Kingdom of Saudi Arabia
Ministry of Education
Umm Al-Qura University
031

المملكة العربية السعودية
وزارة التعليم
جامعة أم القرى
٠٣١

| Name | id |
|---|---|
| Nora Tarek al-suhaibi | 444002730 |
| Raya abdullah alsaab | 444000732 |

# Task3 Report

**Title**: Amazon Fine Food Reviews

The dataset covering reviews from October 1999 to October 2012 contains 568,454 reviews from 256,059 users and 74,258 products, with 260 users having more than 50 reviews; it includes product and user information, star ratings, textual reviews, providing opportunities for analyzing and studying user experience, sentiment analysis, and developing recommendation systems.

## Introduction :

This dataset features reviews of fine foods from Amazon, spanning over a decade from October 1999 to October 2012. It contains around 568,454 reviews from 256,059 unique users and 74,258 different products. The data includes information such as user profiles, product IDs, ratings, and the textual content of the reviews.

This dataset serves as an excellent resource for studying consumer behavior and analyzing product sentiments.

## Objectives:

**•Classifying Reviews as Positive, Negative, or Neutral:**

By analyzing the text of each review, sentiment analysis can classify the overall sentiment of the review, based on the tone and words used. For example, words like "good," "great," or "delight" may indicate positive sentiment, while words like "not as advertised" or "cough medicine" could indicate negative sentiment.

**•Understanding Customer Satisfaction:**

Sentiment analysis can help identify how satisfied or dissatisfied customers are with the products. This can be inferred by the sentiment associated with each review, allowing businesses to gauge customer feedback at scale.

**•Identifying Common Themes or Issues:**

By analyzing the sentiment of reviews along with the specific terms mentioned (e.g., "quality," "price," "taste"), companies can detect recurring themes, problems, or praised features of the products.

**•Correlating Sentiment with Ratings:**

It can be used to check whether the sentiment derived from the review text aligns with the numerical rating provided. For instance, a positive review should correlate with a higher rating, while a negative review should correlate with a lower rating.

**•Improving Products and Services:**

Insights gained from the sentiment analysis can be used to make product improvements, address customer complaints, and enhance overall product quality.

## Importing Libraries:

The following libraries are imported:

pandas: For data manipulation.

seaborn and matplotlib: For visualizing data.

nltk (Natural Language Toolkit): For text processing and sentiment analysis.

Wordcloud: To create and display a cloud consisting of words from texts.

### Configurations:

Matplotlib's figure size is set to [10,5] for consistent plot dimensions.

## Loading the Data:

The dataset is loaded from a CSV file named Reviews.csv. The file is assumed to contain customer reviews, with at least a Text column for review content and a Score column representing the rating.

The encoding is set to UTF-8 to handle special characters.

The dataset shape and initial data are inspected using df.shape and df.head().

## Handling Missing Values:

The code checks for missing values using df.isnull().sum(), identifying any null entries in the dataset.

Missing values are dropped with df.dropna(), and the dataset is checked again to confirm that there are no remaining null values.

### Distribution of Review Scores:

A bar plot is generated to show the distribution of review scores using the Score column:

This plot provides insight into how customer ratings are distributed (e.g., how many 1-star, 2-star, etc., ratings are present).

## Setting Up Sentiment Analysis:

The code uses the VADER (Valence Aware Dictionary and sEntiment Reasoner) sentiment analysis tool from the nltk library.

Necessary components are downloaded using:

SentimentIntensityAnalyzer is used to perform sentiment analysis on text data.

## Example Analysis:

An example review is selected from the dataset (df['Text'][50]), tokenized, and part-of-speech tagged to demonstrate the text processing.

## Performing Sentiment Analysis:

The code calculates the sentiment polarity scores for the example review:

VADER provides four scores: positive, negative, neutral, and compound, which together indicate the sentiment expressed in the text.

## Sentiment Analysis for the Entire Dataset:

Sentiment scores (positive, negative, neutral) are calculated for each review and stored in new columns.

### Calculating Aggregate Sentiment Scores:

The code sums up the positive, negative, and neutral scores across all reviews.

## Classifying Sentiment Based on Scores:

A function sentiment_scores is defined to classify the overall sentiment based on the aggregate scores.

## Results Display:

The overall sentiment of the dataset is printed using the sentiment_scores function.

The aggregate values for positive, negative, and neutral sentiment are displayed.

## Identified Problems and Recommendations:

### Download data:

While downloading the data, it was not downloaded with me, because it was applied by a system other than the normal system.

**Solution:** I used encoding to make sure that the data is read with proper encryption and supports all characters, also "utf-8” was chosen because it can represent most of the letters and symbols used, which is the most appropriate choice and very secure and can handle text data.

### Null values:

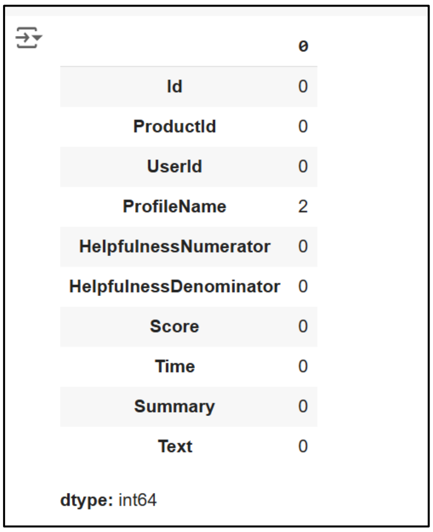
It appeared to us that there are null values in the data I am working on and they should be discarded.

**Solution**: Got rid of null values by deleting them using

the dropna () function.

## output about cood:

## Null values:

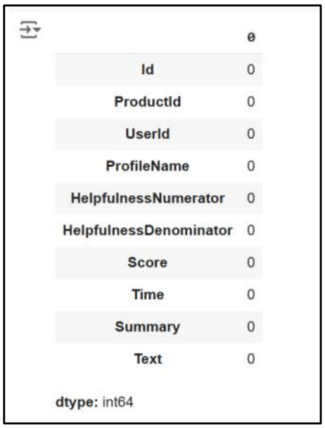


```
                              0
                   Id         0
            ProductId         0
               UserId         0
          ProfileName         2
 HelpfulnessNumerator         0
HelpfulnessDenominator        0
                Score         0
                 Time         0
              Summary         0
                 Text         0

dtype: int64
```

Null values appeared in the data I was working on, and they should have been discarded.

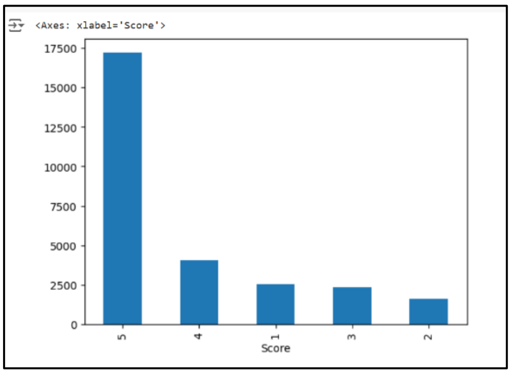
## Null values:



|  | 0 |
|---|---|
| Id | 0 |
| ProductId | 0 |
| UserId | 0 |
| ProfileName | 0 |
| HelpfulnessNumerator | 0 |
| HelpfulnessDenominator | 0 |
| Score | 0 |
| Time | 0 |
| Summary | 0 |
| Text | 0 |

dtype: int64

Null values were eliminated by deleting them using the dropna () function.

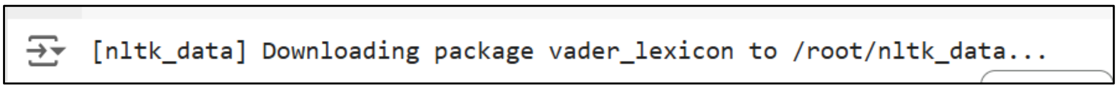


```
<Axes: xlabel='Score'>
```

An illustration of the number of score in the data on which he worked.

It was noted that 5 is the most number inside which there is the largest number of score.

## Vader's message:

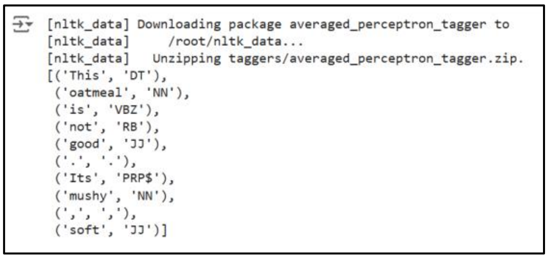


```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
```

I got an encryption message from the vader_lexicon package.

Vader is used to analyze emotions as well as helps determine the type of analyze.

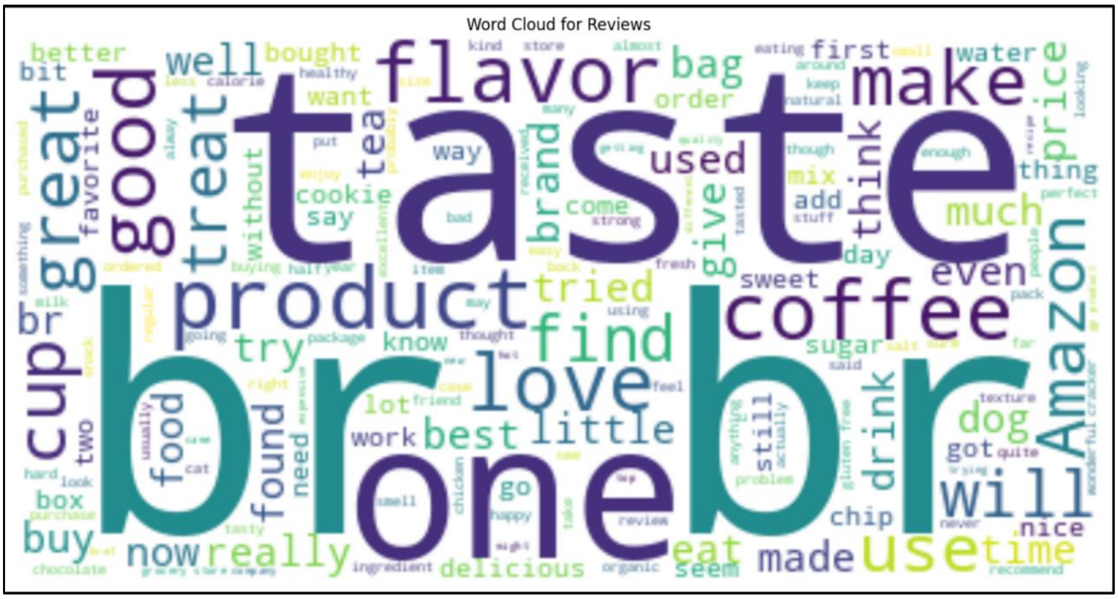
### Determine the type of each word:



```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
[('This', 'DT'),
 ('oatmeal', 'NN'),
 ('is', 'VBZ'),
 ('not', 'RB'),
 ('good', 'JJ'),
 ('.', '.'),
 ('Its', 'PRP$'),
 ('mushy', 'NN'),
 (',', ','),
 ('soft', 'JJ')]
```

Each word and its type in the “ tokens " list were clarified using the pos_tag function.

### Word Cloud for Reviews:

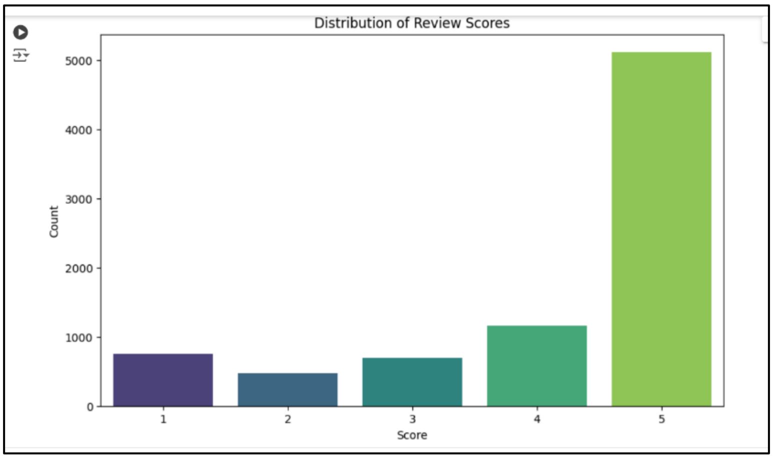


Word Cloud for Reviews

This output contains the most frequent words in the texts from the "text”column.

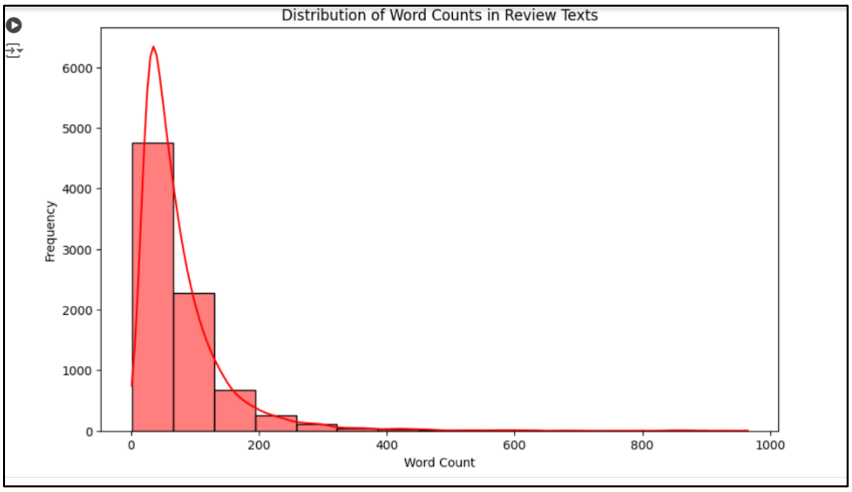
And the larger words are the most frequent in the text.

### Distribution of Review Scores:



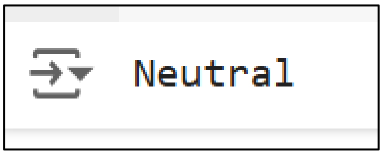
Score 5 is greater than other.

### Distribution of Word Counts in Review Texts:



Distribution of word counts in review text is greater in range 0 to 200.

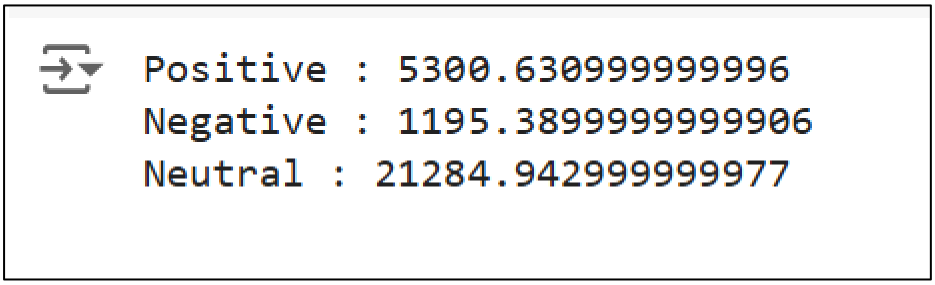
### Total calculation beetween "Positive", "Negative", and "Neutral":



```
Neutral
```

It appeared to me that the product of the values is "Neutral", this means that the values are close and there are no high values.

**Total "Positive", "Negative", and "Neutral":**



```
Positive : 5300.630999999996
Negative : 1195.3899999999906
Neutral : 21284.942999999977
```

And we note that “Neutral " has the highest value, this means that most of the values are Neutral.

## References:

### Link about data:

https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews