## Front End Engineering-II /Artificial

## Intelligence and Machine Learning

Project Report

Semester-IV (Batch-2022)

Customer Sentiment Analysis

A red and white sign

Description automatically generated with low confidence

**Supervised By: Submitted By:**

Faculty Name: Mr. Jatin Arora Nishtha Tandon 2210990621 (G-7)

**Department of Computer Science and Engineering**

## Chitkara University Institute of Engineering & Technology,

## Chitkara University, Punjab

## ****ABSTRACT****

Sentiment analysis is an essential subfield within NLP (Natural Language Processing), dedicated to discerning the underlying sentiment or emotion in digital textual data. Companies today manage vast amounts of text data such as emails, customer support chats, social media comments, and reviews. Sentiment analysis involves examining this text to determine whether the emotional tone of the author is positive, negative, or neutral towards a topic.

Businesses leverage insights from sentiment analysis to enhance customer service and increase brand reputation. They must be quick to respond to potential crises or market trends in today's rapidly changing landscape. Marketers rely on sentiment analysis software to learn what customers feel about the company's brand, products, and services in real time and take immediate actions based on their findings.

In our project, we aim to study the relationship between customer reviews and the product ratings provided. To achieve this, we employed a range of traditional machine learning algorithms, such as Naive Bayes analysis, Support Vector Machines (SVM). We obtained a customer review dataset from Kaggle. Features were extracted from the cleaned reviews using the TF/IDF (Term Frequency-Inverse Document Frequency) method. These features were subsequently used to train a Logistic Regression classifier to predict the sentiment of the reviews. We employed standard metrics such as accuracy, precision, recall, and F1 score to check the classifier performance. Our project illuminates the potent capabilities of machine learning and deep learning techniques in the realm of sentiment analysis. By meticulously analyzing customer reviews and product ratings

**TABLE OF CONENTS**

|  |  |  |
| --- | --- | --- |
| **S.No** | **TITLE** | **Page No.** |
| **1** | Introduction   * 1. **Background**   2. **Objective** | **1-4** |
| **2** | Problem Definition and Requirement  1.1 Problem Statement  1.2 Software Requirements  1.3 Hardware Requirements | **5-9** |
| **3** | Proposed Design/Methodology  1.1Project Directory  1.2Methodology  1.3 Algorithms Used | **10-17** |
| **4** | Results   * 1. Data Preprocessing   2. Text Preprocessing   3. **Model Evaluation**   4. **Result Analysis** | **17-28** |
| **5** | References  1.1Data Set   * 1. Study Material | **29** |

1. **INTRODUCTION**

**1.1 BACKGROUND**

**In today's digital marketplace, understanding customer sentiments has become essential for businesses to thrive amidst fierce competition. With the rise of online platforms like Amazon, where customers freely express their opinions through reviews, extracting meaningful insights from vast volumes of text presents a significant challenge.**

**Traditional methods of market research struggle to keep pace with the sheer volume and diversity of online opinions. Sentiment analysis, also known as opinion mining, addresses this challenge by automating the extraction of sentiments from textual data. However, developing accurate sentiment analysis models faces hurdles due to the complexity and ambiguity of human language.**

**Despite these challenges, sentiment analysis offers immense potential for businesses. By leveraging advanced machine learning algorithms and natural language processing techniques, businesses can derive actionable insights from customer feedback at scale. These insights inform strategic decisions, drive product improvements, and enhance customer satisfaction.**

**In this project, we aim to design and implement an effective sentiment analysis system tailored for Amazon customer reviews. By harnessing state-of-the-art machine learning algorithms, we seek to unlock valuable insights from customer feedback, empowering businesses to make informed decisions and stay competitive in the digital marketplace.**

* 1. **OBJECTIVE**

**The primary aim of this project is to classify customer reviews of various products on Amazon as either positive or negative and to develop a supervised learning model to handle large volumes of these reviews. The dataset comprises customer reviews and ratings sourced from Amazon product reviews.**

**Specific Objectives:**

**1. Feature Extraction:**

* **To Use the TF/IDF (Term Frequency-Inverse Document Frequency) method to extract meaningful features from the reviews.**

**2. Model Development:**

* **To Develop multiple supervised learning models for sentiment classification.**

**3. Performance Comparison:**

* + **To Evaluate and compare the performance of the different models.**

**4. Insight Generation:**

* + **To Analyze the classification results to understand customer sentiments towards different products.**

**By achieving these objectives, we aim to provide insights into customer feedback on Amazon, helping businesses improve their products and services and enhancing customer satisfaction.**

* 1. **INSIGHT**

**In today's digital marketplace, customer reviews play a pivotal role in influencing purchasing decisions. These reviews, often found on e-commerce platforms like Amazon, provide firsthand accounts of customer experiences with products. Reviews typically include a written commentary on the product's performance, quality, and overall satisfaction, often accompanied by a numerical rating. This feedback is invaluable to both potential buyers, who rely on the opinions of others to make informed choices, and to companies, which can use this feedback to improve their products and services. The vast quantity of review data available presents both an opportunity and a challenge: while it offers a rich source of insights, manually analyzing such a large volume of text is impractical.**

**Sentiment analysis, also known as opinion mining, is a technique to automatically determine the emotional tone expressed in textual data. By categorizing the sentiment as positive, negative, or neutral, sentiment analysis helps in understanding the underlying opinions and attitudes conveyed by the text. This process involves several steps, including data preprocessing, feature extraction, and the application of machine learning algorithms to classify the sentiment.**

**In the context of customer reviews, sentiment analysis enables businesses to quickly and efficiently gauge customer satisfaction and identify potential areas for improvement. It can reveal trends and patterns in customer feedback, allowing companies to respond proactively to issues and enhance their products and services.**

* 1. **OVERVIEW**

**In our project, machine learning serves as the cornerstone for constructing efficient sentiment analysis models customized for Amazon's customer reviews. By leveraging various machine learning algorithms, we aim to classify sentiments accurately. Model training on labeled data allows us to capture intricate patterns in customer feedback. Following this, meticulous model evaluation ensures reliability and effectiveness. Through the analysis of classification outcomes, we extract actionable insights into customer sentiments, facilitating informed decision-making for businesses operating in the digital realm.**

**1.4 SIGNIFICANCE**

**This project holds significance for several reasons:**

**1. Enhanced Customer Insights: Accurately classifying customer reviews helps businesses gain insights into customer perceptions and satisfaction levels, facilitating improvements in product quality and customer experience.**

**2. Informed Product Development: Analysis of customer sentiments identifies product strengths and weaknesses, guiding product development strategies and ensuring alignment with customer preferences.**

**3. Competitive Advantage: Effective analysis and response to customer feedback give businesses a competitive edge by allowing them to adapt offerings and maintain customer satisfaction and loyalty.**

**4. Risk Mitigation: Early detection of negative sentiments helps mitigate potential issues, safeguarding brand reputation and minimizing financial losses.**

1. **PROBLEM DEFINITIONAND REQUIREMENT**

**2.1. PROBLEM STATEMENT:**

In today's highly competitive business landscape, understanding and responding to customer sentiments are paramount for sustainable success. However, with the vast amount of unstructured data generated by customers across various channels such as social media, reviews, and surveys, extracting actionable insights poses a significant challenge for businesses. The problem at hand is to develop an effective customer sentiment analysis system that can automatically analyze and classify customer sentiments into positive, negative, or neutral categories.

**2.2**. **SOFTWARE REQUIREMENT:**

For a sentiment analysis project, you'll need several software tools and libraries to preprocess text data, build and train sentiment analysis models, and evaluate their performance. Here's a refined list of essential software requirements:

**Programming Language:**

**PYTHON**: Python provides a versatile and easy-to-use environment for natural language processing (NLP) tasks and machine learning model development.

**Integrated Development Environments (IDE):**

These popular IDEs offer robust support for Python development, providing features such as code autocompletion, debugging, and visualization capabilities.

**Python Libraries:**

* 1. **PANDAS:** Pandas offers powerful tools for data manipulation and analysis, making it ideal for handling structured text data like social media posts or customer reviews. Its DataFrame structure simplifies tasks such as data loading, cleaning, and preprocessing.
  2. **NUMPY**: NumPy is essential for numerical computing in Python, providing support for multidimensional arrays and mathematical functions. It complements Pandas for efficient data processing and manipulation.
  3. **SEABORN**: Seaborn is a statistical data visualization library that works seamlessly with Pandas DataFrames. It simplifies the creation of informative and visually appealing plots to visualize relationships within text data.
  4. **MATPLOTLIB**: Matplotlib is a versatile plotting library that enables the creation of static, interactive, and animated visualizations. Pyplot, a submodule of Matplotlib, offers a MATLAB-like interface for generating customized plots, which is useful for visualizing sentiment analysis results.
  5. **NLTK (NATURAL LANGUAGE TOOLKIT):** NLTK is a comprehensive library for NLP tasks in Python, offering modules for tokenization, stemming, lemmatization, part-of-speech tagging, and named entity recognition. NLTK is indispensable for preprocessing text data and extracting relevant linguistic features for sentiment analysis.
  6. **IMBALANCED-LEARN (IMBLEARN):** Used for handling imbalanced datasets, a common challenge in fraud detection.
  7. **SCIKIT-LEARN (SKLEARN):** Utilized for implementing machine learning algorithms, model evaluation, and preprocessing.

**METHODS:**

1. **Support Vector Machines (SVM):**

Strengths: Effective in high-dimensional spaces, making them suitable for text classification tasks with many features. Versatile kernel functions allow handling non-linear decision boundaries. Robust against overfitting, especially in high-dimensional spaces.

Use Case: SVMs can be effective for sentiment analysis when dealing with text data with a large number of features (e.g., Bag-of-Words or TF-IDF representations).

1. **Logistic Regression:**

Strengths: Simple and interpretable model, making it easy to understand the relationship between features and sentiment.

* 1. Performs well when the decision boundary is linear or close to linear.
  2. Less prone to overfitting, especially with regularization.

Use Case: Logistic Regression is suitable for sentiment analysis tasks where the relationship between features (words or n-grams) and sentiment is approximately linear.

1. **Naive Bayes:**

Strengths: - Simple and computationally efficient algorithm, making it suitable for large datasets.

* 1. Performs well with high-dimensional sparse data, such as text data represented using TF-IDF.
  2. Robust against irrelevant features due to conditional independence assumption.

Use Case: Naive Bayes is commonly used for sentiment analysis, especially in scenarios with limited computational resources or when quick prototyping is required.

**2.3. HARDWARE REQUIREMENT:**

**The hardware requirements for running a Linear Support Vector Machine (SVM) algorithm are generally modest, especially for smaller datasets. Here's a breakdown of the typical hardware requirements:**

**1. Processor (CPU):**

* **A modern multi-core CPU is sufficient for running Linear SVM algorithms.**
* **While SVM training can be computationally intensive for large datasets, most CPUs available today, even in mid-range laptops and desktops, should be able to handle it effectively.**

**2. Memory (RAM):**

* + **Adequate RAM is essential, especially when dealing with large datasets or when training complex models.**
  + **While smaller datasets may require only a few gigabytes of RAM, larger datasets or more complex models may require 8GB or more.**

**3. Storage:**

* + **Linear SVM algorithms do not typically require a large amount of disk space for storing the model parameters and datasets.**
  + **However, having sufficient storage space to store datasets, intermediate results, and trained models is necessary.**

**2.4. DATASET:**

The Amazon Fine Food Reviews dataset is a popular dataset widely used in sentiment analysis and natural language processing research. It consists of reviews of fine foods from Amazon, spanning over a period of more than 10 years. Here are some key points about the dataset:

**Source**: The dataset is available on Kaggle and has been compiled from Amazon's product reviews section.

**Contents**: The dataset contains various attributes including:

**Id**: Unique identifier for each review.

**ProductId**: Unique identifier for the product being reviewed.

**UserId**: Unique identifier for the user who posted the review.

**ProfileName**: Name of the user who posted the review.

**HelpfulnessNumerator**: Number of users who found the review helpful.

**HelpfulnessDenominator**: Number of users who indicated whether they found the review helpful or not.

**Score**: Rating given by the user (ranging from 1 to 5).

**Time**: Timestamp of the review.

**Summary**: Brief summary of the review.

**Text**: The full text of the review.

**Size**: The dataset contains approximately 568,454 reviews, making it a substantial resource for sentiment analysis tasks.

**Variety of Reviews**: The reviews cover a wide range of fine food products available on Amazon, including snacks, beverages, condiments, and more.

* 1. **PROPOSED DESIGN / METHODOLOGY**

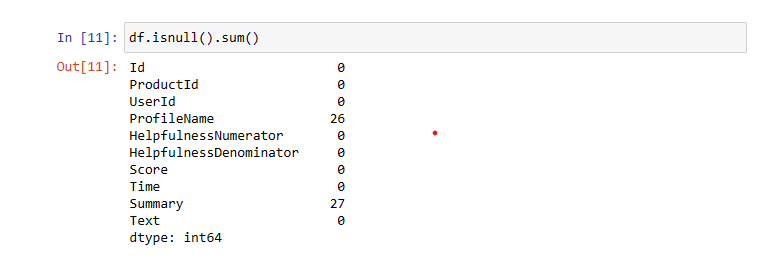
**3.2 METHODOLOGY:**

1. **Data Collection**: We used dataset of Amazon fine food reviews from kaggle. This dataset includes review text, ratings, and sometimes additional information like product ID, user ID, helpfulness votes, etc.

Number of instances: 568454

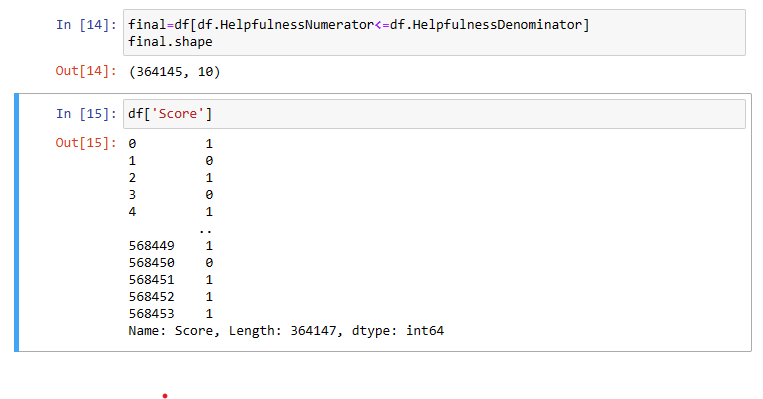
Number of features: 10

1. **Data Preprocessing**:

**Cleaning**: Removed duplicates and handle missing values, in our dataset there were 26 null ProfileNames, 27 null Summary.

After cleaning there were 393914 instances left.

Also in the context of Amazon fine food reviews, the Helpfulness Numerator typically represents the number of people who found a review helpful, while the Helpfulness Denominator represents the total number of people who voted whether the review was helpful or not. It's indeed not practically possible for the Helpfulness Numerator to be greater than the Helpfulness Denominator.

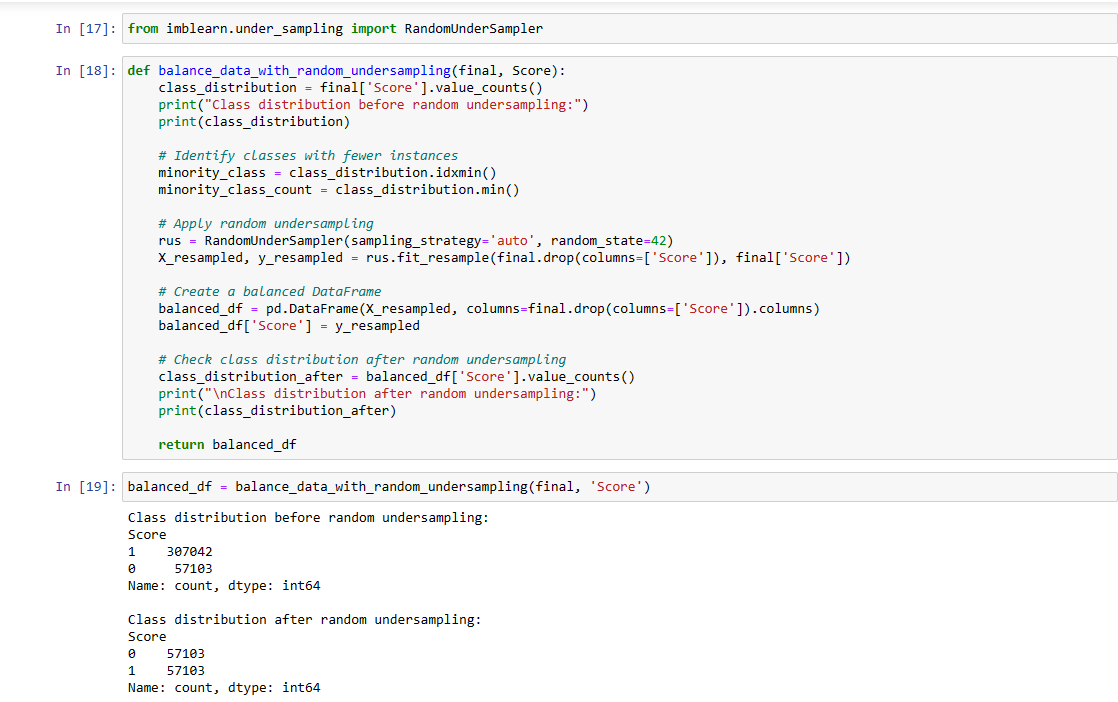
So we removed the data in which contains Helpfulness Numerator greater than HelpfullnessDenominator.

* **Building target values:** In the dataset we were provided with ratings ranging from 1-5, we only want to get the global sentiment of the recommendations (positive or negative), we will purposefully ignore all Scores equal to 3 so we considered the ratings as:

Above 3< : Positive (1)  
Below 3> : Negative (0)

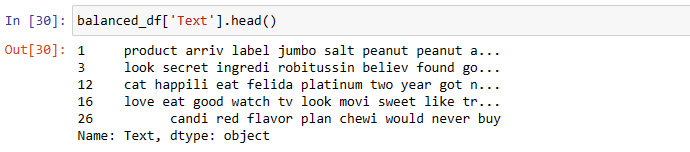
On plotting the Score column we could see there was a large amount of imbalanced which could lead to bias towards the majority class(5 rating). For overcoming this issue we performed RandomUnderSampling which balanced the data.

**Random UnderSampling:** This involves randomly deleting events from the majority class.

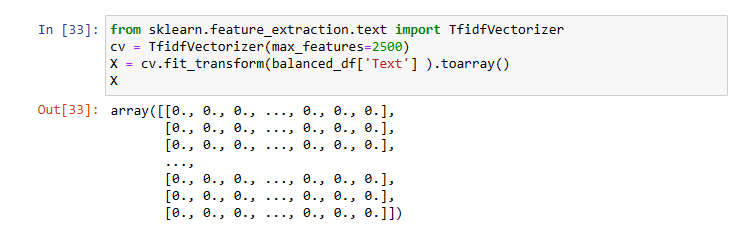


Now we have 57103 instances for each Positive and Negative Scores

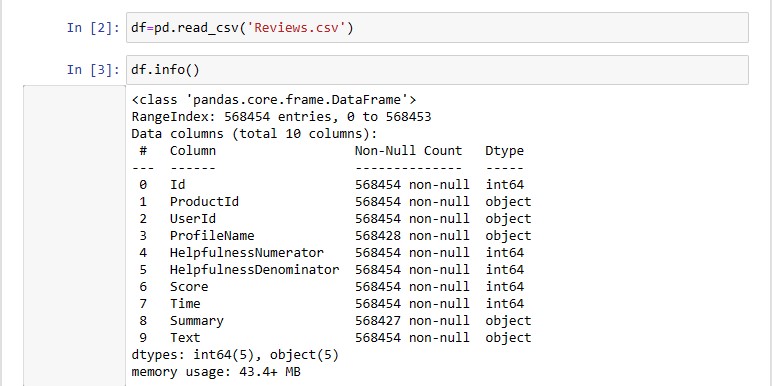
* + **Text Preprocessing**: Eliminated common words that don’t carry much meaning by removing punctuations, numbers, html tags etc. This removed the noise and standardise the data.

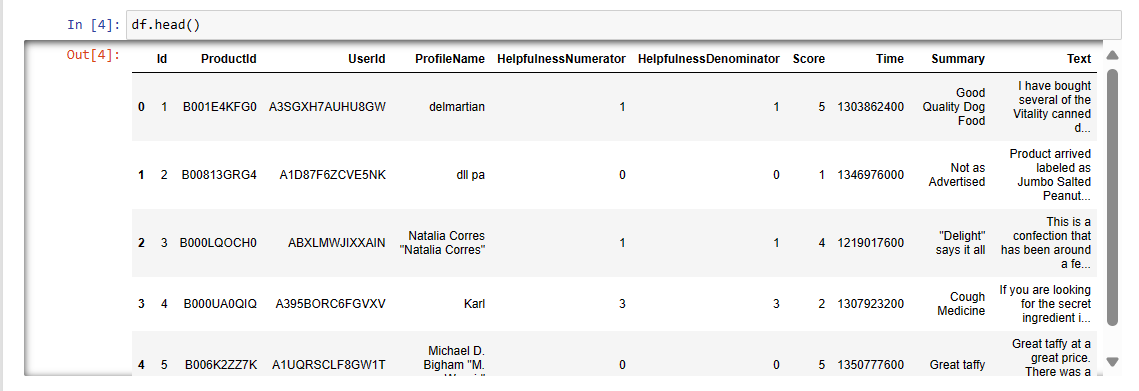


**Lemmatization**: Converted words to their base or root form.

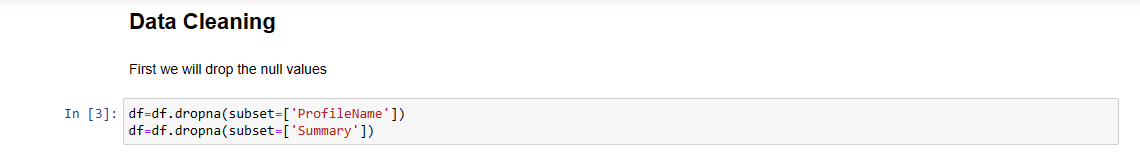
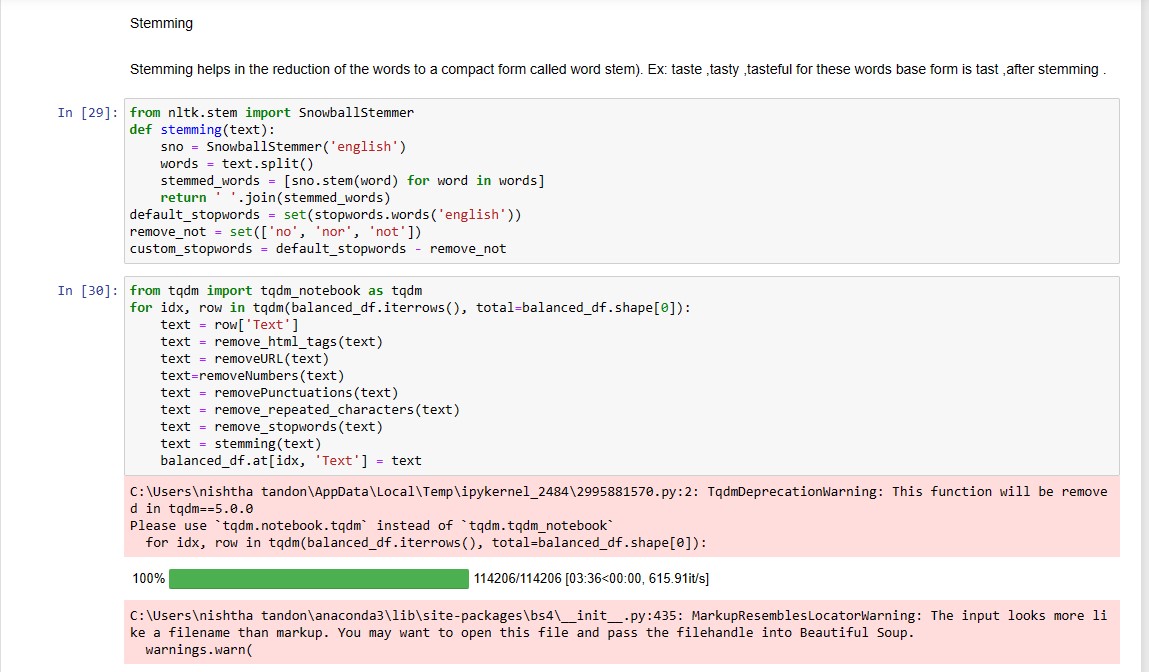
1. **Exploratory Data Analysis (EDA)**: Analyzed the data to understand patterns, trends, and distributions. This may involve visualizing the number of reviews over time, distribution of ratings, etc.
2. **Feature Engineering**: Transformed text data into a format that machine learning algorithms can work with. This often involves creating a bag-of-words model or using TF-IDF (Term Frequency-Inverse Document Frequency) to represent the text.
3. **Model Selection**: Used algorithms like Naïve Bayes, Logistic Regression, or Support Vector Machines for classification. We are performing Supervised learning.
4. **Model Training**: Trained the selected models on the preprocessed dataset.
5. **Evaluation**: Assessed the model’s performance using metrics like accuracy, precision, recall, and F1-score.

Below are the screenshots of the Process:

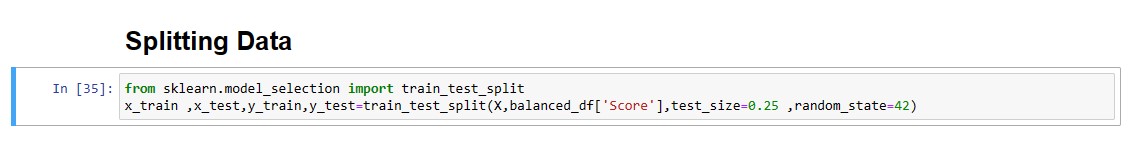
**Import Required Libraries**: Load necessary libraries for data processing, model building, and evaluation.

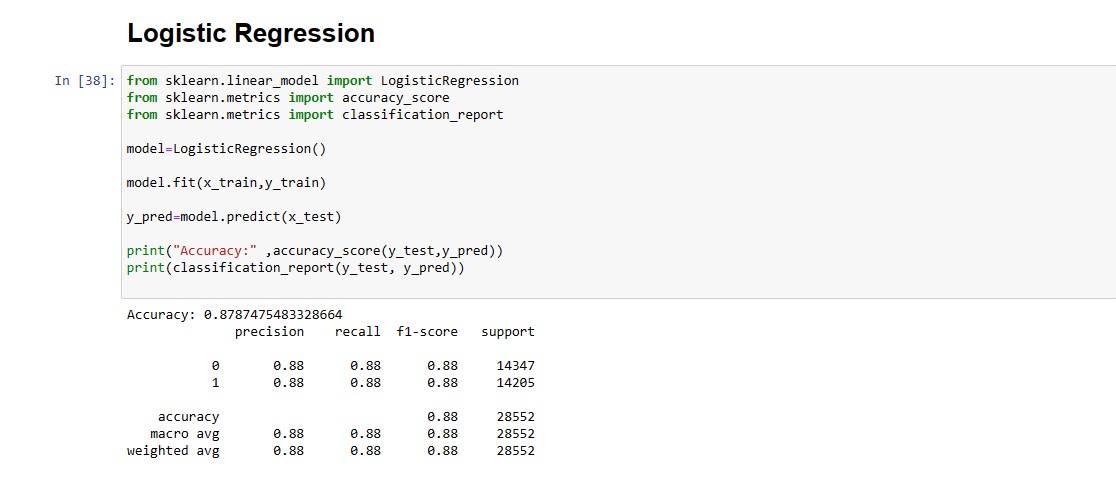


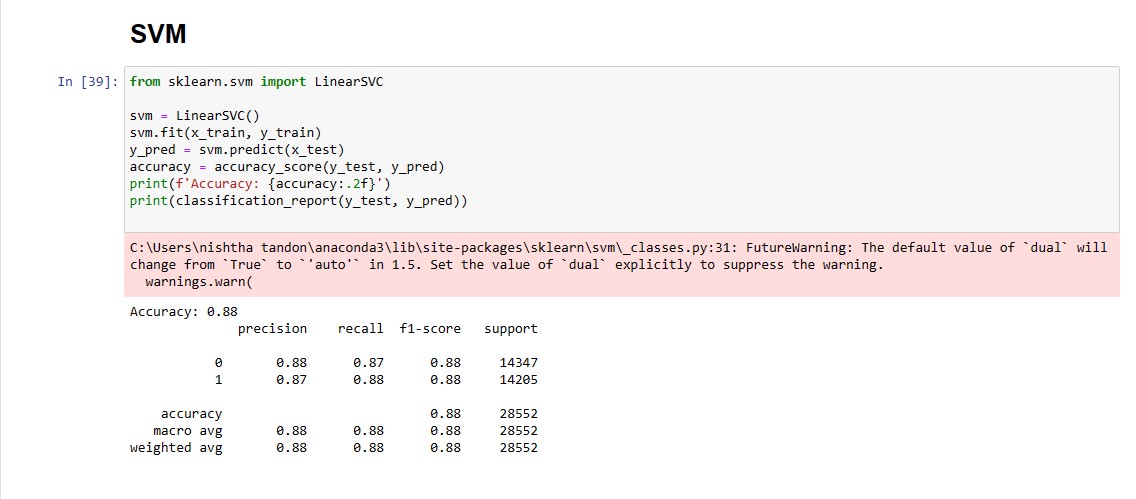
**Load the Dataset**: Load the dataset into a DataFrame.

**Data Preprocessing**: Clean the data by dropping null values and duplicates..

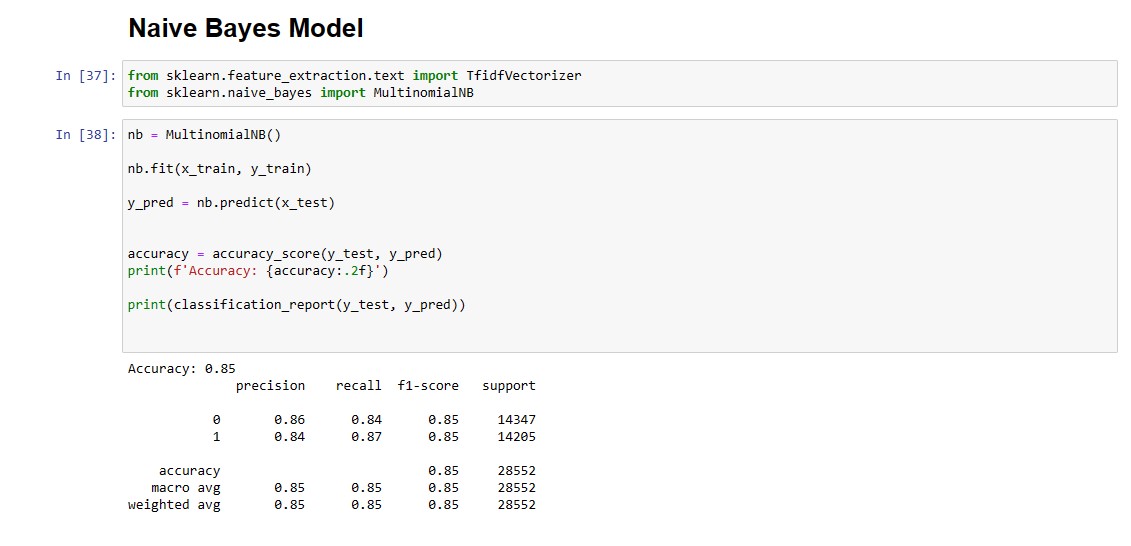
**Text Preprocessing:** Cleaning text to remove noise and standardize the data.

**Feature Extraction**: Convert text data to numerical features using techniques like TF-IDF. 

**Split the Dataset**: Divide data into training and testing sets.

**(I)Logistic Regression**

**(II)SVM**



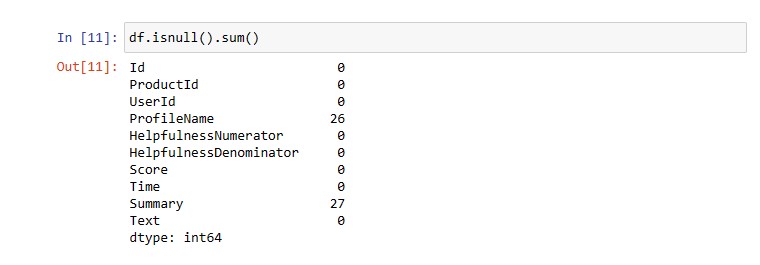
**(III) Naïve Bayes Model**

**Train the Model**: Train a machine learning model on the training data.

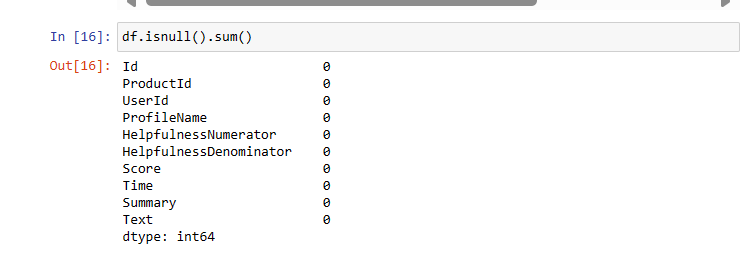
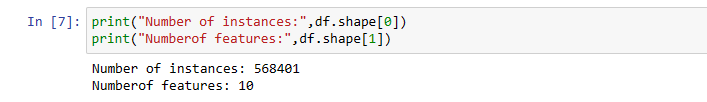
**Evaluate the Model**: Evaluate the model's performance on the test data.

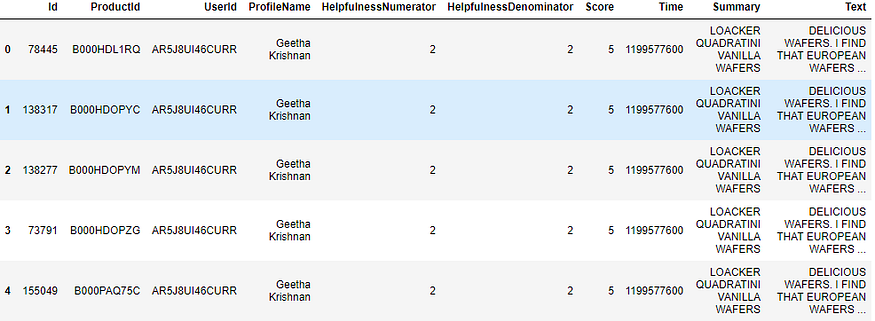
1. **RESULTS**

The entire dataset has 568454 reviews from which 57 were dropped as they consisted of some null values in the data :



Before Deletion

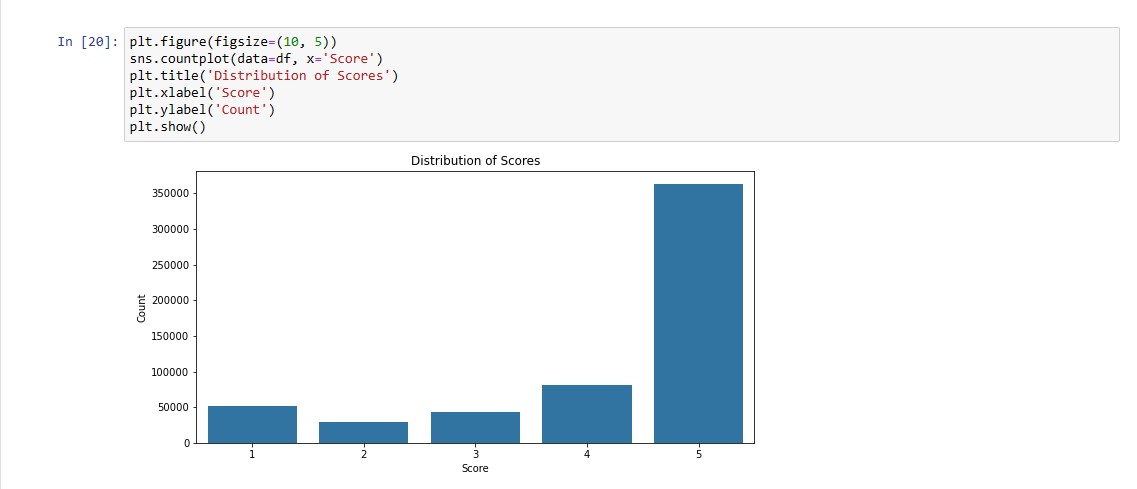
After Deletion

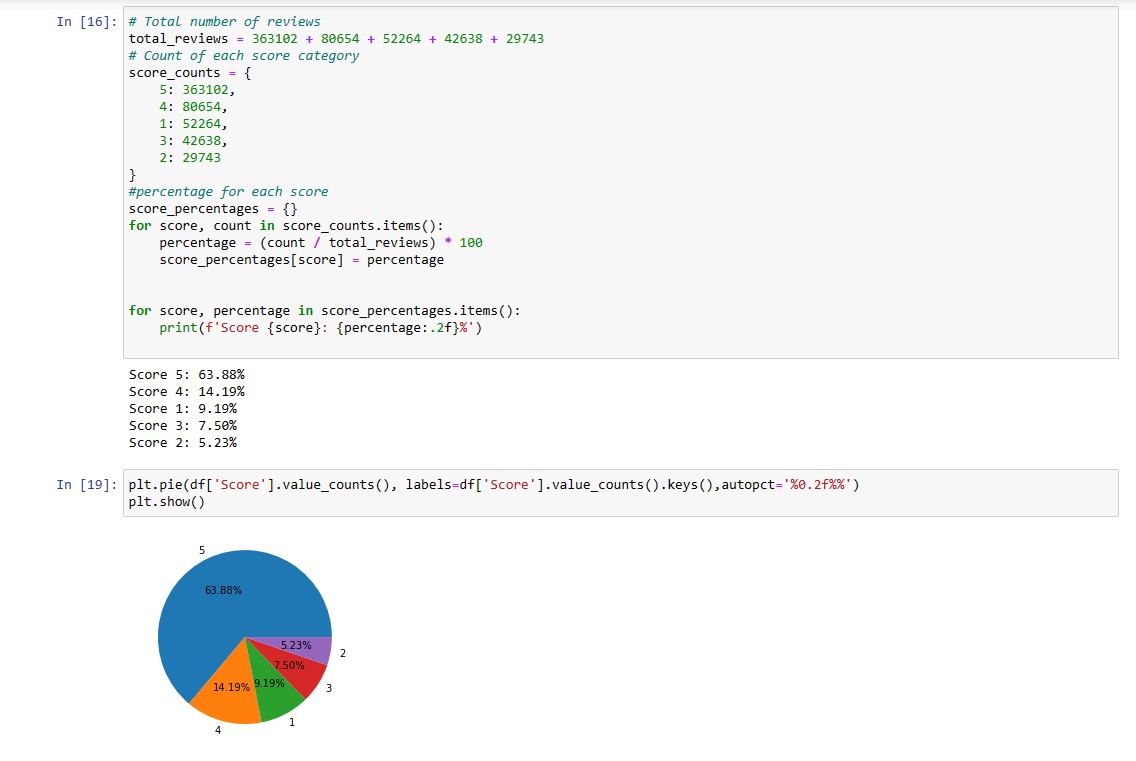
There were multiple colums which consisted of duplicate data by the same user , 

As can be seen above the same user has multiple reviews of the with the same values for Helpfulness Numerator, HelpfullnessDenominator, Score, Time, Summary and Text and on doing analysis were found.

It is necessary to remove duplicates in order to get unbiased results for the analysis of the data. So we removed all the duplicates and were left with 393914 rows.

We are working on Score and Text column to train our model. We firstly worked on the Score column which was used to obtain the target values i.e. 1 (positive) or 0(negative).

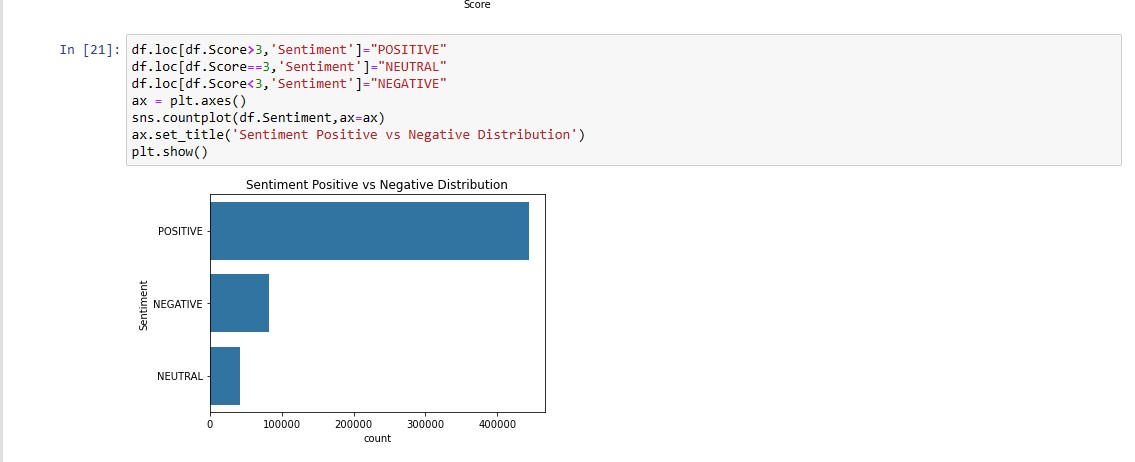
Distribution of the Scores:**

From the Above graph we can see the majority of the score is 5 stars.

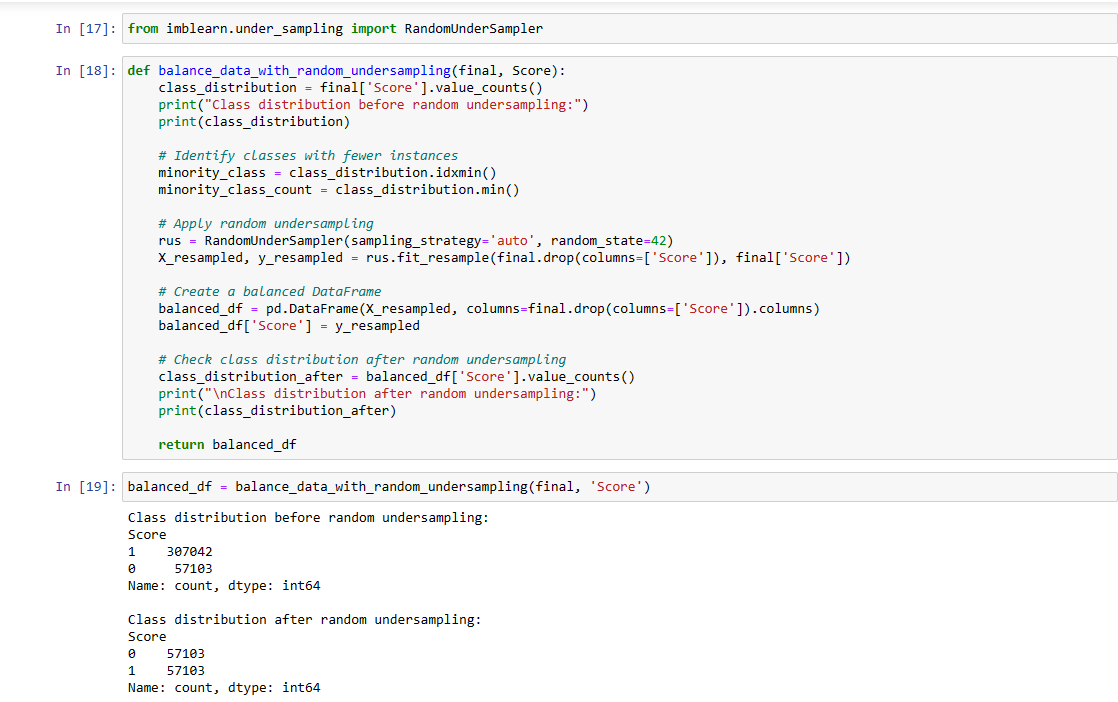
63.88% of data has 5 stars rating making the dataset highly imbalanced. Imbalanced data could lead to biased results towards one class. In an imbalanced data , one class significantly outnumbers the others . This may lead to high accuracy but poor performance in recognizing the minority class.

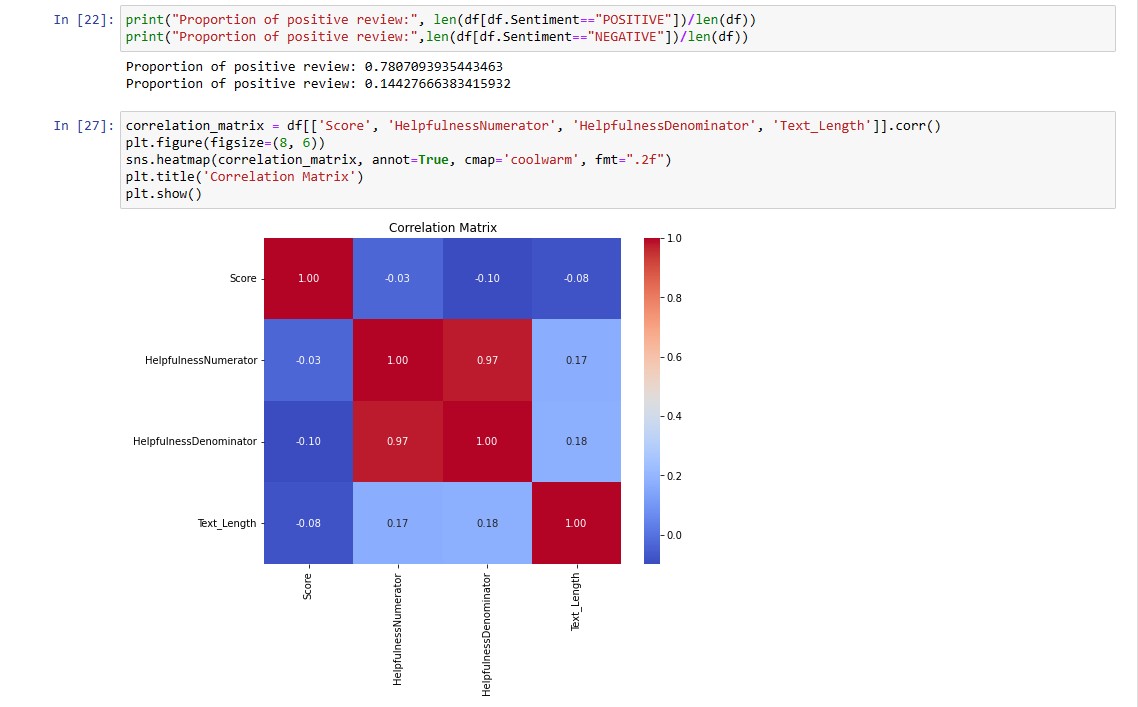
We consider that the Score 4-5 are positive, 3 is neutral, 1-2 are negative.

Here is the distribution:



As stated in the Methadology we convert the ratings in from of 0 and 1. With the use of RandomUnderSampler, we remove the imbalance by removing the rows from majority class. It is a very common used technique.



Now we have 57103 instances for each Positive and Negative Scores

Further, With the use of EDA we analyse patterns present in out dataset.

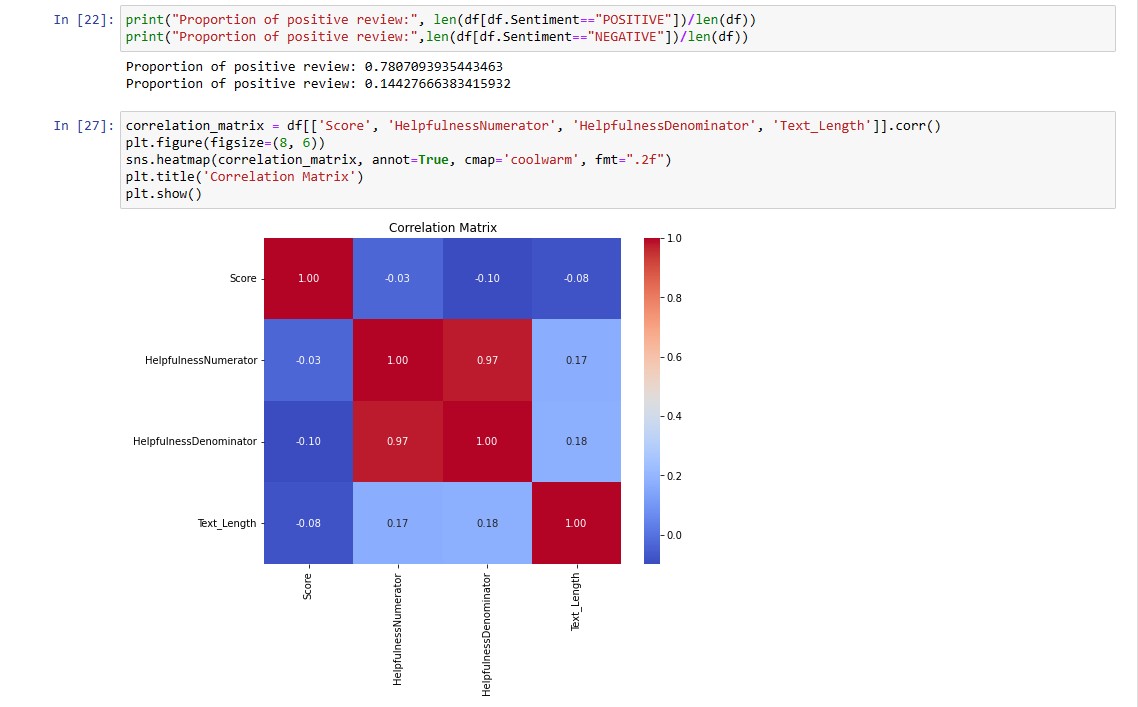
In the heatmap shown above, the diagonal from the top left to bottom right shows 1s, which is normal because it represents the correlation of each variable with itself, which is always perfect (1.0). The other cells show how each of the variables correlates with one another.

Figure 1

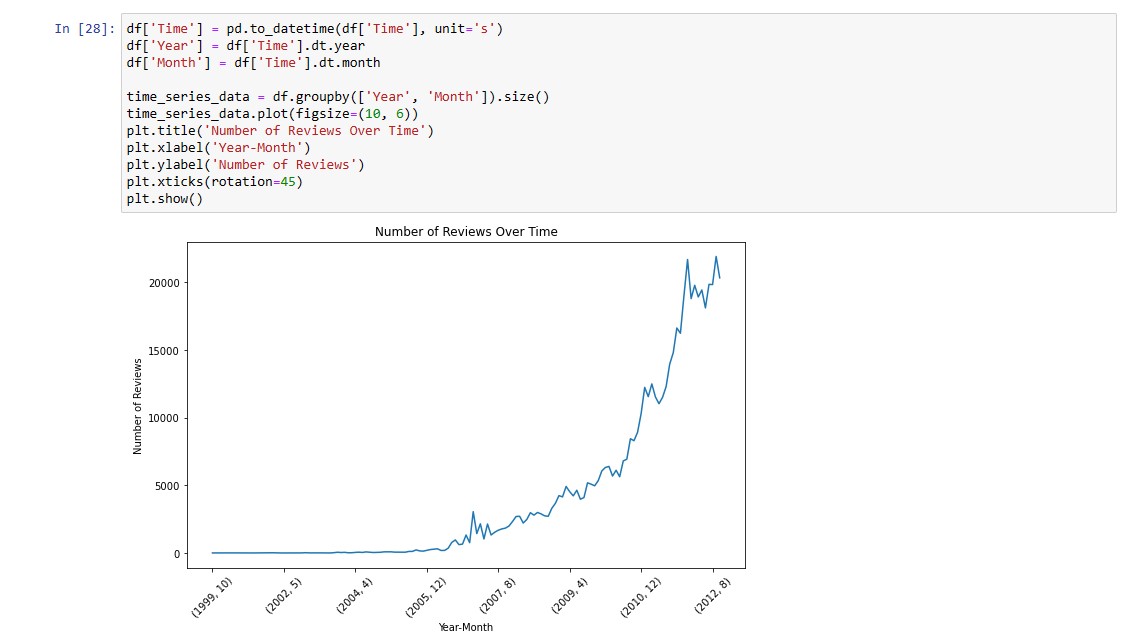
This graph below shows the number of reviews that were taken over the time. With this we can observe how the virtual reviews have grown over the time.

Figure 2

In machine learning features like text length, summary length, and word count are often used because they can provide valuable insights into the data and can be predictive of certain outcomes.

**Text Length**: The length of the text can be indicative of the detail and depth of the content. For example, longer reviews might provide more comprehensive information, while shorter texts might be less informative.

**Summary Length**: The length of a summary can reflect the complexity or simplicity of the original text. It can also indicate how much key information is captured in the summary.

**Word Count**: The number of words in a text can show verbosity or conciseness. It can also affect the readability and understandability of the text.

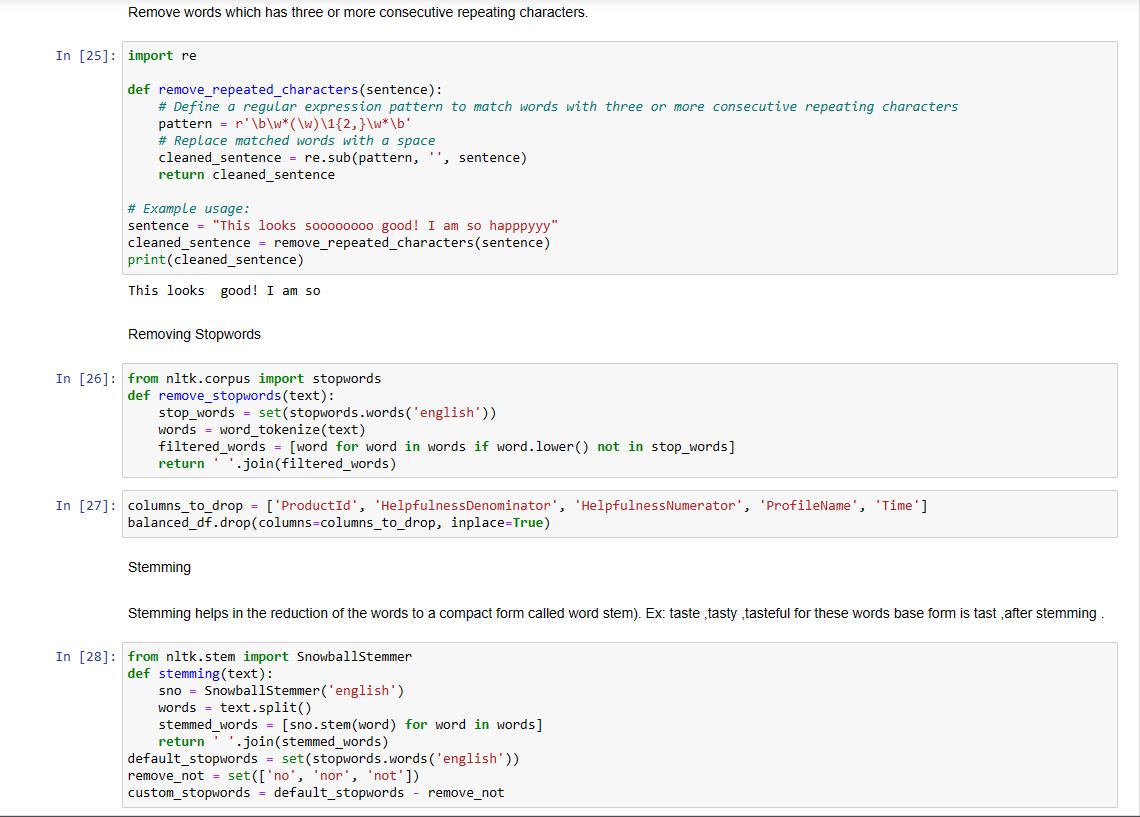
For instance, in product reviews, a longer text length might correlate with a more positive or negative sentiment, depending on the c Noise removal from review text is a critical preprocessing step in natural language processing (NLP) and machine learning because it helps in achieving cleaner and more meaningful data for analysis. Here’s why it’s important:

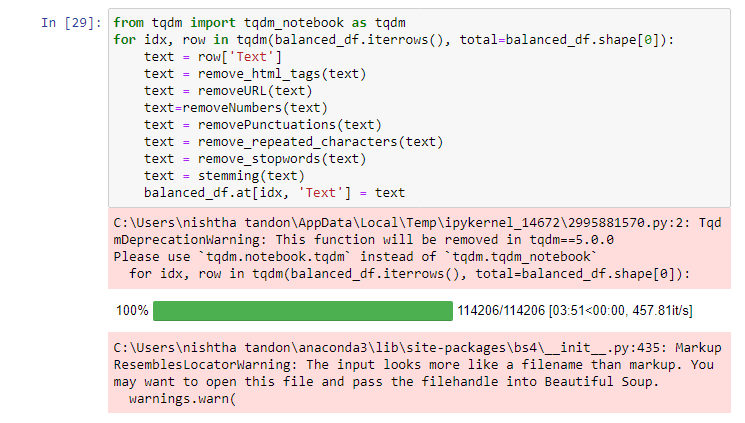
[**Improves Model Accuracy**: By removing irrelevant characters, digits, and pieces of text, the model can focus on the important features that contribute to the prediction](https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machine-learning.html)[1](https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machine-learning.html).

**Consistency**: Noise can cause inconsistency in the text data, leading to unreliable results. [Removing noise helps in maintaining consistency across the dataset](https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machine-learning.html)[1](https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machine-learning.html).

**Reduces Complexity**: Unnecessary noise increases the complexity of the data, making it harder for the model to learn.

So to do this we have performed several operations such as removal of punctuations, html tags to reduce the noise hence cleaning our text. Here are some snapshots:



Stemming and stopwords removal functions are performed in the above image.ta is !

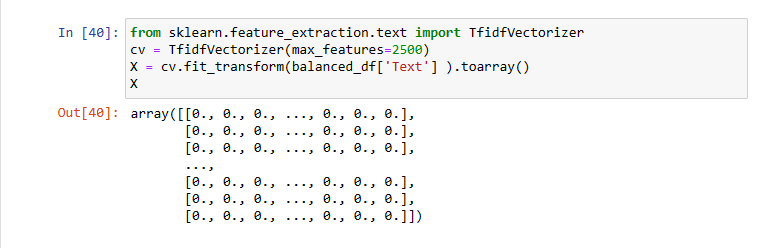
We apply these functions on the text column and now our data is Noise free.

Assigning numerical values to words during sentiment analysis is a fundamental technique in natural language processing (NLP) that allows machine learning models to process and analyze text data. Here’s why it’s important:

[**Quantification of Sentiments**: Numerical values enable the quantification of sentiments, making it possible to measure the intensity and polarity (positive, negative, or neutral) of emotions expressed in the text](https://nealcaren.org/lessons/wordlists/)[1](https://nealcaren.org/lessons/wordlists/).

**Model Input**: Machine learning models require numerical input to perform computations. [By converting words to numbers, we can feed textual data into algorithms for training and prediction purposes](https://nealcaren.org/lessons/wordlists/)[2](https://builtin.com/machine-learning/sentiment-analysis).

**Vector Representation**: Words are often represented as vectors in a high-dimensional space, where each dimension corresponds to a specific feature of the word, such as its context or sentiment score.

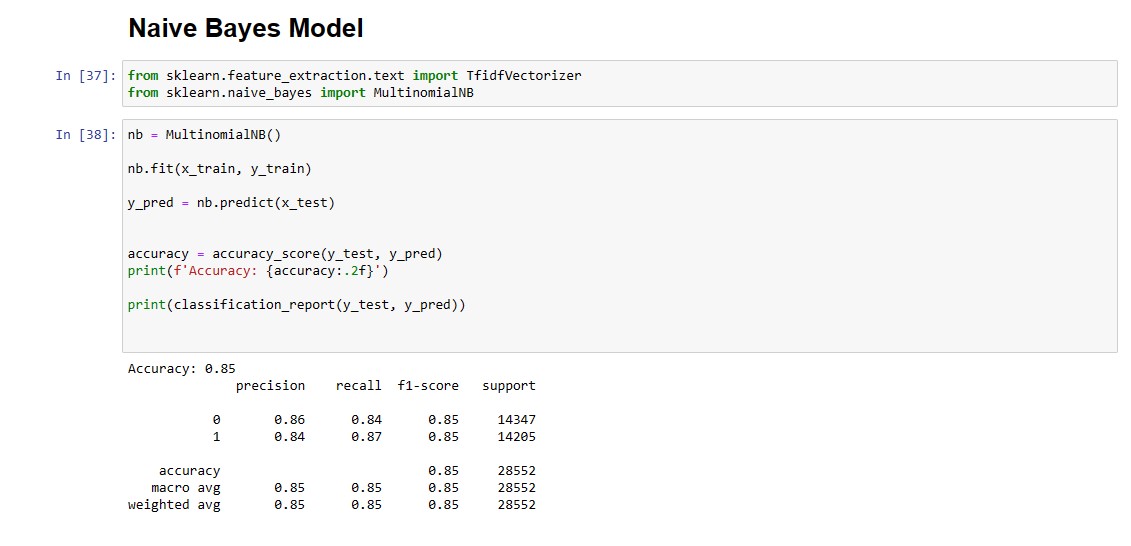


This is the representation of the text once we vectorize it.

We are done with vectorization now we can split our dataset and apply models like SVM, Logistic regression and Naïve Bayes algorithm to it. To test the model we will also use metrices to evaluate the accuracy.

Moreover after that we will compare the results.

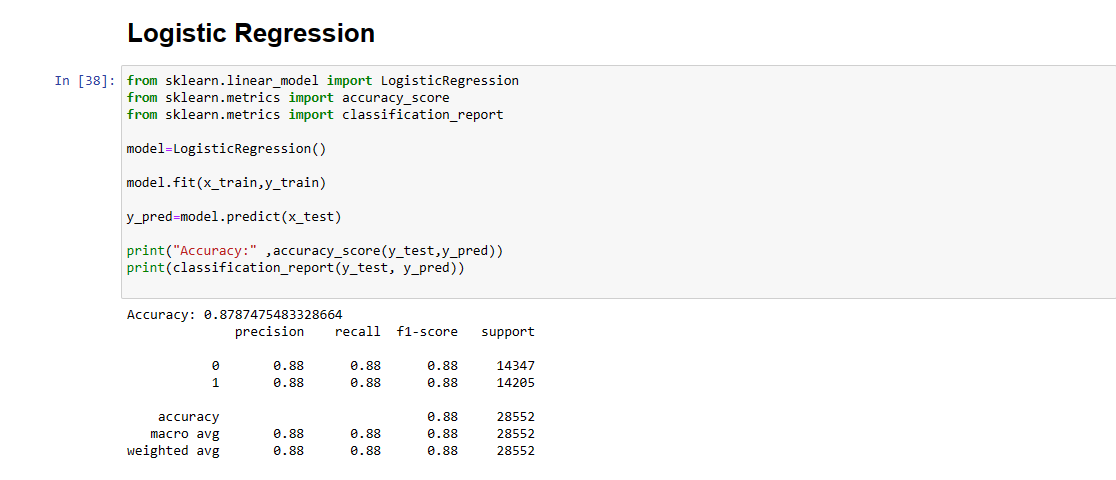
* 1. Naïve Bayes Model



**Naive Bayes**:

**Principle**: Based on Bayes’ theorem, it calculates the probability of each class (like positive or negative sentiment) given a text, and classifies the text based on the highest probability.

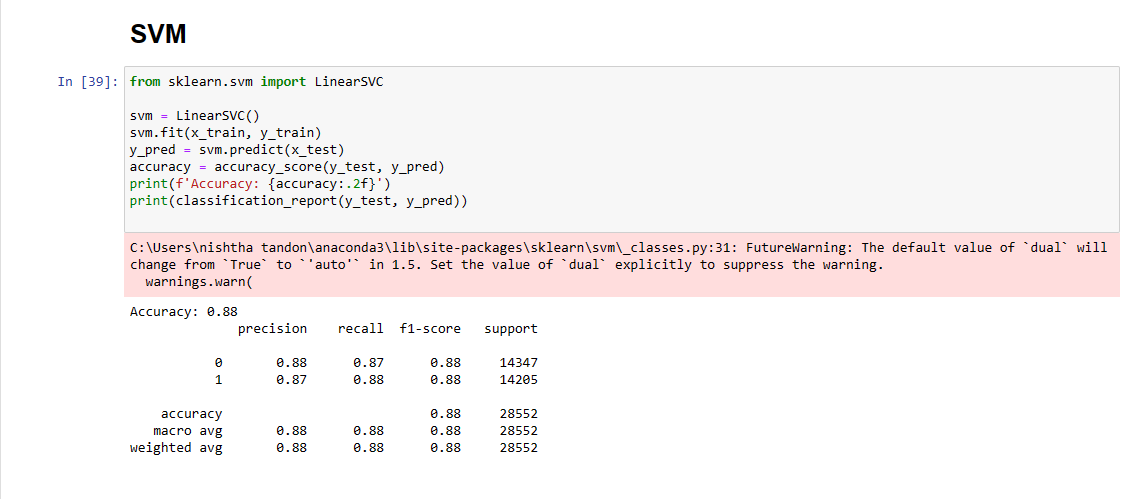
**Sentiment Analysis**: It’s particularly useful for large datasets and works well with textual data. [The model is simple and fast, making it a popular choice for initial baselines in sentiment analysis tasks](https://towardsdatascience.com/sentiment-analysis-introduction-to-naive-bayes-algorithm-96831d77ac91)

****

**Logistic Regression**:

**Principle**: A statistical method that models the probability of a binary outcome. It uses a logistic function to model a binary dependent variable.

**Sentiment Analysis**: In sentiment analysis, logistic regression can be used to predict the probability that a given text is positive or negative. [It’s effective for binary classification problems and can handle non-linear relationships in the data](https://towardsdatascience.com/sentiment-analysis-using-logistic-regression-and-naive-bayes-16b806eb4c4b)

****

Support Vector Machine (SVM):

Principle: SVM finds the hyperplane that best separates classes in a high-dimensional space. It uses kernel functions to transform the data and then finds the optimal boundary.

Sentiment Analysis: SVM is powerful for text classification tasks, including sentiment analysis. It’s capable of handling both linear and non-linear data. [For sentiment analysis, a linear kernel is often used when the data is linearly separable (just positive and negative sentiments](https://medium.com/scrapehero/sentiment-analysis-using-svm-338d418e3ff1).

In conclusion from the metrices we can see that Logistic Regression is better.

**REFRENCES**

**Dataset:**

[**https://www.kaggle.com/datasets/arhamrumi/amazon-product-reviews**](https://www.kaggle.com/datasets/arhamrumi/amazon-product-reviews)

**Study material:**

1. <https://medium.com/analytics-vidhya/amazon-fine-food-reviews-featurization-with-natural-language-processing-a386b0317f56>
2. <https://cs229.stanford.edu/proj2018/report/122.pdf>
3. <https://www.geeksforgeeks.org/amazon-product-reviews-sentiment-analysis-in-python/>
4. <https://medium.com/analytics-vidhya/amazon-fine-food-reviews-featurization-with-natural-language-processing-a386b0317f56>