words.
- **Feature Selection**: We use mutual information to select the top 30 features for our reduced models. This reduces dimensionality and training time while preserving predictive power.
- **Dimensionality Reduction**: In some preprocessing versions, we apply Truncated SVD to reduce the TF-IDF matrix to 100 components. This transformation simplifies the feature space, making it more manageable for models like SVM and Logistic Regression.

Model Settings

- **Random Forest**: We configure the model with 100 decision trees, which balances accuracy and computational efficiency. This setup is effective for handling high-dimensional data from TF-IDF vectorization.
- **Naive Bayes**: We use the default settings of the Gaussian Naive Bayes classifier, as it often performs well out of the box, especially on text classification tasks[11].
- **SVM**: We apply a linear kernel with default settings. While this configuration is effective for sparse, high-dimensional data, further tuning—such as using different kernels or regularization values—could enhance performance.
- **Decision Tree**: We use default parameters, which prioritize simplicity and interpretability. However, the model may benefit from hyperparameter tuning (e.g., controlling max depth or pruning) to mitigate overfitting.
- **Logistic Regression**: We use scikit-learn's default solver with a maximum of 1,000 iterations to ensure convergence in the high-dimensional feature space. Logistic regression performs well as a linear baseline for binary classification.
- **XGBoost**: We use the default configuration for XGBoost, which already includes regularization to prevent overfitting and automatic handling of missing values. Its gradient boosting framework allows for robust performance even without extensive tuning.

These adjustments help streamline our models and align with standard practices in text classification[12].

V. Conclusion

In this work, we developed a machine learning pipeline to detect fake product reviews by combining Natural Language Processing techniques, dimensionality reduction, feature selection, and multiple classifiers. We utilized two publicly available datasets containing both real and fake reviews, enabling us to build a diverse and balanced training dataset.

TF-IDF vectorization was used to transform review text into numerical features suitable for machine learning. To improve efficiency and reduce overfitting, we applied Truncated SVD and mutual information-based feature selection, allowing the models to focus on the most relevant signals in the data.

We evaluated six machine learning classifiers—Random Forest, SVM, Naive Bayes, Decision Tree, Logistic Regression, and XGBoost—on both full and reduced feature sets. Among these, XGBoost achieved the highest accuracy and overall performance, demonstrating its strength in handling high-dimensional, sparse text data. In contrast, the Decision Tree model showed the weakest performance, likely due to its sensitivity to feature selection and tendency to overfit.

Our results highlight that thoughtful preprocessing and feature engineering significantly impact model performance in text classification tasks. Reduced-feature models performed comparably to full-feature ones, emphasizing the importance of simplification without substantial loss in accuracy. Overall, this study presents a practical and scalable approach to identifying fake reviews, contributing to improved trust and transparency on online platforms.

Future Work

While our approach produced strong results, there are several opportunities for improvement:

- **Parameter Tuning:** Further experimentation with hyperparameter tuning—especially for models like XGBoost and SVM—could improve accuracy and generalization.
- **Dataset Diversity:** Our dataset, while diverse, may still generalize certain review patterns. Future work could explore more domain-specific or multilingual review data.
- **Feature Expansion:** Including additional features such as review length, use of exclamations, or reviewer metadata could help models detect subtle patterns in fake reviews.
- **Deep Learning Models:** Exploring advanced deep learning techniques like LSTMs or transformer-based models (e.g., BERT) may yield further performance gains.
- **Real-World Deployment:** A potential next step involves integrating this model into a web extension that detects suspicious reviews in real-time on platforms like Amazon or Yelp.

References

[1] M. Mexwell, "Fake Reviews Dataset," *Kaggle*, 2023. [Online]. Available: https://www.kaggle.com/datasets/mexwell/fake-reviews-dataset/data. [Accessed: Apr. 15, 2025].

[2] University of California, Irvine, "Amazon Commerce Reviews Set," *UCI Machine Learning Repository*. [Online]. Available: https://archive.ics.uci.edu/dataset/215/amazon+commerce+reviews+set. [Accessed: Apr. 15, 2025].

[3] Weka, “Weka ARFF to CSV (file process framework 20170212),” [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/. [Accessed: Apr. 15, 2025].

[4] M. W. Berry, S. T. Dumais, and G. W. O’Brien, “Using linear algebra for intelligent information retrieval,” *SIAM Review*, vol. 37, no. 4, pp. 573–595, 1995.

[5] A. Karpowicz et al., “The Naive Bayes method outperforms more powerful classification methods in a variety of veterinary clinical datasets,” *J. Dairy Sci.*, vol. 102, no. 12, pp. 11262–11272, 2019.

[6] Y. Yang and J. O. Pedersen, “A comparative study on feature selection in text categorization,” in *Proc. 14th Int. Conf. Machine Learning*, 1997, pp. 412–420.

[7] L. Gutierrez-Espinoza, F. Abri, A. Siami Namin, K. S. Jones, and D. R. W. Sears, “Fake Reviews Detection through Ensemble Learning,” *arXiv preprint arXiv:2006.07912*, 2020.

[8] S. B. Kotsiantis, “Decision trees: a recent overview,” *Artificial Intelligence Review*, vol. 39, no. 4, pp. 261–283, 2013.

[9] A. McCallum and K. Nigam, “A comparison of event models for Naive Bayes text classification,” in *Proc. AAAI-98 Workshop on Learning for Text Categorization*, vol. 752, pp. 41–48, 1998.

[10] V. Sugumaran, V. Muralidharan, and K. I. Ramachandran, “Feature selection using Decision Tree and classification through Proximal Support Vector Machine for fault diagnostics of roller bearing,” *Mech. Syst. Signal Process.*, vol. 21, pp. 930–942, 2007.

[11] E. K. Bjarkason, O. J. Maclaren, J. P. O’Sullivan, and M. J. O’Sullivan, “Randomized truncated SVD Levenberg-Marquardt approach to geothermal natural state and history matching,” *Water Resources Research*, vol. 54, no. 3, pp. 2376–2404, 2018.

[12] S. Sarica and J. Luo, “Stopwords in technical language processing,” *PLOS ONE*, vol. 16, no. 8, e0254937, 2021.