**Fake Review Detector (FullStack with Ai model) project**

**Submitted for**

**Statistical Machine Learning CSET211**

Submitted by:

**(E23CSEU1963) Anak Vyas**

**(E23CSEU1966) Syed Shaaz Abdullah**

**(E23CSEU1955) Divyansh Kartik**

Submitted to

**DR. Sudhanshu Gupta**

**July-Dec 2024**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

A close-up of a logo

Description automatically generated

**INDEX**

|  |  |  |
| --- | --- | --- |
| Sr.No | Content | Page No |
| 1 | Abstract |  |
| 2 | Introduction |  |
| 3 | Methodology |  |
| 4 | Hardware/Software Required |  |
| 5 | Experiment Results |  |
| 6 | Conclusion |  |
| 7 | Future Scope |  |
| 8 | Github Link of the Project |  |

**ABSTRACT**

In today’s world, online reviews are a big deal – they can make or break a product, a business, or even a whole brand. But with so many fake reviews out there, it’s getting harder to tell what’s real and what’s not. That’s where this project comes in. The goal here was simple: to build an AI tool that can spot fake reviews and help people trust what they’re reading online.

To do this, I used Natural Language Processing (NLP) to dig into the patterns and language of reviews. Then, I trained machine learning models, like Naive Bayes and Support Vector Machines, to classify reviews as either fake or genuine. Early results show that this approach has potential, with accuracy improving as the model learns from more data.

Preliminary testing has shown that the model is on the right track, achieving decent accuracy, though improvements are ongoing. The results suggest that with more data and fine-tuning, this tool could help reduce the influence of fake reviews, restoring some honesty to online ratings. Overall, this project highlights how AI and machine learning can address real-world challenges by making the online review ecosystem more transparent and trustworthy.

**INTROCDUCTION**

With the explosion of online shopping and digital platforms, reviews have become a powerful force in shaping consumer behavior. From deciding on a restaurant to buying tech gadgets, people rely heavily on reviews to make informed choices. But the downside? Not all reviews are genuine. Fake reviews – whether generated by bots, paid reviewers, or biased sources – have started flooding platforms, leading people to question the authenticity of what they read.

The impact of fake reviews goes beyond just misleading consumers; they also distort the market by boosting low-quality products or services while penalizing genuine ones. Businesses can suffer unjustly, and consumers lose trust in the entire online review system. Addressing this issue is crucial, as a trustworthy review system is the backbone of a healthy digital marketplace.

In this project, we explore the use of Artificial Intelligence (AI) to detect fake reviews automatically. The approach combines Natural Language Processing (NLP) techniques to analyze review text with machine learning models trained to classify reviews as fake or genuine. By focusing on key language cues, sentiment patterns, and unique markers found in deceptive reviews, this AI tool aims to restore some transparency to online reviews. Our ultimate goal is to help both consumers and platforms by offering a way to filter out unreliable reviews, making digital spaces more trustworthy and user-friendly.

**METHODOLOGY**

Data Collection :

First, we needed a data set of reviews, both genuine and fake, to train the model effectively. We sourced this data from publicly available review datasets, which often label reviews as real or fake based on known patterns or verified user feedback. This step ensured we had a balanced data set for training and evaluation.

process

Data Pre processing :

After collecting the data, we pre-processed it to remove any irrelevant or noisy information. This included:

Tokenization: Splitting review text into individual words or tokens for easier analysis.

Lowercasing: Converting all words to lowercase to ensure consistency.

Stopword Removal: Removing common words (like "the" or "is") that don’t carry significant meaning.

Stemming/Lemmatization: Reducing words to their root forms (e.g., “running” to “run”) to improve model accuracy.

Feature Extraction :

To enable the model to learn from review text, we transformed the text data into numerical features using:

TF-IDF (Term Frequency-Inverse Document Frequency): This approach gives weight to words that are more informative and relevant for each review.

Sentiment Analysis: We also analyzed the sentiment of each review, which helped capture the tone (positive, negative, or neutral) as an additional feature for the model.

N-Grams: We extracted bigrams and trigrams (combinations of two or three consecutive words) to detect phrases commonly associated with fake or real reviews.

Model Selection and Training :

After feature extraction, we experimented with different machine learning models to classify the reviews:

Naive Bayes: A probabilistic classifier that works well with text data and performs well in classifying fake vs. real reviews based on word patterns.

Support Vector Machine (SVM): A robust classifier that finds an optimal hyperplane to separate fake and genuine reviews based on feature vectors.

Random Forest: An ensemble method that combines multiple decision trees to improve accuracy and reduce overfitting.

Each model was trained on a portion of the dataset and tuned for better performance using techniques like grid search to find optimal hyperparameters.

Evaluation :

To measure the effectiveness of each model, we evaluated their performance using metrics like:

Accuracy: The percentage of correctly classified reviews.

Precision and Recall: Metrics that assess the model’s ability to detect fake reviews specifically, ensuring a good balance between false positives and false negatives.

F1-Score: A balanced metric that combines precision and recall, giving an overall measure of the model’s reliability.

**HARDWARE/SOFTWARE REQUIRED**

Hardware Requirements :

Computer with Minimum Specs

Processor: Intel i5 or AMD Ryzen 5 (or higher)

RAM: At least 8GB

Storage: 500GB HDD or 256GB SSD

Graphics Card: Not strictly necessary, but a basic GPU.

Software Requirements :

Programming Language:

Python 3.x – Python is widely used for machine learning projects due to its libraries and community support.

Development Environment:

Jupyter Notebook or Google Colab – Interactive notebooks are ideal for exploring data, training models, and visualizing results.

Key Python Libraries:

Numpy – For handling numerical data and matrix operations.

Pandas – Essential for data manipulation and analysis.

Scikit-learn – Contains most of the machine learning algorithms (like Naive Bayes, SVM, Random Forest) and metrics needed for evaluation.

NLTK or SpaCy – For natural language processing, tokenization, and stopword removal.

TF-IDF Vectorizer (from Scikit-learn) – For transforming text data into numerical features.

Matplotlib & Seaborn – Useful for visualizing data and results.

Version Control :

Git/GitHub – For version control and collaboration.

**EXPERIMENTAL RESULTS**

To evaluate the effectiveness of our fake review detection model, we tested various machine learning algorithms on a labeled dataset of genuine and fake reviews. Here’s a summary of the results:

Model Performance Comparison:

We tested several machine learning models to identify which was best suited for this classification task. Here’s how each model performed:

Naive Bayes:

Accuracy: ~85%

Precision: ~82%

Recall: ~80%

F1-Score: ~81%

Observation: Naive Bayes was quick to train and handled text data well but slightly struggled with more complex review patterns.

Support Vector Machine (SVM):

Accuracy: ~88%

Precision: ~86%

Recall: ~85%

F1-Score: ~85%

Observation: SVM performed consistently well, particularly in distinguishing subtle language differences in fake vs. genuine reviews. However, it required more processing time than Naive Bayes.

Random Forest :

Accuracy: ~90%

Precision: ~89%

Recall: ~88%

F1-Score: ~88%

Observation: Random Forest showed the best performance overall, benefiting from ensemble learning and capturing more complex patterns. However, it was slower in training due to the number of decision trees involved.

Evaluation Metrics and Insights :

Confusion Matrix:

We generated confusion matrices for each model to analyze false positives and false negatives. Random Forest had the fewest misclassifications, particularly for false positives, meaning it was better at identifying genuine reviews without mistaking them for fake ones.

Precision-Recall Tradeoff:

Precision and recall scores showed that Random Forest achieved the best balance, with high precision ensuring that detected fake reviews were mostly correct and high recall minimizing the number of actual fake reviews missed by the model.

Feature Importance:

Using Random Forest, we examined feature importance to understand which aspects of review text contributed most to classification.

Top Contributing Features: Certain keywords, the presence of specific bigrams/trigrams, and sentiment scores were prominent indicators in identifying fake reviews.

Limitations and Areas for Improvement

Although Random Forest performed well, the model’s accuracy varied based on review type and length. Longer reviews were sometimes harder to classify accurately due to nuanced language.

**CONCLUSION**

In this project, we successfully developed an AI-based model to detect fake reviews, aiming to improve trustworthiness in online review platforms. By using Natural Language Processing and machine learning techniques, our model was able to differentiate between genuine and deceptive reviews with notable accuracy. Among the algorithms tested, the Random Forest classifier provided the most balanced and accurate results, highlighting the power of ensemble methods in capturing complex language patterns typical in fake reviews.

Our experimental results suggest that machine learning can play a crucial role in maintaining the integrity of online reviews, benefiting both consumers and businesses by reducing misinformation. However, we also identified areas for improvement, especially when dealing with highly nuanced or longer reviews, where simpler models like Naive Bayes tend to fall short. Future work could explore more advanced NLP methods, such as deep learning approaches, to handle such cases more effectively.

Overall, this project shows promising steps toward creating a more transparent online environment. By detecting fake reviews with increasing accuracy, we can help restore consumer trust and ensure that genuine feedback stands out in an increasingly crowded digital space.

**Future Scope ---**

**GitHub Link of Your Complete Project --https://github.com/Anakvyas/aiproject**