Amazon Fake Review Detection

***Abstract*—** Online reviews play a key role in shaping consumer decisions, but the rise of fake reviews on platforms like Amazon has made it harder to trust them. In this project, we focus on detecting fake reviews using a combination of text mining and clustering. We performed exploratory data analysis (EDA) and statistical analysis to gain an overview of the data and identify key patterns. Building on these insights, We leverage features from the review text and metadata, integrating word embeddings and Recursive Feature Elimination (RFE) for effective feature selection. To uncover patterns in the data, we apply clustering methods like HDBSCAN and K-means, which allow us to group reviews based on hidden structures. Our analysis provides insights into the behaviors and characteristics of fake reviews, paving the way for more robust review filtering mechanisms. This approach combines cutting-edge methods with practical applications, addressing a critical challenge in the e-commerce ecosystem.

I. INTRODUCTION

Online reviews have become an essential part of modern e-commerce platforms, heavily influencing consumer purchasing decisions. It helps customers make informed choices but also provide valuable feedback to sellers. However, with the increasing reliance on reviews, the rise of fake or deceptive reviews has emerged as a significant challenge. Fake reviews mislead customers, distort product reputations, and undermine the credibility of platforms like Amazon.

As online marketplaces expand, ensuring the authenticity of reviews is critical for maintaining trust and transparency. Traditional approaches to detect fake reviews often rely on simplistic features or rule-based systems, which can be easily bypassed by sophisticated fake review strategies. This underscores the need for more advanced and data-driven methods to identify and combat fake reviews effectively.

Through the application of clustering algorithms like HDBSCAN and K-means, we explore hidden structures in the data to group reviews based on their characteristics. By combining exploratory analysis with ML methodologies, this project provides a comprehensive framework for detecting fake reviews, which contributes to the broader goal of ensuring authenticity and trust in e-commerce platforms.

II.LITERATURE REVIEW

This study utilizes an Amazon reviews dataset containing textual content, user interactions, and metadata, such as timestamps and review details, to detect fake reviews. Similar datasets and methodologies have been explored in research, such as:

Fake Reviews Detection: A Survey [1] this paper highlights datasets like Yelp and TripAdvisor, focusing on user credibility, review patterns, and metadata to detect anomalies in online reviews. Detecting Fake Reviews Using Machine Learning Techniques: A Survey [4] provides insights into datasets and state-of-the-art methods, emphasizing the analysis of textual and behavioral features to distinguish fake reviews.

Fake Review Detection Using Neural Networks [5] Explores the use of machine learning models on datasets like Yelp and Amazon, focusing on textual embeddings and metadata for classification tasks.

These papers emphasize similar datasets and methodologies, aligning with this study's approach to leveraging textual patterns, user behavior, and metadata for fake review detection.

State-of-the-art methods in fake review detection primarily involve Natural Language Processing (NLP), behavioral analysis, and machine learning. NLP techniques like sentiment analysis, subjectivity scoring, and word embeddings (e.g., TF-IDF, BERT) extract valuable textual features, while behavioral features like review frequency, helpfulness votes, and time gaps provide additional context. Machine learning approaches, ranging from traditional models like Random Forests to deep learning architectures, are widely employed to classify reviews and rank feature importance.

This study builds on prior work by combining textual features (e.g., sentiment polarity and subjectivity) with behavioral features (e.g., review frequency and time differences) to improve detection accuracy. While earlier studies often rely heavily on textual analysis, this research demonstrates that integrating behavioral attributes yields a more robust model for identifying fake reviews. This approach contributes to advancing methodologies in fake review detection, emphasizing the importance of a holistic feature engineering framework.

III. EDA AND STATISTICAL ANALYSIS

Exploratory Data Analysis (EDA) was conducted to gain a deeper understanding of the dataset, identify key patterns, and detect potential anomalies that could inform the detection of fake reviews. The dataset, comprising verified purchase reviews and product metadata, was analyzed across various dimensions, including text features, temporal trends, ratings distribution, and review characteristics.

A. Word Frequency Analysis

Word Cloud visualization the most frequently used words in the review text. Words like "love," "use," "product," and "hair" dominated the reviews, reflecting a prevalence of positive adjectives and generic terms. This overuse of enthusiastic language may indicate fake or promotional reviews, especially when paired with specific product categories. This insight underscores the need for further sentiment analysis to identify suspicious review patterns.

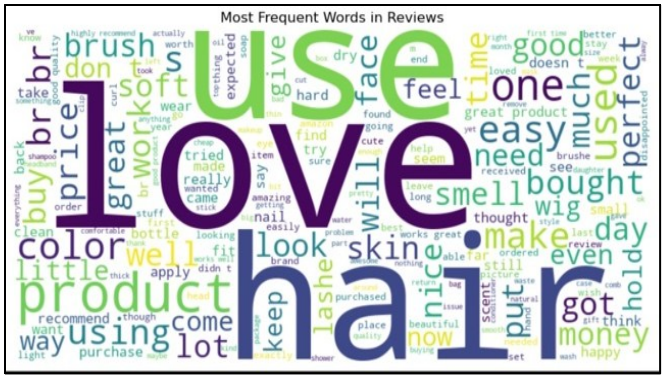

Fig. 1. Word Frequency Analysis

B. Temporal Trends

An analysis of the number of reviews over time revealed a significant spike during the COVID-19 pandemic (March 2020 to June 2021). This period saw a surge in online shopping activity, which likely contributed to an increase in reviews. However, it also raises the possibility of increased fake reviews, as many sellers may have attempted to capitalize on the heightened demand. This anomaly warrants further investigation into review authenticity during this timeframe.

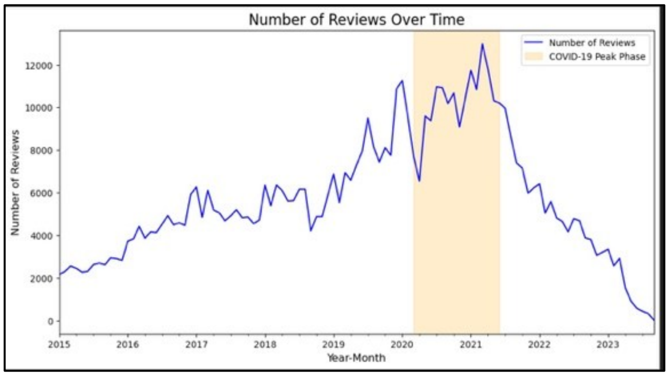

Fig. 2. Temporal Trends

C. Review Length Analysis

The relationship between review length and star ratings was explored using a boxplot. Most reviews were relatively concise, with a few outliers featuring exceptionally long review lengths. These outliers were present across all rating levels and may represent either highly detailed feedback or suspicious, overly descriptive promotional content. Further clustering and anomaly detection are necessary to determine their nature.

D. Ratings Distribution

The distribution of ratings showed a heavy skew toward 5-star reviews, which accounted for most of the dataset. While this could reflect high customer satisfaction, the disproportionate number of perfect ratings and the relatively low occurrence of mid-range ratings (e.g., 3-star reviews) suggest potential manipulation. Additionally, a significant number of 1-star reviews indicate dissatisfaction, creating a polarized pattern of feedback. This polarization highlights the importance of sentiment analysis and clustering to better understand rating authenticity.

E. Data Quality and Preprocessing

We performed several preprocessing steps to clean the data. Duplicates were removed based on a combination of `user_id`, `asin`, `text`, and `parent_asin`. Columns with high percentages of missing data, such as `price` and `bought_together`, were dropped, while columns like `store` were imputed with the value "Other." Features such as `main_category`, which lacked variability, were removed for efficiency.

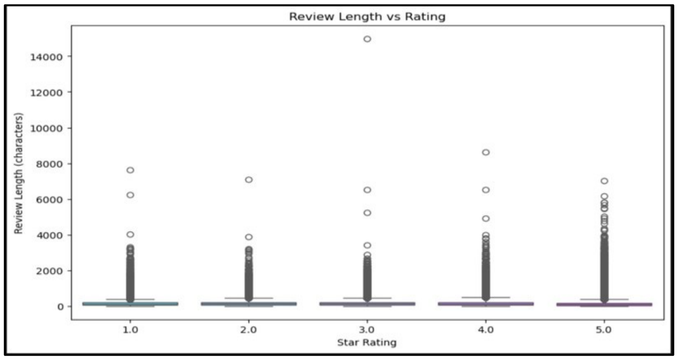

Fig. 3. Review Length VS Rating

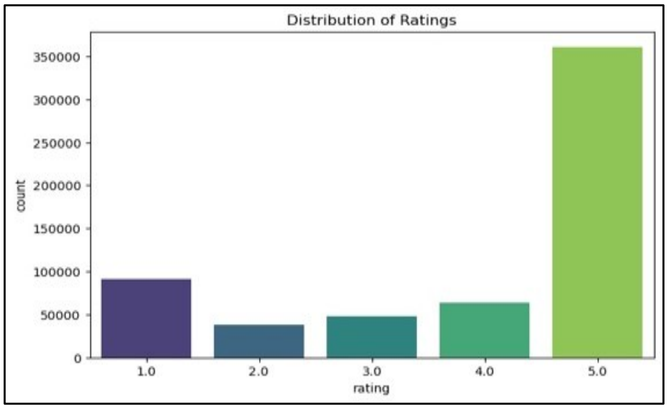

Fig. 4. Rating Distribution

IV. TEXT PROCESSING

To ensure the efficacy of data-driven models for fake review detection, it is essential to preprocess the text data effectively. Raw textual data from reviews often contains noise such as HTML tags, special characters, and irrelevant information, which can hinder analytical performance. Our preprocessing pipeline focuses on transforming raw text into a clean and structured format suitable for machine learning models.

To prepare raw textual data for analysis and modeling, we implemented the following preprocessing steps:

A. HTML Tag Removal

Stripped HTML tags using BeautifulSoup to retain only the core text.

B. Special Characters and Numbers Removal

Used re.sub to remove non-alphanumeric characters and numbers, focusing on meaningful content.

C. Lowercasing

Converted all text to lowercase to ensure uniformity and avoid case-sensitive redundancies.

D. Tokenization

Split text into individual words (tokens) using nltk.word_tokenize for word-level analysis.

E. Stopword Removal

Removed commonly used words (e.g., "and," "the") using NLTK's stopword list to reduce noise.

F. Lemmatization

Reduced words to their base form using WordNetLemmatizer for linguistic consistency (e.g., "running" → "run").

G. Custom Preprocessing for Noisy Data

Cleaned the details column by removing special characters, standardizing keys/values, and handling non-dictionary entries.

H. Cleaned Additional Text Columns

Applied the same pipeline to Review_Title, text, and description columns, creating *_cleaned versions and dropping the noisy originals.

V. ENCODING AND SCALING

The store column was encoded using LabelEncoder to assign unique integers to each category. Numerical features, including helpful_vote, rating, Review_character_count, and others, were standardized using StandardScaler to ensure uniform scaling with a mean of 0 and standard deviation of 1. These steps improved model compatibility and performance.

VI. FEATURE ENGINEERING AND SELECTION

A. Feature Engineering

To detect fake reviews, key features were engineered to capture behavioral and textual patterns:

1. Review Character Count: Highlights unusually short or long reviews often associated with fake content.
2. Sentiment Polarity: Scores sentiment (-1 to +1) to detect overly positive or negative reviews indicative of manipulation.
3. Text Subjectivity: Quantifies subjectivity (0 to 1), as fake reviews are often excessively subjective.
4. Time Difference Between Reviews: Measures gaps between consecutive reviews by the same user; short intervals suggest suspicious activity.

These features form the foundation for robust machine learning models to identify fake reviews effectively.

B. Feature Selection

To enhance model efficiency, Recursive Feature Elimination (RFE) with a RandomForestClassifier was employed to identify the most influential features:

Methodology:

1. The dataset included 11 features: user interactions, review attributes, text embeddings, and time-based metrics.
2. RFE systematically retained the top 6 features using a RandomForestClassifier with 15 estimators.

Results:

1. The top 6 features were: rating number, review character count, text sentiment, text subjectivity, store encoded, combined_w2v_mean.
2. Features like average rating and Description_w2v_mean were deemed less relevant.

This process ensured that only the most significant features were used, improving model interpretability and predictive power for detecting fake reviews.

VII. FAKE REVIEW DETECTION

For detecting fake reviews Amazon All Beauty Review Dataset is used. Fig.5 shows the block diagram of the fake review detection system.

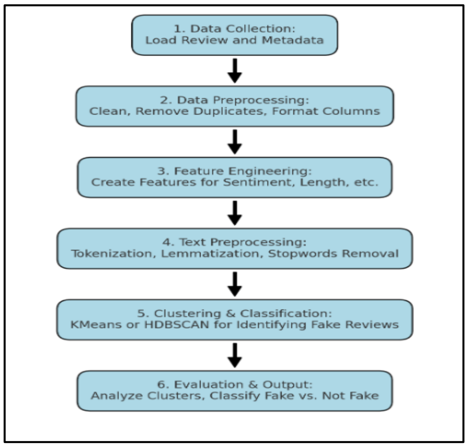
1. Data Collection:
Load Review and Metadata
2. Data Preprocessing:
Clean, Remove Duplicates, Format Columns
3. Feature Engineering:
Create Features for Sentiment, Length, etc.
4. Text Preprocessing:
Tokenization, Lemmatization, Stopwords Removal
5. Clustering & Classification:
KMeans or HDBSCAN for Identifying Fake Reviews
6. Evaluation & Output:
Analyze Clusters, Classify Fake vs. Not Fake

Fig. 5. Fake Review Detection System

A. Datasets

For analysis, Amazon reviews and product meta data dataset is used. All Beauty category is chosen and filtered for the years 2015 to 2023 to make the processing computationally less expensive. The filtered dataset consists of 601,556 total reviews with 559,701 unique users and 106,924 unique products for which reviews are collected. Predictions for fake and genuine reviews were made on the entire dataset using unsupervised learning.

B. Unsupervised Learning - Clustering

The task of grouping similar data points is called clustering. This is a branch of unsupervised learning which aims to collect insights from unlabeled data points. Depending on the type of clustering algorithm several techniques are employed to group data points. In our model we have used K-means clustering which employs Centroid-based clustering and Hierarchical Density-Based Spatial Clustering (HDBSCAN) which employes density-based clustering for the analysis of fake reviews.

K-means algorithm takes the input parameter K from the user and partitions the dataset into K clusters grouping similar objects into a cluster. The number of clusters determines the size of clusters. If K is high it results in smaller clusters while, a low K value results in larger clusters. Centroid based clustering uses the center of each cluster to minimize the sum of the distances between the data points and their corresponding cluster centroids.

HDBSCAN is a fast implementation of DBSCAN and its related algorithms. It calculates the density-based clustering hierarchy which creates clusters from a densely connected data point. This hierarchical structure enables recognition of clusters of various shapes and sizes. Density-based clustering identifies clusters in data by finding the areas of high point density separated by areas of low density. The datapoints in the sparse regions are considered as noise/outliers.

C. Model Implementation

Features are prepared, standardized to ensure scaling and for further processing.

K-means implementation: This system makes use of elbow method by iterating over a range of cluster counts (k=2 to k=20) to calculate the inertia. A plot of inertia vs number of clusters is analyzed to identify the 'elbow point' i.e. the number of cluster (K). Fig. 6 shows the elbow method implementation and the optimal value of K=6 is chosen.

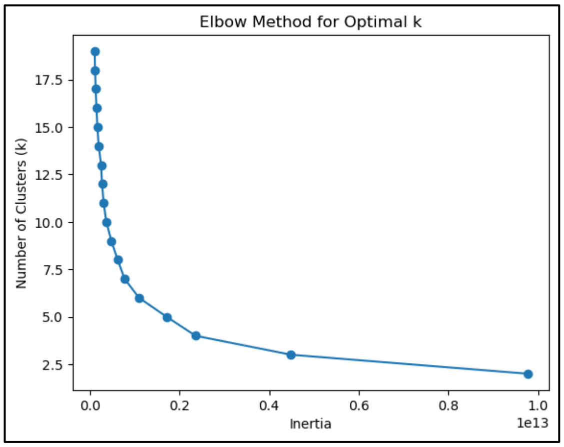

Fig. 6. Elbow Method for Optimal K

K-means clustering is performed with the number of clusters set to 6. Principle Component Analysis (PCA) is used to reduce the dataset to 2-dimensions for visualization as seen in Fig. 7. Thresholds like size threshold (sum cluster sizes x 0.15), sentiment threshold (0.7), subjectivity threshold (0.9) and short review threshold (-0.07) are set to classify the data points. Data points with cluster size < cluster size threshold, text subjectivity score > subjectivity threshold, review character count score < short review threshold and text sentiment score > sentiment threshold are marked as 'Fake' and others are marked as 'Not Fake'. Finally, the model is evaluated using silhouette score. The score is calculated for a subsample of 50000 datapoints to reduce the computation.

HDBSCAN implementation: HDBSCAN is implementd with min_samples set to 4 which defines the minimum number of points required to form a core point. A low value is chosen to ensure detection of smaller, denser clusters. The min_cluster_size set to 9 which defines the minimum size of cluster ensuring that nnoise points or small groups below this threshold are not considered as valid clusters.

The min_cluster_size set to 9 which defines the minimum size of cluster ensuring that noise points or small groups below this threshold are not considered as valid clusters. HDBSCAN clustering algorithm is applied to the preprocessed data, assigning cluster label to each data point. Outliers are defined with the label -1. A threshold of 20 points is used to classify the clusters. Cluster sizes greater than or equal to 20 are labelled as 'Not Fake' and the others are classified as 'Fake' reviews. PCA is used to reduce the high-dimensional data to two dimensions for visualization as seen in Fig. 8. The stability scores of clusters were evaluated using probabilities provided by HDBSCAN which indicate confidence of each point's assignment to its cluster. The average score is calculated to assess the overall clustering robustness.

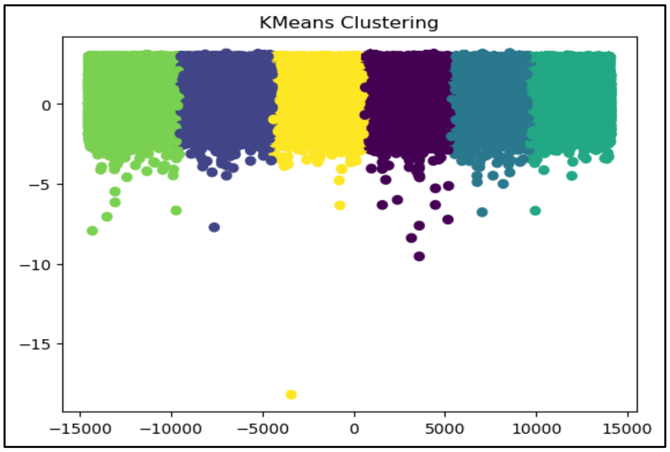

Fig. 7. K – means clustering PCA plot

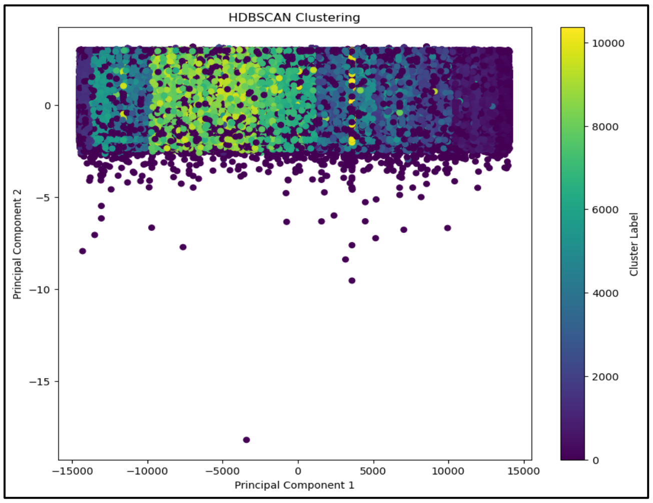

Fig. 8. HDBSCAN clustering

VIII. EXPERIMENTAL RESULTS

Fake review detection system is implemented using unsupervised learning. We used Python for data cleaning, preprocessing, text processing and implementing the authentication algorithms. The performance for K-means algorithm is measured with the silhouette score obtained on a sample of 50000 data points. A silhouette score of 0.56709 is obtained. The proportion of Fake and Genuine reviews obtained are seen in Table. 1.

For HDBSCAN the performance is measure in terms of the average cluster stability score which indicates how well-defines a cluster is. A stability score of 0.6013 is obtained and the classification of the proportion of Fake and Genuine reviews is seen in Table.1 again.

	No. of fake reviews	No. of genuine reviews
K -means	260069	341487
HDBSCAN	233361	368195

Table. 1. Count of fake and genuine classification

IX. CONCLUSION

K-means clustering demonstrated effective partitioning with a silhouette score of 0.56709, while HDBSCAN exhibited robust performance with a cluster stability score of 0.6013. Both methods highlighted the importance of integrating textual features (e.g., sentiment and subjectivity) with behavioral attributes (e.g., review frequency and character count) to detect anomalies effectively.

The system classified a significant proportion of reviews as fake, emphasizing the prevalence of manipulation in online platforms. These findings underline the potential of data-driven approaches in enhancing review authenticity and building trust in e-commerce ecosystems. Future work can expand on this foundation by incorporating supervised learning techniques and contextual embeddings to further refine fake review detection systems.

X. REFERENCES

[1] IEEE Xplore, "Fake Reviews Detection: A Survey," International Conference on Fake Review Detection, IEEE, 2020.

[2] Knowledge and Information Systems Journal, "Fake Review Detection Techniques, Issues, and Future Research Directions," Volume 23, Issue 5, 2024.

[3] Frontiers in Artificial Intelligence, "Graph Learning for Fake Review Detection," Volume 12, 2022.

[4] IEEE Xplore, "Detecting Fake Reviews Using Machine Learning Techniques: A Survey," IEEE International Conference on Machine Learning in Review Detection, 2023.

[5] IEEE Xplore, "Fake Review Detection Using Neural Network," Neural Computation for E-Commerce Applications, IEEE, 2024.

[6] Cornell University (arXiv), "Impact of Sentiment Analysis in Fake Review Detection," Sentiment-Based Fraud Detection Studies, 2023.

[7] Cornell University (arXiv), "Confounds and Overestimations in Fake Review Detection: Experimentally Controlling for Product-Ownership and Data-Origin," Exploratory Studies in Consumer Behavior Analysis, 2021.

[8] This study examines the effects of data-origin and product ownership on fake review detection, highlighting potential overestimations in classification performance.

[9] Cornell University (arXiv), "Fake or Genuine? Contextualised Text Representation for Fake Review Detection," AI Transformative Approaches for Fraud Detection, 2022.

[10] Cornell University (arXiv), "Unmasking Falsehoods in Reviews: An Exploration of NLP Techniques," Natural Language Processing in Consumer Analysis, 2023.