# Project Report

# Sentiment Analysis on Amazon FineFood Reviews

# INTRODUCTION

This project focuses on performing sentiment analysis on a dataset of Amazon Fine Food product reviews. Sentiment analysis is an essential natural language processing (NLP) task that categorizes text into positive, neutral, or negative sentiments. This project applies machine learning techniques to classify customer feedback, providing insights that can help businesses understand consumer perceptions and improve product offerings.

This report details the steps involved in preprocessing the data, the machine learning models used for sentiment classification, the evaluation of those models, and the results we achieved.

## 1. DATASET ACQUISITION

The **Amazon Fine Food Reviews dataset** contains reviews of fine foods from Amazon, spanning from October 1999 to October 2012. It includes over 568,000 reviews, provided by 256,059 users for 74,258 products and has a size of 300.9 MB. It has been taken from A notable feature of the dataset is that it includes reviews from a variety of categories beyond fine foods. There are also 260 users who have written more than 50 reviews. It has been taken from Kaggle.

**The Amazon Fine Food Reviews dataset contains the following columns:**

- **Id:** Unique identifier for each review.
- **ProductId:** Identifier for the product being reviewed.
- **UserId:** Identifier for the user who wrote the review.
- **ProfileName:** Name of the user profile.
- **HelpfulnessNumerator:** Number of users who found the review helpful.
- **HelpfulnessDenominator:** Total number of users who voted on the helpfulness.
- **Score:** Rating given by the user (1 to 5).
- **Time:** Timestamp of the review.
- **Summary:** Short summary of the review

**Reason for choosing:** The dataset was selected due to its richness in text data and the diversity of products reviewed, which makes it suitable for sentiment analysis. Reviews are labeled as positive, neutral, or negative based on the rating provided by users, offering a balanced view of customer feedback.

## 2. DATA HANDLING & PREPROCESSING

**Data Cleaning**

The initial dataset had missing values in the ProfileName and Summary columns. The following steps were applied:

- **Dropped Irrelevant Columns**: The ProfileName column was deemed unnecessary for sentiment analysis and was removed.
- **Handling Missing Values**: Missing values in the Summary column were filled with the placeholder "No Summary".

# 3. DATA TRANSFORMATION

A new column named **“Sentiment”** was derived from the **“Score”** feature for Sentiment Analysis.Based on the review scores:

- Reviews with a score >3 were labeled **positive**.
- Reviews with a score of 3 were labeled **neutral**.
- Reviews with a score <3 were labeled **negative**.

# 4. TEXT PREPROCESSING

**Removing Unnecessary Data**: URLs, HTML tags, special characters, and numbers were stripped from the reviews to clean the text.

**Tokenization and Lemmatization**: The NLTK library was used for tokenization, stopword removal, and lemmatization, transforming the text to its base form for better analysis.

# 5. VISUALIZATIONS

**Several visualizations were created to explore the data and model predictions:**

1. **Word Cloud**: A word cloud was generated to visualize the most frequent terms in the review text. The size of words like "taste," "good," and "love" indicated the dominance of positive reviews. Negative words were less frequent, showing the dataset's initial imbalance.
2. **Distribution of Review Scores**: A bar plot was created to visualize the distribution of review scores from 1 to 5. This chart highlighted the skew toward higher (positive) scores, which necessitated the use of data balancing techniques.
3. **Helpfulness vs. Rating Score:** A scatter plot was used to show the relationship between the helpfulness numerator and the rating score. Positive reviews (4-5 stars) were associated with higher helpfulness scores, while negative reviews (1-2 stars) showed lower helpfulness, indicating less engagement.
4. **Sentiment, Helpfulness, and Rating (Multivariate Visualization ):** A scatter plot with sentiment, rating score, and helpfulness metrics revealed that positive sentiment correlated with higher ratings and greater helpfulness. Negative sentiment reviews had lower ratings and less engagement, while neutral reviews showed mixed results.

# 6. Data Warehousing (SQLite) & Data Processing (ETL)

### Choice of SQLite and Significance

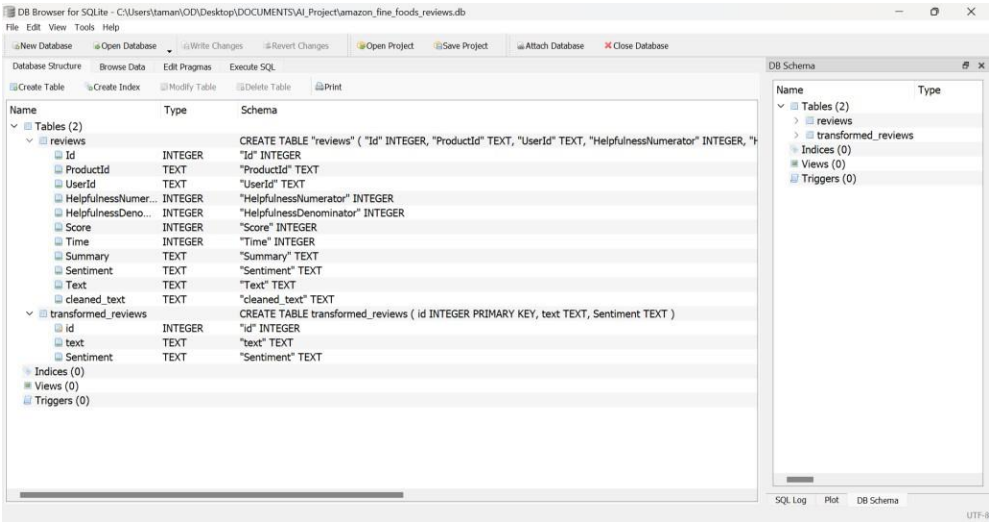
SQLite was chosen for this project due to its ability to handle large datasets efficiently while providing a simple, self-contained database system. With a dataset size of 300MB, SQLite allowed for easy management and querying of the data. Storing the data in SQLite facilitated structured data storage and enabled the use of SQL-based operations for data extraction, transformation, and loading (ETL), which helped streamline the analysis process and ensured easy access to the data for further steps.

## Queries and Results

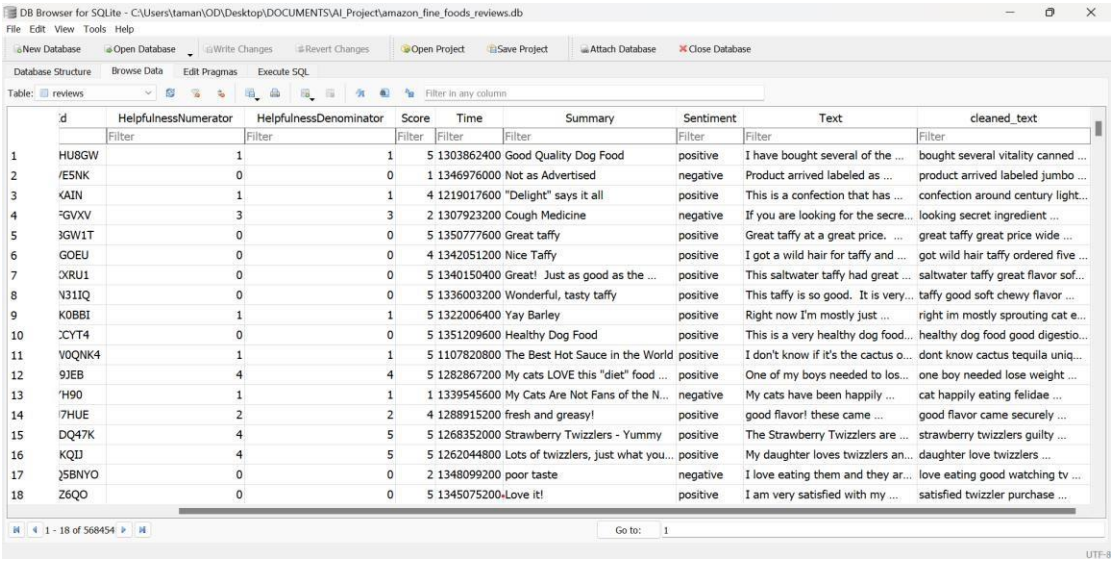
1. **Displaying First Few Rows:** This query ensured the successful insertion of data into the database, confirming the presence of key fields like productId, userId, sentiment, and cleaned text.

2. **Count of Total Records:** This query provided the total number of records in the reviews table, helping assess the completeness of the dataset.
3. **Reviews per Sentiment:** This query summarized the number of reviews classified by sentiment (positive, neutral, negative), providing insights into the sentiment distribution within the dataset.
4. **Simple ETL Workflow:** The ETL process was demonstrated by extracting data from the reviews table, transforming it (e.g., cleaning), and loading it into a new table for better organization and analysis.
5. **Identifying Duplicate Records:** This query identified duplicate records based on review IDs, ensuring the integrity of the data by eliminating redundancies.

**This is the schema of the 2 tables named "reviews" and "transformed_reviews" via ETL queries created in SQLite**




This is the data stored in the database and storing the dataset in SQLite ensures accessibility for performing operations like aggregation, sentiment analysis, and ETL workflows directly within the database.



# 7. HANDLING DATA IMBALANCE

The dataset was highly imbalanced, with most reviews being classified as **positive**. This imbalance can bias the model toward predicting positive sentiments. To mitigate this issue, **Random Undersampling** was used to balance the dataset. Each sentiment class (positive, neutral, negative) was represented equally to ensure the model learned effectively from all categories

# 8. FEATURE EXTRACTION

To convert textual data into numerical form for machine learning models, the **TF-IDF (Term Frequency-Inverse Document Frequency)** technique was used. This approach computes the importance of words in the reviews relative to the entire corpus. The top 3,000 most relevant words were selected as features for model training, reducing dimensionality while keeping the most important terms.

# 9. MODEL BUILDING AND EVALUATION

| Model | Reasons for Selection |
|---|---|
| **Logistic Regression** | 1. **Simplicity**: Easy to implement and interpret, making it a good baseline for sentiment analysis.<br>2. **Linear Relationships**: Works well when sentiment is linearly separable, which often applies in sentiment classification tasks. |
| **Random Forest** | 1. **Captures Complex Patterns**: Handles non-linear relationships and interactions in the data, improving accuracy.<br>2. **Reduces Overfitting**: Combines multiple decision trees to reduce overfitting and generalize better, particularly with imbalanced datasets. |

## Logistic Regression

Logistic Regression is a linear classification model that predicts probabilities for class labels using the logistic function. It converts these probabilities into class predictions based on a set threshold.
In Sentiment Analysis, Logistic Regression uses numerical features derived from text (e.g., word counts, TF-IDF) to classify sentiments. It works well for simple patterns but may struggle with complex or overlapping sentiment classes.

## Random Forest

Random Forest is an ensemble model that combines predictions from multiple decision trees built on random subsets of data and features. This approach captures non-linear patterns and reduces overfitting.

In Sentiment Analysis, Random Forest excels at identifying complex relationships in text features, making it more accurate for classifying nuanced sentiments compared to simpler models like Logistic Regression.

**Performance:**

| Model | Logistic Regression | Random Forest Classifier |
|---|---|---|
| **Overall Accuracy** | 71% | 75% |
| **Overall Precision** | 71% | 75% |
| **Overall Recall** | 71% | 75% |
| **Overall F1-Score** | 71% | 75% |
| **Class-wise Precision** | Negative: 72%, Neutral: 63%, Positive: 77% | Negative: 76%, Neutral: 73%, Positive: 77% |
| **Class-wise Recall** | Negative: 72%, Neutral: 62%, Positive: 78% | Negative: 79%, Neutral: 68%, Positive: 80% |
| **Class-wise F1-Score** | Negative: 72%, Neutral: 63%, Positive: 78% | Negative: 77%, Neutral: 70%, Positive: 78% |

# 10. HYPERPARAMETER TUNING

**Hyperparameter Tuning** is the process of selecting the best set of hyperparameters for a machine learning model to enhance its performance. Hyperparameters are configuration settings set before training, such as learning rate, number of trees in a random forest, or regularization strength.

**Grid Search** is a method that exhaustively tests all possible combinations of a predefined set of hyperparameters to find the optimal configuration while **Random Search** randomly selects hyperparameter combinations from a specified range, offering a faster alternative to grid search by not testing every possible combination.

| Criteria | Logistic Regression | Random Forest |
|---|---|---|
| **Hyperparameter Tuning Method** | GridSearchCV | RandomizedSearchCV |
| **Optimal Parameters** | C = 1.0, penalty = 'l2', solver = 'liblinear' | n_estimators = 100, max_depth = None, min_samples_split = 2 |
| **Accuracy** | 71% | 75.43% |
| **Performance Notes** | - Accuracy and class-wise precision, recall, and F1-scores remained unchanged.<br>- Struggled with accurately classifying neutral sentiments. | Slight improvement in accuracy; model was already near-optimal pre-tuning. |
| Overall Accuracy | 71% | 75% |
| Overall Precision | 71% | 75% |
| **Overall F1 Score** | 71% | 75% |
| **Overall Recall** | 71% | 75% |
| **Class-wise Precision** | Negative: 72%, Neutral: 63%, Positive: 77% | Negative: 76%, Neutral: 73%, Positive: 77% |
| **Class-wise Recall** | Negative: 72%, Neutral: 62%, Positive: 78% | Negative: 79%, Neutral: 68%, Positive: 80% |
| **Class-wise F1-Score** | Negative: 72%, Neutral: 63%, Positive: 78% | Negative: 77%, Neutral: 70%, Positive: 78% |

## 11. EVALUATION METRICS

To assess the performance of the models, several evaluation metrics were used:

- **Accuracy**: Measures the proportion of correct predictions. Random Forest achieved 75%, outperforming Logistic Regression (71%).
- **Precision**: Measures the percentage of correct positive, neutral, or negative sentiment predictions out of all predicted instances. Both models showed strong precision, especially for positive reviews.
- **Recall**: Measures how well the models identified true positives. Random Forest performed best in identifying both positive and negative sentiments.
- **F1-Score**: The harmonic mean of precision and recall, providing a balanced measure of performance. Random Forest achieved the highest F1-Score for positive sentiments (0.78)

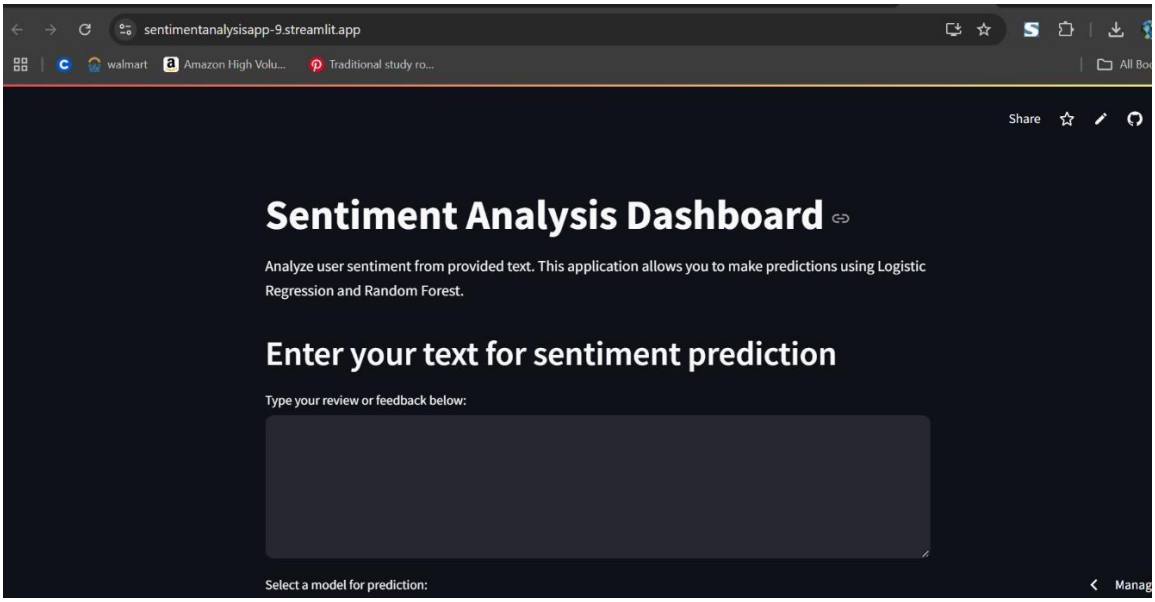
## 12. USER INTERFACE FOR SENTIMENT ANALYSIS

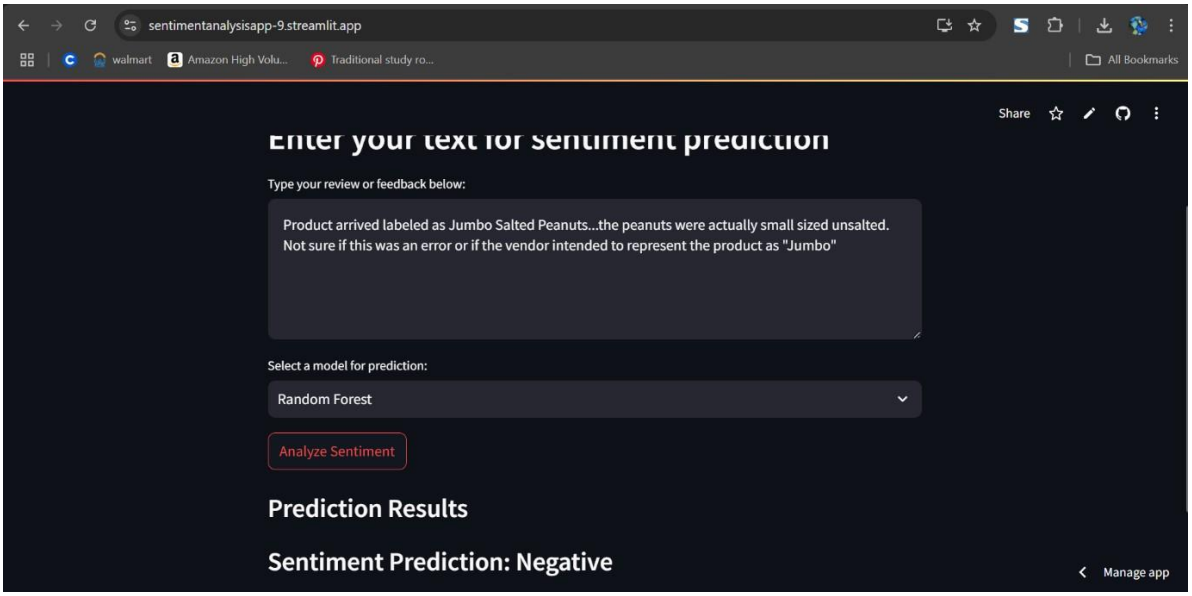
We have created a User Interface (UI) using Streamlit that allows users to input customer reviews and receive sentiment predictions. The interface is simple and user-friendly, where users can easily type or paste their review and get an immediate sentiment classification.

**The sentiment analysis is powered by two models:**

- **Logistic Regression**
- **Random Forest**

**These models work together to accurately predict sentiments as Negative, Neutral, or Positive based on the input. The Interface can be accessed from [https://sentimentanalysisapp-9.streamlit.app/review](https://sentimentanalysisapp-9.streamlit.app/review)**






**NOTE: For detailed Analysis, Insights and Visualizations, kindly refer to jupyter notebook**

## 13. CONCLUSION:

The Random Forest and tuned Random Forest models outperform others, achieving the highest accuracy and balanced performance in sentiment analysis on the Amazon Fine Foods dataset. These models excel in identifying neutral and positive sentiments while maintaining precision and recall. Logistic Regression models, though useful for baseline comparisons, fall short in accuracy and handling neutral sentiment effectively. By using these machine learning classification models, we successfully performed sentiment analysis, providing reliable insights from customer reviews.

## 14. FUTURE WORK:

To further improve the sentiment classification task:

- **Deep Learning Models**: Exploring models such as **Recurrent Neural Networks (RNNs)** or **Transformers** could capture the contextual meaning in the reviews more effectively.
- **Advanced Feature Extraction**: Techniques like **Word2Vec** or **GloVe** embeddings could be used instead of TF-IDF to capture word meaning and relationships more accurately.
- **Data Augmentation**: Augmenting the dataset with additional labeled reviews from other domains could improve model generalization.