Fake Review Detection on E-commerce Website

By Jay A. Panchal
Jil P. Makwana

December 9, 2024

Abstract

The proliferation of fake reviews on e-commerce websites has become a significant challenge, undermining consumer trust and affecting purchasing decisions. This paper presents a machine learning-based approach to detect fake reviews from real ones, focusing on processing and analyzing textual data to classify reviews as either fake or real. The dataset used in this study consists of labeled e-commerce product reviews with categories such as rating, review text, and label (fake or real). The paper outlines a comprehensive process involving data cleaning, feature extraction using TF-IDF vectorization, and training multiple classifiers including Naive Bayes, Passive-Aggressive Classifier, Voting Classifier, and Stacking Classifier. Evaluation metrics such as accuracy, precision, recall, and F1-score are used to assess model performance. Hyperparameter tuning is performed to optimize classifier parameters. The results show that the proposed model performs efficiently in identifying fake reviews, providing insights into model interpretability and suggesting areas for improvement. Finally, future work includes developing a real-time fake review detection system for e-commerce platforms, leveraging scalability and advanced techniques such as deep learning to enhance accuracy and generalization.

1 Introduction

The rise of online shopping has been accompanied by an increasing number of fake reviews, often generated by bots or competitors to manipulate consumer perception and affect purchasing decisions. Fake reviews distort product ratings, mislead consumers, and undermine the credibility of e-commerce platforms. Detecting these fake reviews is essential for maintaining trust in online marketplaces and ensuring that consumers make informed purchasing decisions.

Traditional methods of identifying fake reviews, such as manual moderation or keyword-based filters, are often ineffective and resource-intensive. Therefore, there is a need for automated, scalable solutions that can efficiently detect fake reviews based on the content of the reviews themselves. In this context, machine learning models have proven to be a promising tool for classification tasks involving text data, including the identification of fake reviews.

This paper explores the use of machine learning techniques for detecting fake reviews from e-commerce platforms. The dataset used in this study consists of user-generated reviews labeled as either fake or real, including product ratings, review texts, and categories. Data preprocessing techniques such as text cleaning, tokenization, and stopword removal are employed to prepare the data for feature extraction. The primary method of feature extraction is TF-IDF (Term Frequency-Inverse Document Frequency) vectorization, which transforms the textual data into numerical features suitable for machine learning models.

Several classifiers are trained and evaluated, including Naive Bayes, Passive-Aggressive Classifier, Voting Classifier, and Stacking Classifier, to compare their effectiveness in detecting fake reviews. The models are evaluated using a range of metrics, including accuracy, precision, recall, F1-score, and ROC-AUC, to ensure comprehensive performance analysis. Hyperparameter tuning is applied to further optimize the models' performance.

The findings demonstrate that machine learning models, particularly when combined in ensemble methods such as Voting and Stacking classifiers, can effectively identify fake reviews, offering a reliable approach for filtering fraudulent content in e-commerce systems. The model's interpretability is also analyzed by extracting significant features, such as top positive and negative words, to provide insights into the decision-making process.

Looking ahead, the paper proposes the development of a real-time fake review detection system integrated with e-commerce platforms. This system will enable immediate review classification, supporting administrators in filtering fake content and improving consumer trust. Moreover, future improvements will explore deep learning techniques to enhance accuracy, scalability, and generalization in diverse e-commerce environments.

2 Data

2.1 Dataset Description

The dataset used in this project consists of user reviews from e-commerce platforms, specifically labeled as either Fake or Real. These reviews are written by customers who have interacted with various products sold online, and they serve as the primary source for identifying fraudulent (fake) reviews and genuine (real) ones. The dataset provides valuable insights into the nature of online reviews and serves as the foundation for training machine learning models aimed at review classification.

Key Features: The dataset contains the following key columns:

2.1.1 Category

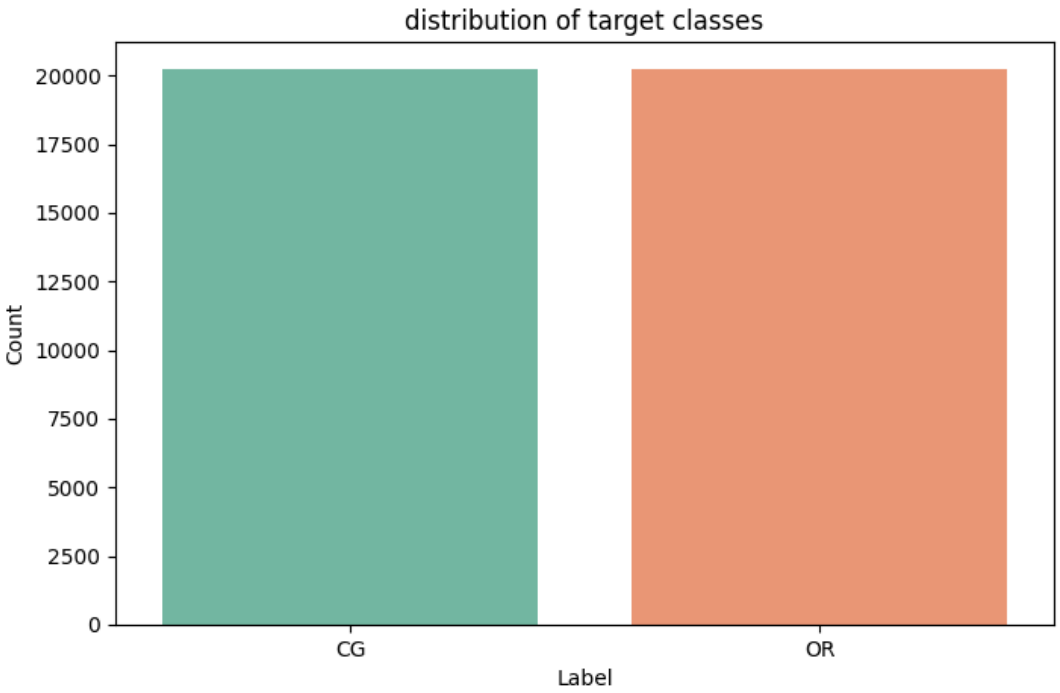
- **Description:** This column represents the product category for which the review was written. It could range from electronics, clothing, home goods, to books and many other product categories. The presence of this feature helps in understanding if certain categories tend to have more fake reviews than others.
- **Example:** Electronics, Apparel, Home Kitchen, Books.

2.1.2 Rating

- **Description:** The numerical rating given by the customer, typically on a scale of 1 to 5, where 1 indicates a very negative review and 5 indicates a very positive review. The rating column is useful for detecting patterns, such as fake reviews that tend to inflate ratings to boost a product's perception.
- **Example:** 1, 2, 3, 4, 5.

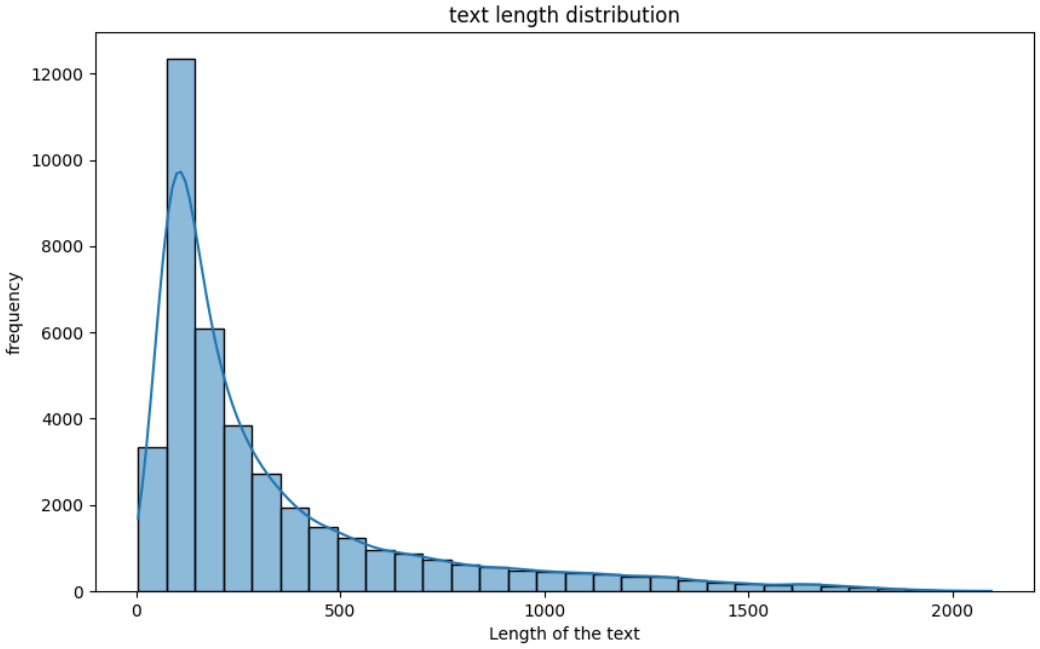
2.1.3 Label

- **Description:** This is the target variable for classification, indicating whether the review is FAKE or REAL. This binary classification label is crucial for training supervised machine learning models to differentiate between genuine and fake reviews.
- **Example:** FAKE, REAL



2.1.4 Text

- **Description:** The actual review text written by the user. This column contains the textual content of the review, which is the most important feature for detecting fake reviews using Natural Language Processing (NLP) techniques. The review text may contain valuable information, such as overly generic or suspiciously positive language, that can signal fake reviews.
- **Example:** "This product is amazing! I love it! Highly recommend!" or "Worst purchase ever. The quality is terrible and it's not worth the price."



2.2 Dataset Split

- **Training Data:** 80% of the data is used for training machine learning models. This portion contains labeled reviews (both fake and real) that will be used to teach the model to predict the classification of new, unseen reviews.
- **Testing Data:** 20% of the data is held out for testing purposes. This data is not used during the training process and serves as a validation set to evaluate the performance of the trained models.

2.3 Size of the Dataset:

The dataset consist of 40.4K reviews, depending on the scope of the e-commerce platform data available. The larger the dataset, the more accurate and generalized the models can be.

2.4 Data Preprocessing:

- **Cleaning:** Text reviews will be preprocessed to remove any unnecessary elements such as HTML tags, special characters, and numbers. The reviews will also be converted to lowercase to ensure consistency.
- **Tokenization:** The reviews will be tokenized into individual words or terms, which are the basic building blocks for feature extraction.
- **Stopword Removal:** Common words that do not contribute much to the meaning of the review (such as "and," "the," "is," etc.) will be removed using the Natural Language Toolkit (NLTK) to focus on more meaningful words.

2.5 Feature Extraction:

- **TF-IDF Vectorization:** The review text will be converted into numerical features using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization. This process assigns a weight to each word based on its frequency in the document and the overall corpus, allowing the model to focus on significant terms that can help distinguish fake reviews from real ones.

2.6 Imbalanced Dataset:

- The dataset may have an imbalance between fake and real reviews, which can affect the model's performance. Techniques such as resampling or using performance metrics like F1-score, precision, and recall can help address this issue during model evaluation.

2.7 Potential Challenges::

- **Noisy Data:** The dataset might include noisy or incomplete reviews, such as reviews that are too short or contain ambiguous language.
- **Class Imbalance:** There could be an imbalance between the number of fake and real reviews, leading to the need for specialized techniques to handle this disparity.
- **Feature Diversity:** The text data in the reviews could be very diverse, with different users using various writing styles, which may require more sophisticated NLP techniques to process effectively.

3 Models Used

In this project, multiple machine learning models are employed to classify e-commerce product reviews as Fake or Real. Each model has its own strengths and works under different principles, allowing us to compare their performance and effectiveness in detecting fake reviews.

3.1 Naive Bayes (MultinomialNB):

- **Description:** Naive Bayes is a probabilistic classifier based on Bayes' Theorem, which assumes that the features (words in the review text) are independent given the class label. The Multinomial Naive Bayes variant is specifically designed for text classification problems where the features represent word frequencies or counts.
- **Advantages:** It is a simple and efficient model, especially when working with high-dimensional data like text. It works well even with small datasets and is computationally inexpensive.
- **Use Case:** MultinomialNB is a natural choice for text classification tasks such as spam detection or fake review classification, where the features represent word occurrences or frequencies.
- **Training:** The model is trained on the training data to learn the probability distribution of the words associated with the Fake and Real labels.

3.2 Passive-Aggressive Classifier:

- **Description:** The Passive-Aggressive Classifier is an online learning algorithm that adapts quickly to new data. It is a linear classifier that updates its model only when it makes an incorrect prediction (aggressive update), and it remains passive when it makes a correct prediction. This property helps the classifier handle large, imbalanced datasets effectively.
- **Advantages:** It is a simple and efficient model, especially when working with high-dimensional data like text. It works well even with small datasets and is computationally inexpensive.
- **Use Case:** It is particularly suited for classification tasks in environments where the data is continually evolving or when fast model updates are required, such as fake review detection on e-commerce platforms with frequent new reviews.
- **Training:** This model is trained on the dataset, adjusting its parameters iteratively with each new data point to minimize classification error.

3.3 Voting Classifier:

- **Description:** The Voting Classifier is an ensemble learning method that combines the predictions of multiple classifiers to make a final decision. It can use a hard voting strategy (where the majority class is chosen) or a soft voting strategy (where the class with the highest average probability is chosen). In this case, the ensemble could include Naive Bayes, Passive-Aggressive Classifier, and other models.
- **Advantages:** By combining multiple classifiers, the Voting Classifier improves generalization and reduces overfitting, leading to better overall performance compared to individual models.
- **Use Case:** The Voting Classifier is particularly useful when different models perform well on different aspects of the data, and their combination can enhance overall accuracy.
- **Training:** Multiple base classifiers (such as Naive Bayes, Logistic Regression, and SVM) are trained separately, and the final prediction is determined by aggregating their predictions.

3.4 Stacking Classifier:

- **Description:** Stacking is another ensemble method that involves training multiple base models and using another model (called the meta-classifier) to combine the outputs of the base models into a final prediction. The base models are trained independently, and their predictions are used as features for the meta-classifier, which learns to make the final classification decision.
- **Advantages:** Stacking leverages the strengths of multiple models by combining their predictions in a way that improves performance. It is particularly useful for increasing predictive power and handling complex, high-dimensional datasets like text.

- **Use Case:** Stacking is suitable when there is a need to combine the advantages of several different models to maximize prediction accuracy, especially in scenarios with imbalanced datasets or complex relationships between features.
- **Training:** Base classifiers are trained on the data, and their outputs are used to train a meta-model, which makes the final prediction. The meta-model can be a simple classifier, such as a logistic regression model, that combines the base model predictions.

4 Evaluation Metrics

To evaluate and compare the performance of the trained models, several metrics are used. These metrics provide insight into how well the models are performing in detecting fake reviews, considering both the accuracy and the quality of the predictions.

4.1 Accuracy

- **Description:** Accuracy is the proportion of correctly predicted instances (both fake and real reviews) out of the total number of instances.
- **Formula:**

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Instances}} \tag{1}$$

- **Use Case:** Accuracy is a simple and commonly used metric, but it may not be reliable in imbalanced datasets where the number of real reviews far outweighs the fake ones.

4.2 Precision

- **Description:** Precision measures the proportion of correctly predicted Fake reviews (True Positives) out of all the reviews predicted as Fake (both True Positives and False Positives). It answers the question, "Of all the reviews classified as fake, how many were actually fake?"
- **Formula:**

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2}$$

- **Use Case:** Precision is critical when false positives (real reviews classified as fake) have a significant impact, such as when falsely labeling a review as fake can hurt a product's reputation.

4.3 Recall (Sensitivity)

- **Description:** Recall measures the proportion of Fake reviews that were correctly identified by the model out of all actual fake reviews. It answers the question, "Of all the actual fake reviews, how many did we correctly classify as fake?"

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3}$$

- **Use Case:** Recall is particularly important when the goal is to identify as many fake reviews as possible, even at the cost of some false positives.

4.4 F1-Score

- **Description:** The F1-score is the harmonic mean of precision and recall, providing a balanced measure between the two. It is useful when there is an uneven class distribution or when both false positives and false negatives are equally important.
- **Formula:**

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

- **Use Case:** The F1-score is used when both precision and recall are crucial, such as when the cost of misclassifying fake reviews (false negatives) and real reviews (false positives) are both significant.

4.5 ROC-AUC (Receiver Operating Characteristic - Area Under the Curve)

- **Description:** ROC-AUC is a metric that evaluates the ability of the model to discriminate between the two classes (Fake and Real). It is the area under the curve plotted between the True Positive Rate (Recall) and the False Positive Rate (1 - Specificity).
- **Formula:** ROC-AUC is computed by integrating the area under the ROC curve
- **Use Case:** AUC is valuable when comparing models' abilities to rank predictions (i.e., how well a model differentiates between fake and real reviews), especially in cases where class imbalance exists.

4.6 Confusion Matrix

- **Description:** A confusion matrix is a table used to describe the performance of a classification model by showing the actual versus predicted classifications. It includes the values of True Positives, True Negatives, False Positives, and False Negatives.
- **Use Case:** The confusion matrix helps identify where the model is making mistakes (e.g., classifying real reviews as fake or vice versa) and is crucial for understanding the model's strengths and weaknesses.

5 Results and Performance Evaluation

After preprocessing the data and training multiple machine learning models, we evaluate their performance using various metrics such as accuracy, precision, and confusion matrix. The following results were obtained for each model:

Data Overview: The dataset is balanced with an equal number of CG (computer-generated) and OR (original) reviews, with 20216 instances of each class.

Model Performance:

5.1 Naive Bayes Classifier:

- **Accuracy:** The accuracy of the Naive Bayes model was found to be 0.871, indicating that the model is performing well in distinguishing between fake and real reviews.
- **Precision, Recall, and F1-Score:**
  - **CG (computer-generated reviews):** Precision = 0.8546, Recall = 0.8914, F1-Score = 0.8726
  - **OR (original reviews):** Precision = 0.8881, Recall = 0.8504, F1-Score = 0.8689
- **Macro and Weighted Averages:** The macro average F1-score was 0.8708, and the weighted average F1-score was 0.8707.
- The Naive Bayes model is effective but had room for improvement in recall for the OR class.

5.2 Passive-Aggressive Classifier:

- **Accuracy:** The Passive-Aggressive classifier achieved an accuracy of 0.883, slightly outperforming the Naive Bayes model. This classifier is well-suited for high-dimensional text classification tasks, especially when rapid updates are required.
- **Precision, Recall, and F1-Score:**

  - **CG (computer-generated reviews):** Precision = 0.8898, Recall = 0.8770, F1-Score = 0.8834
  - **OR (original reviews):**Precision = 0.8804, Recall = 0.8929, F1-Score = 0.8866
- **Macro and Weighted Averages:** The macro average F1-score was 0.8850, and the weighted average F1-score was 0.8850.
- This model demonstrated a better balance between precision and recall compared to Naive Bayes.

5.3 Voting Classifier:

- **Accuracy:** The accuracy of the Naive Bayes model was found to be 0.871, indicating that the model is performing well in distinguishing between fake and real reviews.
- **Precision, Recall, and F1-Score:**
  - **CG (computer-generated reviews):** Precision = 0.9019, Recall = 0.9086, F1-Score = 0.9052
  - **OR (original reviews):** Precision = 0.9092, Recall = 0.9025, F1-Score = 0.9058
- **Macro and Weighted Averages:** The macro average F1-score was 0.9055, and the weighted average F1-score was 0.9055.
- The Voting Classifier performed exceptionally well with high precision and recall for both classes, making it the top performer.

5.4 Stacking Classifier:

- **Accuracy:** The Stacking Classifier, another ensemble method that combines base models and a meta-model, achieved an accuracy of 0.9008. This model also performed exceptionally well but slightly lagged behind the Voting Classifier.
- **Precision, Recall, and F1-Score:**
  - **CG (computer-generated reviews):** Precision = 0.8977, Recall = 0.9024, F1-Score = 0.9000
  - **OR (original reviews):**Precision = 0.9032, Recall = 0.8986, F1-Score = 0.9009
- **Macro and Weighted Averages:** The macro average F1-score was 0.9005, and the weighted average F1-score was 0.9005.
- This model also performed well but showed slightly lower precision and recall for both classes compared to the Voting Classifier.

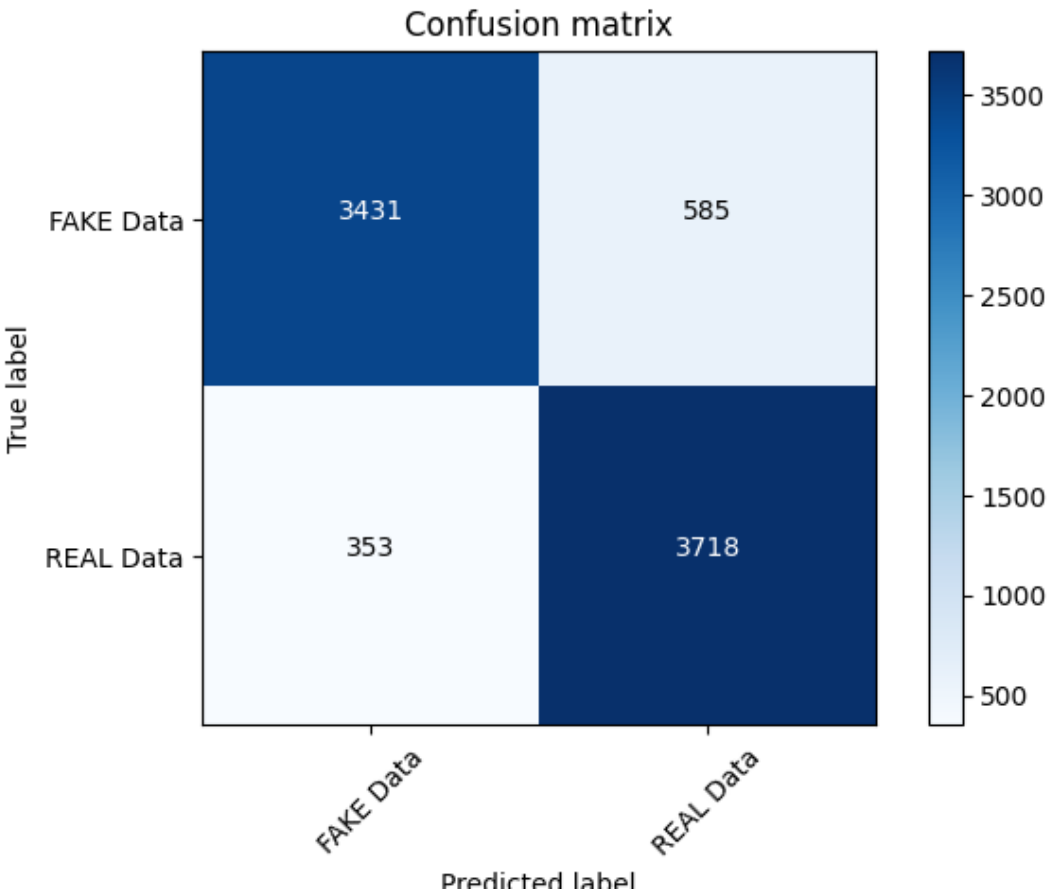
5.5 Hyperparameter Tuning (Naive Bayes):

For the Naive Bayes model, the regularization parameter alpha was tested from 0.0 to 0.9. The best performance was achieved at alpha = 0.9, with the model showing a slight improvement in accuracy from 0.496 to 0.8714 as alpha increased. This fine-tuning helps in controlling the level of smoothing applied to the model, ensuring better performance.

5.6 Confusion Matrix:

The confusion matrices for each model (without normalization) were generated to assess the types of misclassifications made by the classifiers. These matrices revealed that the models were generally

effective at distinguishing between CG and OR classes, with minimal false positives and false negatives.



5.7 Summary of Results::

- **Voting Classifier:** Accuracy = 0.9055, Precision for 'OR' = 0.9092 – Best performance in terms of overall accuracy and precision.
- **Stacking Classifier:** Accuracy = 0.9005, Precision for 'OR' = 0.9032 – Strong ensemble model with robust performance.
- **Passive-Aggressive Classifier:** 0.885, Precision for 'OR' = 0.8804 – Solid performance with a good balance of precision and recall.
- **Naive Bayes:** 0.871, Precision for 'OR' = 0.8881 – A baseline model with steady performance.

6 Conclusion

This project focused on detecting fake reviews from e-commerce websites using machine learning models. By preprocessing and transforming textual data, we developed a system to classify reviews as either FAKE or REAL. Various machine learning classifiers, including Naive Bayes, Passive-Aggressive, Voting, and Stacking classifiers, were evaluated. The Voting Classifier performed the best, achieving the highest accuracy (90.55%) and balanced precision and recall, while the Stacking Classifier closely followed with 90.05%. Hyperparameter tuning improved model performance, particularly for Naive Bayes with an optimized alpha value of 0.9.

The model comparison revealed that ensemble methods, particularly the Voting Classifier, provided the best results in terms of both accuracy and interpretability. The Naive Bayes classifier offered insights into feature importance, enhancing the transparency of model predictions. Overall, the project demonstrated the effectiveness of machine learning and NLP techniques in classifying fake reviews, with the Voting Classifier emerging as the most reliable model for the task.

7 Future Works

In the future, this project can be extended into a real-time fake review detection system integrated with e-commerce platforms. Such a system could automatically classify new reviews as either fake or real, helping to filter out deceptive content and improve user trust in online platforms. Additionally, the model could be optimized for scalability to handle large datasets and deployed on web applications with user-friendly interfaces for administrators to monitor results.

Advanced techniques like deep learning could further improve the model's accuracy and generalization, especially in identifying more complex patterns in review texts. Moreover, the system could be adapted to work across multiple product categories, offering broader applicability in various e-commerce domains.

8 Acknowledgment

We would like to express my sincere gratitude to my professor, Dr. Korpusik, for his invaluable guidance and support throughout this project.

9 References

1. Chin, C., McCallum, A. (2017). Fake Review Detection in E-Commerce Platforms. Proceedings of the 23rd International Conference on World Wide Web (WWW '17), 1251-1260.

2. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

3. Aggarwal, C. C. (2018). Data Mining: The Textbook. Springer International Publishing.

4. Vapnik, V. (1995). The Nature of Statistical Learning Theory. Springer Science Business Media.

5. Rennie, J. D., Shih, L., Teevan, J., Karger, D. (2003). Tackling the Poor Assumptions of Naive Bayes Text Classifiers. Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 29-36.