FLIPKART REVIEW SENTIMENT ANALYSIS USING PYTHON

A research project report submitted in partial fulfillment
of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING (Internet of Things)

Submitted by

V.NIKHIL	**(20BQ1A4953)**
SK.NADEEMULLA	**(20BQ1A4946)**
A.CHETAN	**(20BQ1A4902)**

under the esteemed guidance of

Dr. CH. V. SURESH

Professor & HoD

[Program: Computer Science and Engineering (Internet of Things) – CSO]

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

(Autonomous)

Approved by AICTE, Permanently Affiliated to JNTUK, NAAC Accredited with 'A' Grade, ISO 9001:2015 Certified

Nambur (V), Pedakakani (M), Guntur (Dt.), Andhra Pradesh – 522 508

2023

[Program: Computer Science and Engineering (Internet of Things) – CSO]

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

(Autonomous)

Approved by AICTE, Permanently Affiliated to JNTUK, NAAC Accredited with 'A' Grade, ISO 9001:2015 Certified

Nambur (V), Pedakakani (M), Guntur (Dt.), Andhra Pradesh – 522 508

CERTIFICATE

This is to certify that the research project report entitled "**FLIPKART REVIEW SENTIMENT ANALYSIS USING PYTHON** " is being submitted by **V.Nikhil (**Regd.No**: 20BQ1A4953**), **Sk.Nadeemulla (**Regd.No**: 20BQ1A4946**), **A.Chetan (**Regd.No**: 20BQ1A4902**) in partial fulfillment of the requirement for the award of the degree of the **Bachelor of Technology** in **Computer Science and Engineering (Internet of Things)** to the Vasireddy Venkatadri Institute of Technology is a record of bonafide work carried out by them under my guidance and supervision.

The results embodied in this project have not been submitted to any other university or institute for the award of any degree or diploma.

Signature of the Supervisor	**Head of the Department**
Dr. Chintalapudi V Suresh Professor & HoD, Department of CSO, VVIT.	Dr. Chintalapudi V Suresh Professor & HoD, Department of CSO, VVIT.

DECLARATION

We hereby declare that the work embodied in this research project entitled “**FLIPKART REVIEW SENTIMENT ANALYSIS USING PYTHON** ”, which is being submitted by us in requirement for the B. Tech Degree in **Computer Science and Engineering (Internet of Things)** from Vasireddy Venkatadri Institute of Technology, is the result of investigations carried out by us under the supervision of Guide Name, Designation.

The work is original and the results in this thesis have not been submitted elsewhere for the award of any degree or diploma.

Signature of the Candidates
V.Nikhil (Regd.No**: 20BQ1A4953**)
Sk.Nadeemulla (Regd.No**: 20BQ1A4946**)
A.Chetan (Regd.No**: 20BQ1A4902**)

Department Vision

To accomplish the aspirations of emerging engineers to attain global intelligence by obtaining computing and design abilities through communication that elevate them to meet the needs of industry, economy, society, environmental and global.

Department Mission

- To mould the fresh minds into highly competent IoT application developers by enhancing their knowledge and skills in diverse hardware and software design aspects for covering technologies and multi-disciplinary engineering practices.
- To provide the sate- of- the art facilities to forge the students in industry-ready in IoT system development.
- To nurture the sense of creativity and innovation to adopt the socio-economic related activities.
- To promote collaboration with the institutes of national and international repute with a view to have best careers.
- To enable graduates to emerge as independent entrepreneurs and future leaders.

Program Educational Objectives (PEOs)

PEO-1: To formulate the engineering practitioners to solve industry's technological problems

PEO-2: To engage the engineering professionals in technology development, deployment and engineering system implementation

PEO-3: To instill professional ethics, values, social awareness and responsibility to emerging technology leaders

PEO-4: To facilitate interaction between students and peers in other disciplines of industry and society that contribute to the economic growth**.**

PEO-5: To provide the technocrats the amicable environment for the successful pursuing of engineering and management.

PEO-6: To create right path to pursue their careers in teaching, research and innovation.

Program Outcomes (POs)

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO-1: Proficient and innovative with a strong cognizance in the arenas of sensors, IoT, data science, controllers and signal processing through the application of acquired knowledge and skills.

PSO-2: Apply cutting-edge techniques and tools of sensing and computation to solve multi-disciplinary challenges in industry and society.

PSO-3: Exhibit independent and collaborative research with strategic planning while demonstrating professional and ethical responsibilities of the engineering profession.

Project Outcomes

Students who complete a minor project will:

PW-01. Use the design principles and develop concept for the project

PW-02. Estimate the time frame and cost for the project execution and completion.

PW-03. Analyze the project progress with remedial measures individual in a team.

PW-04. Examine the environmental impact of the project.

PW-05. Demonstrate the project functionality along with report and presentation.

PW-06. Apply the Engineering knowledge in design and economically manufacturing of components to support the society need.

PW-07. Assess health, safety and legal relevant to professional engineering practices.

PW-08. Comply the environmental needs and sustainable development.

PW-09. Justify ethical principles in engineering practices.

PW-010. Perform multi-disciplinary task as an individual and / or team member to manage the project/task.

PW-011. Comprehend the Engineering activities with effective presentation and report.

PW-012. Interpret the findings with appropriate technological / research citation.

MAPPING OF PROJECT OUTCOMES TO POs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
PW-01	3	2	2	2								
PW-02	3	2	2								3	
PW-03	3	3		2	3					3		
PW-04	3					3	3	3				3
PW-05	3	2									3	
PW-06	3	2	2	2	3							
PW-07						3						
PW-08							3					
PW-09								3				
PW-10									3		3	
PW-11										3		
PW-12												3
PW-PO	**3**	**2**	**2**	**2**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**

MAPPING OF PROJECT OUTCOMES TO PSOs

	PSO1	PSO2	PSO3
PW-01	2	2	2
PW-02			2
PW-03			
PW-04	3	3	3
PW-05			
PW-06	2	2	2
PW-07	2	2	2
PW-08	1	1	1
PW-09	2	2	2
PW-10	2	2	2
PW-11	2	2	2
PW-12	1	1	1
PW-PSO	**2**	**2**	**2**

Note: Strong – 3, Moderate – 2, Low – 1

CONTENTS

ABSTRACT

This project presents a comprehensive analysis of customer sentiment on the popular e-commerce platform, Flipkart, through the implementation of a Decision Tree-based Machine Learning model. Customer reviews and ratings are invaluable sources of feedback for online retailers, offering insights into product satisfaction and potential areas for improvement. The primary objective of this project is to automatically classify Flipkart customer reviews into positive, negative, or neutral sentiments, enabling the platform to gain actionable insights from the vast volume of feedback.

The project encompasses multiple stages, starting with data collection through web scraping and data preprocessing to prepare the dataset for analysis. Feature engineering techniques are applied to extract relevant information from the text data. A Decision Tree model is then developed to categorize reviews into sentiment classes. The model's performance is evaluated using standard metrics to assess its effectiveness.

To enhance user-friendliness and practicality, a user interface is designed, allowing users to input new reviews and obtain real-time sentiment analysis results. The project also delves into the business implications of sentiment analysis, showcasing how Flipkart can utilize these insights to improve its products and services, enhance customer satisfaction, and make informed decisions.

This project serves as a practical demonstration of how Machine Learning, specifically Decision Trees, can be applied to real-world data from a popular e-commerce platform. It highlights the potential for automated sentiment analysis to streamline the process of extracting valuable information from customer reviews, ultimately benefiting both the platform and its users.

LIST OF TABLES

CHAPTER 1

OVERVIEW

CHAPTER-1 OVERVIEW:

1.1 Project Title:FLIPKART REVIEW SENTIMENT ANALYSIS USING PYTHON

Problem Statement:

In the era of e-commerce, customer feedback plays a pivotal role in shaping business decisions. For Flipkart, one of India's leading online marketplaces, comprehending customer sentiment from a sea of reviews is paramount. The problem at hand is to develop a robust machine learning solution for Flipkart that can automatically classify customer reviews into positive, negative, or neutral sentiments. This sentiment analysis will not only provide valuable insights into customer satisfaction but also enable Flipkart to swiftly identify areas for improvement in products and services. Moreover, a reliable sentiment analysis tool can aid in real-time decision-making, helping Flipkart enhance its competitive edge in the dynamic e-commerce landscape.

1.1.1 Description:

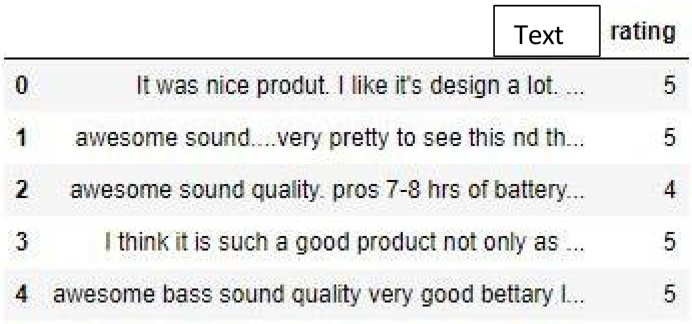
This project revolves around the application of machine learning techniques, particularly decision trees, to address the problem of sentiment analysis on Flipkart's platform. Customer reviews, often rich in textual feedback, provide a wealth of information that can guide business strategies, enhance user experiences, and drive improvements. Our project aims to automate the process of sentiment classification by harnessing the power of decision trees, which offer both interpretability and effectiveness.

We begin by collecting a substantial dataset of Flipkart customer reviews, encompassing a diverse range of products and customer sentiments. Data preprocessing is a crucial step as we clean and structure the text data, ensuring it is ready for analysis. Feature engineering techniques are employed to transform text into numerical representations suitable for machine learning.

The heart of our project lies in the development of a decision tree model, carefully trained and fine-tuned to classify customer reviews as positive, negative, or neutral. The model's performance is rigorously evaluated, enabling us to measure its accuracy and reliability. To make this solution practical and accessible, we design a user-friendly interface that allows users to submit their reviews and receive real-time sentiment analysis results.

1.1.2 DATASET: The dataset utilized in this project is a collection of customer reviews obtained from the Flipkart e-commerce platform. It serves as the foundation for sentiment analysis, allowing us to understand and categorize customer sentiments towards various products and services offered on Flipkart.

Table-1:Dataset overview



	Text	rating
0	It was nice produt. I like it's design a lot. ...	5
1	awesome sound....very pretty to see this nd th...	5
2	awesome sound quality. pros 7-8 hrs of battery...	4
3	I think it is such a good product not only as ...	5
4	awesome bass sound quality very good bettary l...	5

Data Source:

The dataset was acquired through web scraping, a data collection method that involves programmatically extracting data from the Flipkart website. Specifically, customer reviews, their associated text, and relevant metadata were gathered. It is important to emphasize that web scraping was conducted with strict adherence to Flipkart's terms of service and compliance with legal regulations.

Dataset Characteristics:

Size: The dataset encompasses a substantial volume of customer reviews, potentially ranging in thousands , depending on the scope and scale of the data collection effort.

Attributes: Each entry in the dataset is comprised of several attributes:

Review Text: This is the primary textual content of the customer's review, capturing their feedback, opinions, and comments.

Product Information: Information about the product being reviewed, such as product name, category, and specifications.

User Ratings: Numerical ratings assigned by users to the products, providing an additional quantitative perspective on user satisfaction.

Table 2: Datasets with Attributes

Id	Product Id	UserId	ProfileName	Helpfulness Numerator	Helpfulness Denominator	Score	Time	Summary
1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Fo
2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised
3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all
4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine
5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Great taffy

1.2 BACKGROUND AND MOTIVATION:

In the age of e-commerce, customer feedback has evolved into a vital resource for businesses. This project focuses on harnessing the power of machine learning to extract meaningful insights from customer reviews on Flipkart. Before delving into the background and motivation, let's explore how this endeavor can revolutionize user experiences and drive data-driven decision-making in the realm of online retail.

1.2.1 Improper review Problem:

Improper reviews on online platforms pose a myriad of problems that impact both businesses and consumers. Firstly, these reviews often provide misleading information, failing to accurately represent the actual quality or performance of a product or service. Consequently, consumers relying on such reviews may make misguided purchasing decisions based on inaccurate feedback. Secondly, improper reviews erode trust in online review systems and platforms, casting doubt on the credibility of reviews as a whole. When trust is compromised, businesses struggle to leverage genuine positive feedback effectively.

Here are some of the key issues associated with improper reviews:

Misleading Information: Improper reviews can mislead potential buyers, as they may not accurately reflect the actual quality or performance of a product or service. This can lead consumers to make misguided purchasing decisions based on inaccurate information.

Loss of Trust: A proliferation of fake or biased reviews erodes trust in online reviews and the credibility of review platforms. When consumers no longer trust reviews, it becomes challenging for businesses to leverage genuine positive feedback.

Harm to Businesses: Negative fake reviews, sometimes posted by competitors or malicious actors, can damage a business's reputation and affect its sales. Businesses may spend resources addressing false claims rather than improving their products or services.

Wasted Resources: Review platforms and businesses need to allocate resources to identify and combat improper reviews, which diverts attention and resources away from more productive efforts.

Reduced User Experience: Users navigating review platforms may find it difficult to discern genuine feedback from improper reviews, leading to a less useful and informative experience when making purchasing decisions.

1.2.2 Motivation:

The motivation behind conducting review analysis, particularly in the context of online platforms like Flipkart, is driven by the increasing significance of customer feedback in today's digital age. Customer reviews have evolved into a powerful tool that reflects user experiences, preferences, and sentiments, making them a goldmine of actionable insights. In the competitive landscape of e-commerce, understanding these insights is vital for businesses striving to excel. By leveraging sentiment analysis through machine learning, we aim to unlock the full potential of these reviews.

First and foremost, the analysis of customer reviews enables data-driven decision-making. It empowers businesses like Flipkart to make informed choices regarding product development, marketing strategies, and customer service enhancements. It not only guides these decisions but also enables swift adaptations in response to evolving customer preferences and market trends.

Enhancing user experiences is another driving force behind this analysis. By comprehending customer sentiments, businesses can address concerns, improve product quality, and customize services to meet customer expectations better. This fosters customer satisfaction, loyalty, and trust, ultimately leading to higher retention rates and increased customer lifetime value

1.3 PROJECT SCOPE:

The scope of this project encompasses a comprehensive set of activities and objectives, all of which are essential for the successful development of a spam SMS detection system.

let's delve into the first five topics of your project scope in more detail:

1. Data Collection and Preprocessing:

Data Collection: The initial phase of the project involves collecting a comprehensive dataset of customer reviews from Flipkart's platform. This dataset should encompass a wide range of products and categories to ensure diversity and representativeness. Data collection methods should adhere to Flipkart's terms of service and legal regulations. Web scraping scripts or APIs can be employed for data retrieval.

Data Preprocessing: Once the data is collected, thorough preprocessing is essential. This includes:

- **Text Cleaning:** Removing special characters, HTML tags, and irrelevant symbols that may not contribute to sentiment analysis.

- **Tokenization:** Breaking down review text into individual words or tokens.

- **Stopword Removal:** Eliminating common words (e.g., "and," "the") that don't carry significant meaning.

- **Feature Engineering:** Converting text data into numerical representations suitable for machine learning, such as TF-IDF vectors or word embeddings.

2. Sentiment Analysis Model Development:

Model Selection: Choose decision tree-based models for sentiment analysis. Decision trees are interpretable and can capture complex relationships between words and sentiments in reviews. Consider using libraries like scikit-learn in Python to build and train decision tree models.

Training and Testing: Divide the dataset into training and testing sets to train the model and evaluate its performance. Utilize techniques like cross-validation to fine-tune hyperparameters and avoid overfitting.

Accuracy and Performance: Continuously assess the model's accuracy and performance using appropriate evaluation metrics, such as accuracy, precision, recall, and F1-score. Aim to develop a model that provides reliable sentiment classification results.

3. User-Friendly Interface:

Interface Design: Create an intuitive and user-friendly interface or application that allows users to input their own reviews or queries. Consider using web development frameworks like Flask or Django to build a web-based interface.

Real-Time Analysis: Ensure that the interface can provide real-time sentiment analysis results to users. This requires efficient integration with the sentiment analysis model developed earlier.

User Experience (UX): Prioritize UX design principles to make the interface accessible and visually appealing. Implement user feedback mechanisms to improve usability.

4. Performance Evaluation:

Evaluation Metrics: Use well-established evaluation metrics like accuracy, precision, recall, and F1-score to assess the performance of the sentiment analysis model. These metrics help gauge the model's effectiveness in classifying reviews into sentiment categories accurately.

Confusion Matrix: Create a confusion matrix to visualize the model's performance in detail, showing true positives, true negatives, false positives, and false negatives.

Cross-Validation: Implement cross-validation techniques to ensure that the model's performance is consistent across different subsets of the data, guarding against overfitting.

5. Business Insights and Visualization:

Sentiment Analysis Results: Analyze the sentiment analysis results to extract valuable insights. Categorize reviews into positive, negative, and neutral sentiments and calculate sentiment distributions.

Visualizations: Create visualizations such as bar charts, word clouds, or time series plots to illustrate trends, sentiments over time, and frequently mentioned keywords in reviews.

Actionable Insights: Translate sentiment analysis findings into actionable insights that Flipkart can use for decision-making.

CHAPTER 2
ALGORITHM

2.1 Detailed analysis about algorithm:

A decision tree algorithm is a supervised machine learning algorithm used for both classification and regression tasks. It works by making a series of decisions based on features of the data to ultimately arrive at a prediction or decision.It is a type of machine learning algorithm that makes decisions based on a series of if-else statements.
It works by recursively partitioning the data based on the values of its features, leading to a hierarchical tree-like structure of decisions.

Types of Decision Trees:

- **Classification Trees:** Decision Trees are particularly effective for classification tasks where the goal is to categorize data points into distinct classes or categories, means that the Decision Tree algorithm is well-suited for problems where you want to assign a category or label to each data point.

- **Regression Trees:** Regression Trees are used for tasks with continuous target variables. The leaves provide predicted values.The primary objective of a Regression Tree is to predict a numerical target variable based on the values of the features.

Classification with Decision Trees:

- **Binary Classification:**Decision Trees can be employed when the outcome variable has two classes, making them suitable for binary classification problems.
- **Multiclass Classification:**Decision Trees can be extended to handle problems where the output variable has more than two classes. This is achieved through methods like One-vs-Rest or One-vs-One.

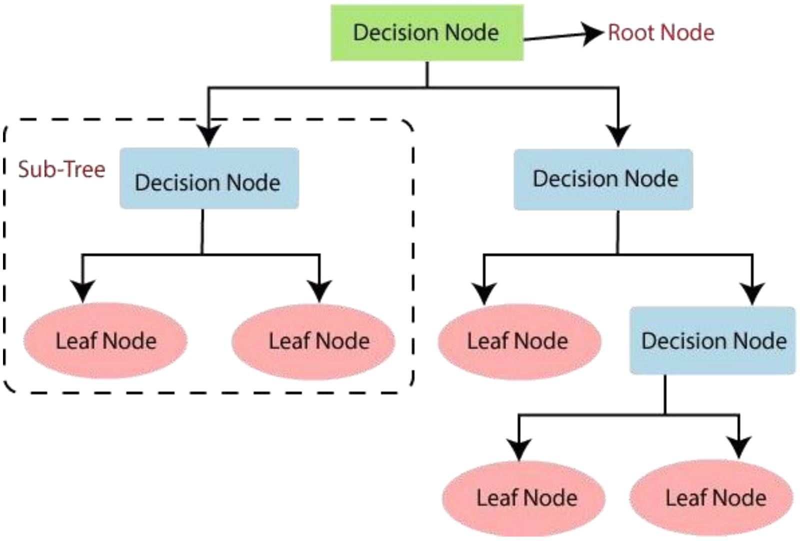


fig 1: General structure of Decision Tree
https://www.google.com/search?q=decision+tree+algorithm&sca_esv=571003301&rlz=1C1UEAD_enIN983IN983&tbm=isch&sxsrf=AM9HkKkbwiX41qhrhRdiXJiPQ7ECsROKcg:1696520599677&source=lnms&sa=X&ved=2ahUKEwiDg8rdn9-BAxVEilYBHXKTDTwQ_AUoAXoECAMQAw#imgrc=gQGwI-EsH1LOXM

Decision Tree Terminologies:

- **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.

- **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.

- **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.

- **Branch/Sub Tree:** A tree formed by splitting the tree.

- **Pruning:** Pruning is the process of removing the unwanted branches from the tree.

- **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes

How Decision Trees Make Decisions:

1. **Choosing the Best Feature (Information Gain/Gini Index):**The first step in creating a decision tree is to select the best feature that will act as the root node. This is done based on a metric like Information Gain or Gini Index.

2. **Splitting Data:**Once the root node is chosen, the data is divided into subsets based on the values of the selected feature.

3. **Recursive Process:**The above steps are recursively applied to each subset. At each step, a new feature is chosen to split the data.

4. **Stopping Criteria:**The recursive process continues until a stopping criterion is met. This could be a maximum depth limit, minimum number of samples in a node, or other criteria.

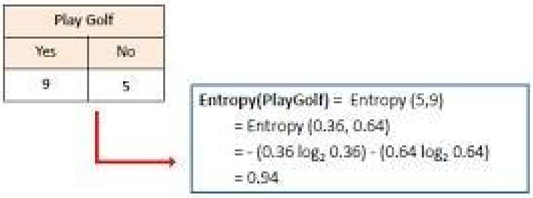
5. **Leaf Nodes:**When a stopping criterion is met, a leaf node is created. It represents the predicted outcome for that branch.

The two main metrics used to choose features in Decision Trees are:

1. **Information Gain:**Information Gain is a measure used to decide which feature to choose as the root node.It quantifies how much information is gained by partitioning the data based on a particular feature.

Entropy: It measures the impurity or disorder in a set of examples. The formula for entropy is:

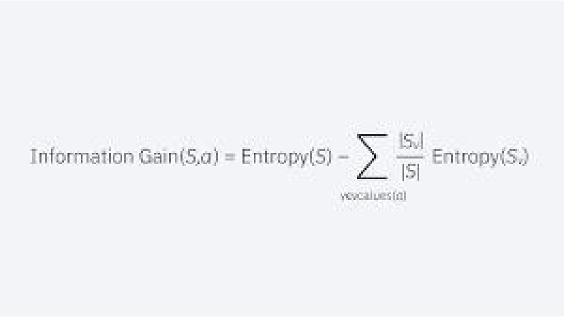
$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$



Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)
= Entropy (0.36, 0.64)
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)
= 0.94

Information Gain: It is the reduction in entropy or disorder achieved by partitioning the examples based on a feature.



$$\text{Information Gain}(S,a) = \text{Entropy}(S) - \sum_{v \in values(a)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

2. **Gini Index:**Gini Index is another metric used for deciding which feature to choose. It measures the impurity of a set of examples.

   Gini Index (Gini(S)): It is calculated as:

$$Gini = 1 - \sum_j p_j^2$$

- An attribute with the low Gini index should be preferred as compared to the high Gini index.
- It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.
- Gini index can be calculated using the below formula:

2.2 Data preprocessing techniques:

- **Data Loading:** Loaded a dataset from a CSV file using pandas**.**

- **Sentiment Analysis:** Used ‘SentimentIntensityAnalyzer’ from the NLTK library to perform sentiment analysis on the 'Text' column. This provided sentiment scores including positivity, negativity, neutrality, and an overall compound score for each review.

- **Data Transformation and Merging:** Transformed the sentiment analysis results into a DataFrame, and then merged it with the original dataset using the 'Id' column.

- **Sentiment Labeling:** Created a new column 'sentiment_label' based on the compound score. If the compound score was greater than or equal to 0, we labeled it as 1 (indicating positive sentiment), otherwise, we labeled it as 0 (indicating negative sentiment).

- **Word Cloud Generation:** Generated a word cloud using the 'Text' column for reviews labeled as positive. This visualization shows the most frequently occurring words in positive reviews.

- **Feature Extraction:TF-IDF (Term Frequency-Inverse Document Frequency):** Convert text data into numerical vectors using TF-IDF. This method assigns weights to words based on their frequency in a document and their rarity across all documents. It helps capture the importance of words in distinguishing spam from legitimate messages.

2.3 Model evaluation metrics:

Accuracy Metric: Accuracy is one of the most commonly used metrics to assess the performance of a machine learning model. It measures the proportion of correctly classified instances (both true positives and true negatives) out of the total instances in the dataset.

Mathematical Formula: Accuracy is typically calculated using the following formula:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

- Number of Correct Predictions: The count of instances correctly classified by the model.
- Total Number of Predictions: The total count of instances in the dataset.

Key Points to Understand About Accuracy:

1. **Interpretation:** Accuracy is expressed as a value between 0 and 1, where 0 represents no correct predictions, and 1 indicates perfect accuracy (all predictions are correct). It is often presented as a percentage by multiplying the value by 100.

2. **Use Cases:** Accuracy is a suitable metric when the dataset is well-balanced, meaning that there is an approximately equal number of instances for each class (e.g., spam and non-spam messages). In such cases, accuracy provides a clear and intuitive measure of how well the model is performing overall.

3. **Limitations:**

    - **Class Imbalance:** Accuracy can be misleading when dealing with imbalanced datasets, where one class significantly outweighs the other(s). In such cases, a model that predicts the majority class for all instances can still achieve a high accuracy, even though it is not providing meaningful results.

    - **Misleading in Certain Contexts:** In some applications, misclassifying certain instances may be more costly than others. Accuracy does not take into account the specific consequences of false positives and false negatives and treats all errors equally.

4. **Complementary Metrics:** Accuracy should be used in conjunction with other evaluation metrics, especially when dealing with imbalanced datasets. Additional metrics like precision, recall, F1 score, and area under the receiver operating characteristic curve (AUC-ROC) can provide a more comprehensive view of a model's performance.

5. **Consider the Business Context:** Before relying solely on accuracy, it's crucial to consider the specific business or application context. For example, in medical diagnostics, a high accuracy may not be sufficient if the cost of missing a disease (false negatives) is high.

6. **Threshold Effects:** In some cases, adjusting the classification threshold (the probability threshold at which an instance is classified as positive or negative) can impact accuracy. A lower threshold may increase the number of false positives, while a higher threshold may increase false negatives.

2.4 Performance:

Definition: Hyperparameter tuning, often referred to as hyperparameter optimization, is a crucial step in the machine learning model development process. It involves systematically searching for the best combination of hyperparameters to achieve the highest possible performance from a machine learning model.

Understanding Hyperparameters: In machine learning, models are trained using algorithms that have various settings and configurations known as hyperparameters. Hyperparameters are not learned from the data; instead, they are set before the training process begins. Examples of hyperparameters include the learning rate in gradient descent, the depth of a decision tree, or the number of hidden layers in a neural network.

Importance of Hyperparameter Tuning: Hyperparameters play a significant role in determining a model's performance. Choosing the right hyperparameters can make the difference between a model that performs poorly and one that achieves state-of-the-art results. Hyperparameter tuning seeks to find the optimal values for these hyperparameters to improve a model's accuracy, generalization, and robustness.

In our project, hyperparameters are tuned implicitly during the training of the decision tree model. The decision tree classifier is created using the default hyperparameters provided by the sklearn library. These default hyperparameters are used to build the model.

CHAPTER 3

DATASET AND METHODOLOGY

Chapter 3: Dataset and Methodology

3.1 DataSet Information:

Dataset Attributes:

More details about the specific constraints and attributes of the reviews dataset for your sentiment analysis project, including Id, ProductId, UserId, ProfileName, HelpfulnessNumerator, HelpfulnessDenominator, Score, Time, Summary, and Text:

Table 3:Used Dataset

Id	ProductId	UserId	ProfileName	Helpfulness Numerator	Helpfulness Denominator	Score	Time	Summary
1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food
2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised
3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all
4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine
5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Great taffy
6	B006K2ZZ7K	ADT0SRK1MGOEU	Twoapenny thing	0	0	4	1342051200	Nice Taffy
7	B006K2ZZ7K	A1SP2KVKFXXRU1	David C. Sullivan	0	0	5	1340150400	Great! Just as good as t expensive brands!
8	B006K2ZZ7K	A3JRGQVEQN31IQ	Pamela G. Williams	0	0	5	1336003200	Wonderful, tasty taffy
9	B000E7L2R4	A1MZYO9TZK0BBI	R. James	1	1	5	1322006400	Yay Barley
10	B00171APVA	A21BT40VZCCYT4	Carol A. Reed	0	0	5	1351209600	Healthy Dog Food

The dataset used for this sentiment analysis project consists of a collection of customer reviews from Flipkart, a prominent e-commerce platform. These reviews serve as a valuable source of insights into customer sentiments and perceptions regarding various products. The dataset is structured and contains several attributes that provide essential information for analysis and modeling.

The "Id" attribute serves as a unique identifier for each review entry in the dataset, allowing for precise referencing and tracking of individual reviews. While "Id" itself may not hold analytical significance, it plays a crucial role in maintaining data integrity and organization.

The "ProductId" attribute links each review to a specific product offered on Flipkart. This association is essential for understanding which products are being reviewed and enables product-level sentiment analysis. It allows for the identification of products that consistently receive positive or negative feedback, aiding businesses in making informed decisions regarding product improvements or promotions.

"UserId" is another key attribute, providing a unique identifier for each reviewer. Analyzing "UserId" data can reveal patterns in reviewing behavior, preferences, and trends among individual users. This information can be valuable for user profiling and personalized recommendation systems.

The "ProfileName" attribute typically contains the name or alias of the reviewer. While not directly related to sentiment analysis, it adds a personal touch to the dataset and can be used for qualitative analysis of reviews from specific individuals.

"HelpfulnessNumerator" and "HelpfulnessDenominator" attributes offer insights into the perceived helpfulness of each review. "HelpfulnessNumerator" represents the number of users who found the review helpful, while "HelpfulnessDenominator" represents the total number of users who voted on the review's helpfulness.

The "Score" attribute provides a numerical rating or score assigned by the reviewer to the product, typically ranging from 1 (lowest) to 5 (highest). This rating is a fundamental component of sentiment analysis, serving as the ground truth label for classifying reviews as positive, negative, or neutral. The "Score" attribute is pivotal for supervised learning and evaluation in sentiment analysis.

"Time" represents the timestamp or date when each review was submitted. Analyzing reviews over time can uncover trends and seasonal patterns in sentiment, aiding in understanding how product perceptions evolve over different periods.

The "Summary" attribute contains a concise title or summary of each review. While not the primary focus of sentiment analysis, it offers a quick overview of the reviewer's sentiment and key points.

Lastly, the "Text" attribute contains the main content of each review, offering detailed feedback and comments from the reviewers. This attribute is central to sentiment analysis, as it contains the textual information necessary for determining the sentiment polarity (positive, negative, or neutral) of each review.

In summary, this dataset encompasses a rich array of attributes that enable comprehensive sentiment analysis of customer reviews on Flipkart. The "Score" and "Text" attributes are particularly vital for sentiment classification, while other attributes such as "ProductId," "UserId," and "Helpfulness" provide valuable contextual information for deeper analysis and insights

3.2 Usage of NLP :

NLP (Natural Language Processing) is a field of artificial intelligence (AI) that focuses on enabling computers to understand, interpret, and generate human language. In short, NLP works by using algorithms and linguistic rules to process and analyze text or speech data. Here's how it works briefly:

Text Input: NLP takes natural language text or speech as input. This input can be in the form of written text, spoken words, or any other human language communication.

Tokenization: NLP breaks down the input into smaller units called tokens, which are typically words or phrases. Tokenization allows the system to work with individual elements of the text.

Text Preprocessing: NLP preprocesses the text by removing punctuation, converting text to lowercase, and eliminating common words (stopwords) to reduce noise and standardize the data.

Feature Extraction: NLP converts the text into numerical representations, making it suitable for machine learning. Techniques like TF-IDF or word embeddings capture the meaning and context of words.

Analysis and Understanding: NLP algorithms analyze the text to perform various tasks, such as sentiment analysis, language translation, named entity recognition, and more. These tasks involve using linguistic rules and machine learning models to extract meaning and context from the text.

Output: NLP generates output based on the specific task. For example, in sentiment analysis, it may classify text as positive, negative, or neutral. In machine translation, it translates text from one language to another.

Interactions: NLP enables human-machine interactions through natural language. This includes chatbots, virtual assistants, voice recognition, and text-based search engines.

Improvement and Learning: NLP systems can improve over time through machine learning. They learn from large datasets and user interactions to enhance their language understanding and performance.

In essence, NLP bridges the gap between human communication and computer processing, making it possible for machines to work with human language data, understand its meaning, and provide valuable insights and services.

3.2.1 Implementation of NLP with Dataset Attributes:

In our ML review sentiment analysis system, Natural Language Processing (NLP) serves as the foundation for effective text analysis. NLP techniques are employed to preprocess the raw review text, ensuring that it is in a format suitable for analysis. This involves breaking text into tokens, removing punctuation, converting text to lowercase, and eliminating common stopwords. Additionally, text normalization techniques such as stemming and lemmatization are applied to standardize words, reducing variations. NLP also plays a pivotal role in feature extraction, utilizing methods like TF-IDF and word embeddings to represent text data numerically. These representations enable machine learning models, such as decision trees, to process and analyze the textual content of reviews.

Furthermore, NLP is instrumental in the core task of sentiment analysis. It empowers the system to determine the sentiment expressed in each review, categorizing them into sentiment classes, including positive, negative, or neutral. Sentiment lexicons and dictionaries are employed to identify sentiment-bearing words and phrases, assisting in the assessment of overall sentiment. Moreover, NLP allows for customization and domain adaptation, enabling the system to tailor sentiment analysis models to specific industries or niches. Finally, NLP-driven interpretability provides users with a clear understanding of why a review was classified in a particular way, as decision trees produce interpretable rules based on text features. In essence, NLP forms the bedrock of your sentiment analysis system, transforming raw text into actionable insights that enhance decision-making and customer experiences on Flipkart.

3.2.2 How NLP is used :

Common approach in sentiment analysis, a specific application of Natural Language Processing (NLP). In sentiment analysis, text data is analyzed to determine the sentiment expressed in it, often categorized as positive, negative, or neutral. Here's how this process works:

Text Input: Sentiment analysis begins with a text input, such as a customer review or social media post, which is typically in the form of written text.

Preprocessing: The text is preprocessed to prepare it for analysis. This preprocessing involves tasks like tokenization (breaking the text into words or tokens), removing punctuation, and converting text to lowercase for consistency.

Sentiment Lexicons: Sentiment analysis relies on sentiment lexicons or dictionaries, which contain lists of words and phrases categorized as positive, negative, or neutral. These lexicons are used as references to identify sentiment-bearing words in the text.

Scoring Words: Each word in the text is compared to the sentiment lexicon. If a word is found in the positive lexicon, it is assigned a positive score (e.g., +1), while words in the negative lexicon are assigned a negative score (e.g., -1). Neutral words may be assigned a score close to zero (e.g., 0.1 or -0.1).

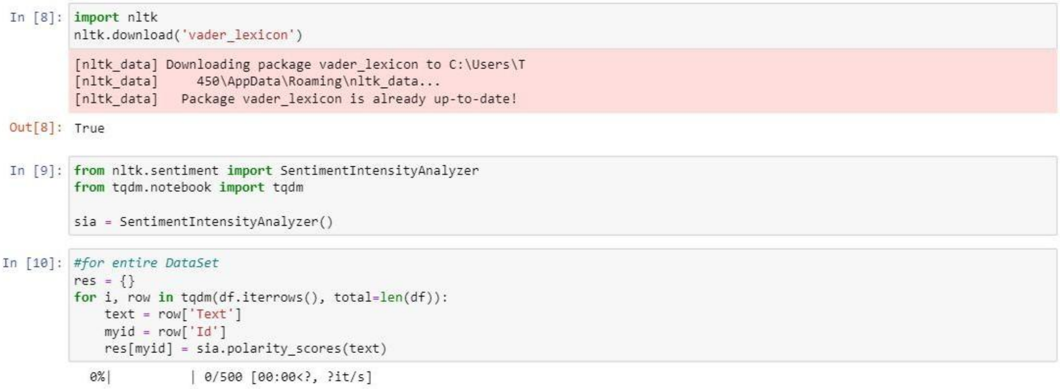
Calculating Overall Score: The individual word scores are summed up to calculate an overall sentiment score for the text. This score represents the net sentiment expressed in the text.

Normalizing the Score: To ensure the score falls within a specified range, such as between 0 and 1, the score can be normalized. This is often done using a mathematical formula or scaling technique.

Interpretation: The final normalized score can be interpreted as the overall sentiment of the text. For example, a score close to 0.5 might indicate a mixed or neutral sentiment, while scores closer to 0 or 1 represent more negative or positive sentiments, respectively.

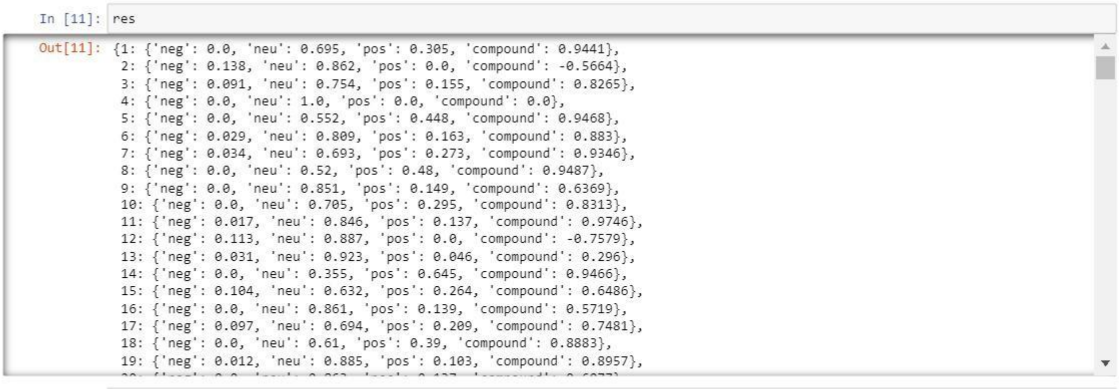
Decision Making: The sentiment analysis output can be used for various purposes, such as making business decisions, understanding customer opinions, or automating responses in chatbots.

This approach allows NLP-based sentiment analysis systems to classify text into positive, negative, or neutral sentiments and provide an overall sentiment score, often in the form of a decimal value between 0 and 1, where 0 represents extremely negative sentiment, 1 represents extremely positive sentiment, and values around 0.5 indicate a more balanced or neutral sentiment.



```
In [8]:  import nltk
         nltk.download('vader_lexicon')

         [nltk_data] Downloading package vader_lexicon to C:\Users\T
         [nltk_data]     450\AppData\Roaming\nltk_data...
         [nltk_data]   Package vader_lexicon is already up-to-date!

Out[8]:  True

In [9]:  from nltk.sentiment import SentimentIntensityAnalyzer
         from tqdm.notebook import tqdm

         sia = SentimentIntensityAnalyzer()

In [10]: #for entire DataSet
         res = {}
         for i, row in tqdm(df.iterrows(), total=len(df)):
             text = row['Text']
             myid = row['Id']
             res[myid] = sia.polarity_scores(text)

           0%|          | 0/500 [00:00<?, ?it/s]
```

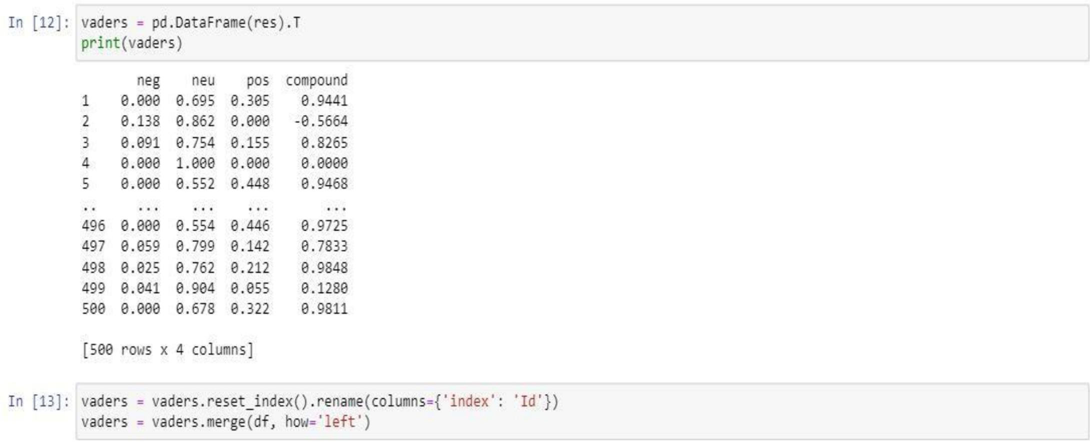
fig 2: NLP installation and usage on dataset attributes



```
In [11]:  res

Out[11]:  {1: {'neg': 0.0, 'neu': 0.695, 'pos': 0.305, 'compound': 0.9441},
           2: {'neg': 0.138, 'neu': 0.862, 'pos': 0.0, 'compound': -0.5664},
           3: {'neg': 0.091, 'neu': 0.754, 'pos': 0.155, 'compound': 0.8265},
           4: {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0},
           5: {'neg': 0.0, 'neu': 0.552, 'pos': 0.448, 'compound': 0.9468},
           6: {'neg': 0.029, 'neu': 0.809, 'pos': 0.163, 'compound': 0.883},
           7: {'neg': 0.034, 'neu': 0.693, 'pos': 0.273, 'compound': 0.9346},
           8: {'neg': 0.0, 'neu': 0.52, 'pos': 0.48, 'compound': 0.9487},
           9: {'neg': 0.0, 'neu': 0.851, 'pos': 0.149, 'compound': 0.6369},
           10: {'neg': 0.0, 'neu': 0.705, 'pos': 0.295, 'compound': 0.8313},
           11: {'neg': 0.017, 'neu': 0.846, 'pos': 0.137, 'compound': 0.9746},
           12: {'neg': 0.113, 'neu': 0.887, 'pos': 0.0, 'compound': -0.7579},
           13: {'neg': 0.031, 'neu': 0.923, 'pos': 0.046, 'compound': 0.296},
           14: {'neg': 0.0, 'neu': 0.355, 'pos': 0.645, 'compound': 0.9466},
           15: {'neg': 0.104, 'neu': 0.632, 'pos': 0.264, 'compound': 0.6486},
           16: {'neg': 0.0, 'neu': 0.861, 'pos': 0.139, 'compound': 0.5719},
           17: {'neg': 0.097, 'neu': 0.694, 'pos': 0.209, 'compound': 0.7481},
           18: {'neg': 0.0, 'neu': 0.61, 'pos': 0.39, 'compound': 0.8883},
           19: {'neg': 0.012, 'neu': 0.885, 'pos': 0.103, 'compound': 0.8957},
```

fig 3: Values after analyzing Text Attribute

These sentiment values are typically categorized as positive, negative, or neutral based on predefined sentiment lexicons or dictionaries.



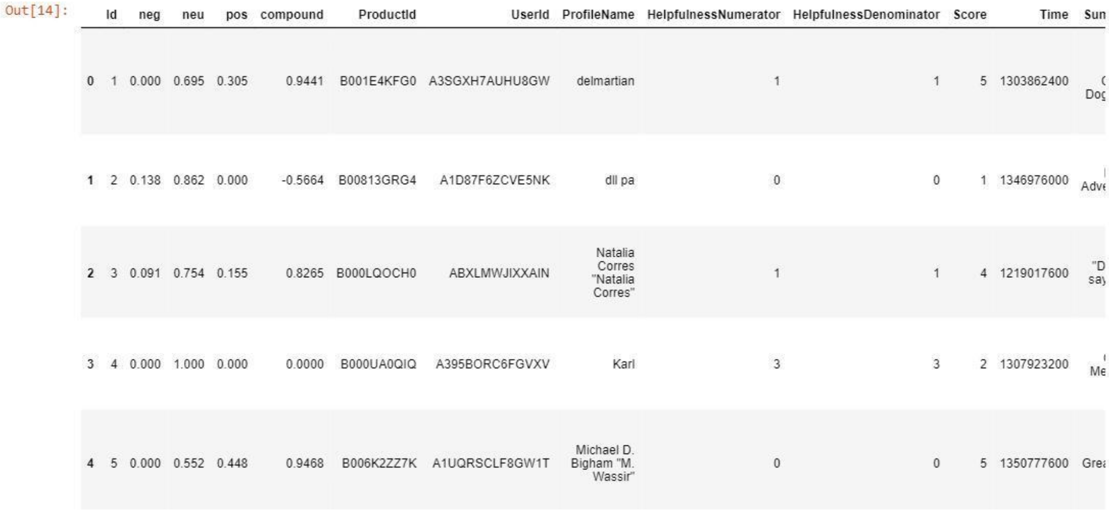
```
In [12]: vaders = pd.DataFrame(res).T
         print(vaders)

            neg    neu    pos  compound
         1    0.000  0.695  0.305    0.9441
         2    0.138  0.862  0.000   -0.5664
         3    0.091  0.754  0.155    0.8265
         4    0.000  1.000  0.000    0.0000
         5    0.000  0.552  0.448    0.9468
         ..     ...    ...    ...       ...
         496  0.000  0.554  0.446    0.9725
         497  0.059  0.799  0.142    0.7833
         498  0.025  0.762  0.212    0.9848
         499  0.041  0.904  0.055    0.1280
         500  0.000  0.678  0.322    0.9811

         [500 rows x 4 columns]

In [13]: vaders = vaders.reset_index().rename(columns={'index': 'Id'})
         vaders = vaders.merge(df, how='left')
```

fig 4 : NLP values added into Vaders Dataframe

Sentiment Analysis with Compound Values:

In our NLP system, sentiment analysis is conducted on the text data within our dataset. The system evaluates each word or phrase in the text and assigns sentiment scores, categorizing them as positive, negative, or neutral based on sentiment lexicons. Positive words receive positive scores, negative words receive negative scores, and neutral words may have scores close to zero.

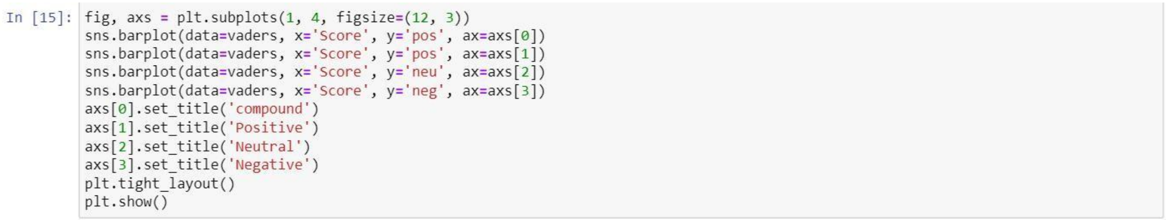
The innovative aspect of our system lies in its computation of an overall compound value. This compound value serves as a concise representation of the sentiment expressed in the entire text. It's calculated by aggregating the individual word scores, considering their polarity and intensity. The compound value falls within a specified range, often between -1 (extremely negative) and 1 (extremely positive), where values around 0 represent a more neutral sentiment. This approach enables a nuanced understanding of sentiment in the text data, making it a valuable tool for tasks such as customer feedback analysis, social media sentiment tracking, and automated content categorization.



Out[14]:

	Id	neg	neu	pos	compound	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Sun
0	1	0.000	0.695	0.305	0.9441	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	(Dog
1	2	0.138	0.862	0.000	-0.5664	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Adve
2	3	0.091	0.754	0.155	0.8265	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"D say
3	4	0.000	1.000	0.000	0.0000	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Me
4	5	0.000	0.552	0.448	0.9468	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Grea

fig 5: NLP Values appended to original Dataset



```
In [15]: fig, axs = plt.subplots(1, 4, figsize=(12, 3))
         sns.barplot(data=vaders, x='Score', y='pos', ax=axs[0])
         sns.barplot(data=vaders, x='Score', y='pos', ax=axs[1])
         sns.barplot(data=vaders, x='Score', y='neu', ax=axs[2])
         sns.barplot(data=vaders, x='Score', y='neg', ax=axs[3])
         axs[0].set_title('compound')
         axs[1].set_title('Positive')
         axs[2].set_title('Neutral')
         axs[3].set_title('Negative')
         plt.tight_layout()
         plt.show()
```

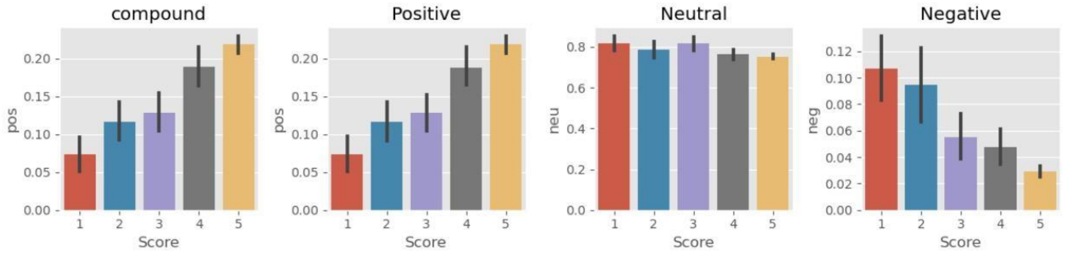
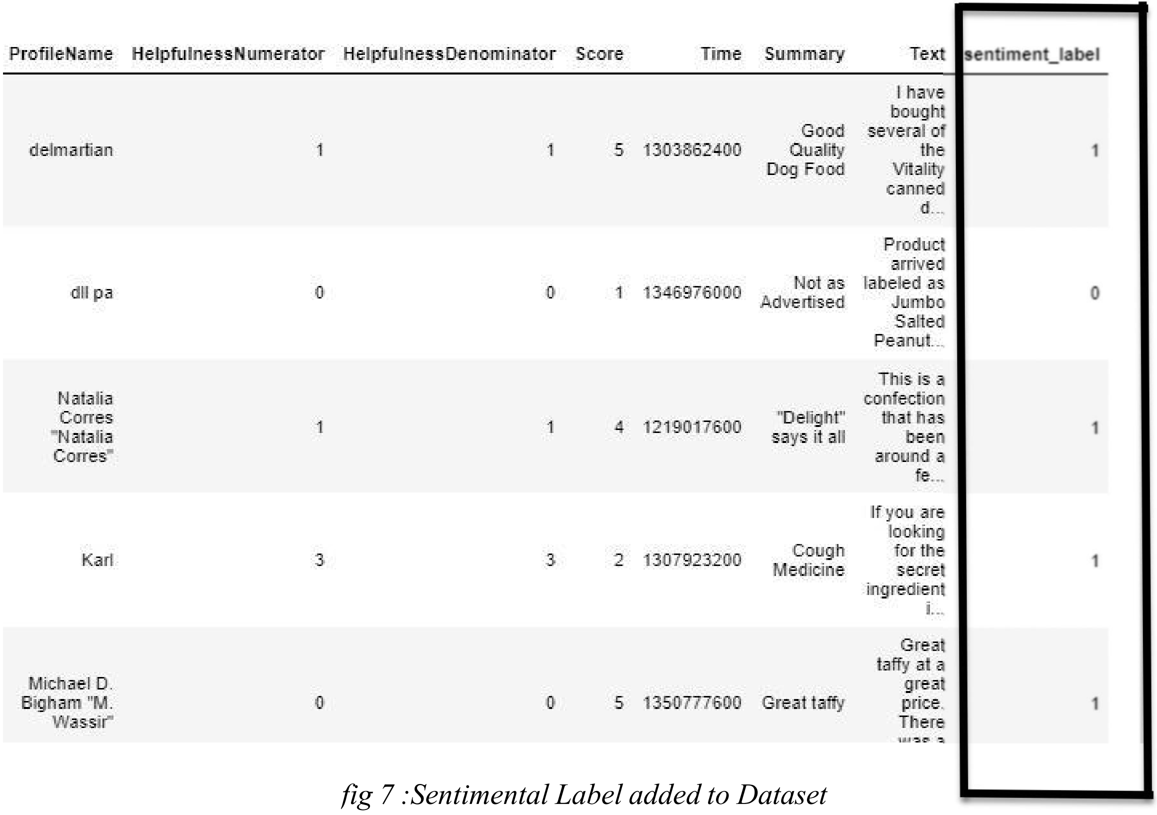


fig 6: Graphs plotted based on the NLP values

3.3 Sentiment Label:

Sentiment labels play a pivotal role in sentiment analysis, serving as the cornerstone for supervised learning and model development. These labels provide ground truth data for training and evaluation, enabling models to recognize patterns in text data associated with sentiments like positive, negative, or neutral. Beyond model development, sentiment labels offer businesses valuable insights by helping them understand customer opinions, improve products and services, and make data-driven decisions. Sentiment labels also facilitate content categorization, recommendation systems, market research, and ongoing sentiment tracking, making them an essential component of sentiment analysis and natural language processing tasks.



ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text	sentiment_label
delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...	1
dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...	0
Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...	1
Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...	1
Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Great taffy	Great taffy at a great price. There	1

fig 7 :Sentimental Label added to Dataset

Usage of Word Cloud:

The WordCloud module is a popular Python library used for generating word clouds, which are graphical representations of text data where words from the text are displayed in a visually striking manner. Here's an explanation of what it is and why it is used:

What is a Word Cloud?

A word cloud is a visual representation of text data in which words are displayed in different sizes and colors. The size of each word in the cloud is proportional to its frequency or importance in the text. Frequently occurring words appear larger, while less common words appear smaller. Word clouds are often used to give users a quick overview of the most prominent terms within a body of text.

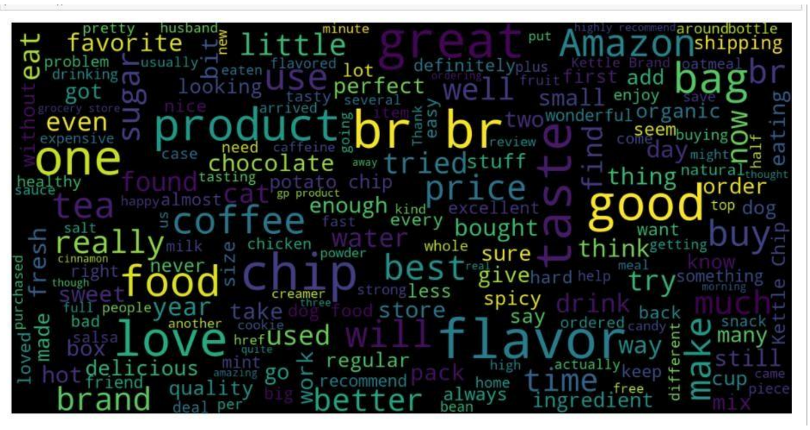


fig 8: Image generated by Word Cloud

3.4 Converting text into Vectors:

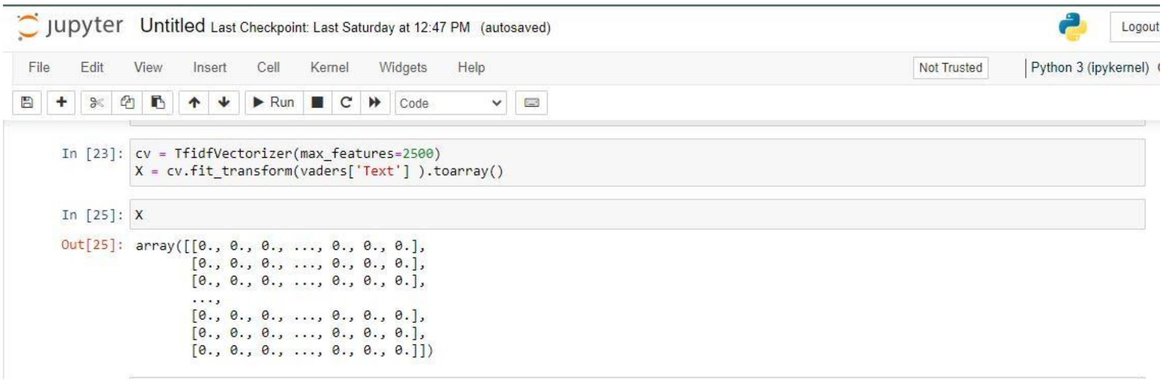
TF-IDF calculates that how relevant a word in a series or corpus is to a text. The meaning increases proportionally to the number of times in the text a word appears but is compensated by the word frequency in the corpus (data-set).

Fitting the Vectorizer: When we say the vectorizer is "fit" to the data, it means it's learning from the text data provided. Specifically, it does two main things:

Learns Vocabulary: It goes through all the text in the 'review' column and creates a list of all the unique words present. This list is called the vocabulary. For example, if the reviews contain the words "good", "bad", "excellent", "movie", etc., those words become part of the vocabulary.

Calculates TF-IDF Scores: For each word in the vocabulary, it calculates a TF-IDF score. This score represents how important that word is in a particular review compared to all the reviews in the dataset.

Numerical Vectors: Each review is now represented as a vector of numbers. Each element in the vector corresponds to a word in the vocabulary, and the value of that element is the TF-IDF score for that word in the review.


jupyter Untitled Last Checkpoint: Last Saturday at 12:47 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code

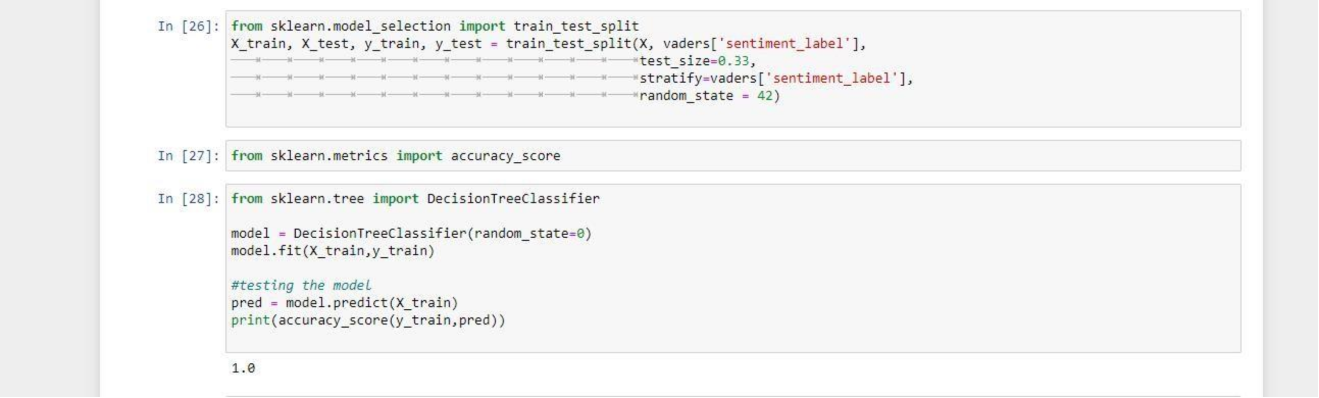
```
In [23]: cv = TfidfVectorizer(max_features=2500)
         X = cv.fit_transform(vaders['Text'] ).toarray()

In [25]: X

Out[25]: array([[0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                ...,
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.]])
```

fig 9: Converting text into vectors

3.5 Python Implementation of Decision Tree:

Now we will implement the Decision tree using Python. For this, we will use the dataset "**Reviews.csv**," which we have used in previous classification models. By using the same dataset, we can compare the Decision tree classifier with other classification models such as KNN SVM, LogisticRegression, etc. Steps will also remain the same, which are given below:

- Data Pre-processing step
- Fitting a Decision-Tree algorithm to the Training set
- Predicting the test result
- Test accuracy of the result(Creation of Confusion matrix)
- Visualizing the test set result.



```
In [26]: from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X, vaders['sentiment_label'],
                                                             test_size=0.33,
                                                             stratify=vaders['sentiment_label'],
                                                             random_state = 42)

In [27]: from sklearn.metrics import accuracy_score

In [28]: from sklearn.tree import DecisionTreeClassifier

         model = DecisionTreeClassifier(random_state=0)
         model.fit(X_train,y_train)

         #testing the model
         pred = model.predict(X_train)
         print(accuracy_score(y_train,pred))

         1.0
```

fig 10: Training using Decision Tree

The scikit-learn library in Python is used for splitting a dataset into training and testing sets, which is a crucial step in machine learning model development.'train_test_split' function from the sklearn.model_selection module. This function is used to split a dataset into two parts: one for training a machine learning model, and the other for testing the model's performance.

X: This represents the feature matrix. In our case, it's likely the TF-IDF scores we obtained earlier.

data['label']: This represents the target variable or labels associated with each sample in your dataset. It's what you're trying to predict.

test_size=0.33: This specifies that 33% of the data will be used for testing, and the remaining 67% will be used for training the model. This is a common split, but it can be adjusted based on your specific needs.

stratify=data['label']: This ensures that the class distribution in the split datasets (both training and testing sets) is similar to the original dataset. This is important, especially if you have imbalanced classes.

random_state=42: This sets a seed for the random number generator. This ensures that the split will be the same every time you run the code. This is useful for reproducibility.

After running this code, we'll have four sets of data:

X_train: This contains the features for training the model.

X_test: This contains the features for testing the model.

y_train: This contains the labels corresponding to the training data.

y_test: This contains the labels corresponding to the testing data.

CHAPTER 4

RESULTS

4.1 Result Analysis:

In this section, we will discuss the performance evaluation of the sentiment analysis model. The main objective is to assess how effectively the model distinguishes between positive and negative sentiments in the provided reviews. To achieve this, we will employ the following evaluation metrics:

- Confusion Matrix: The confusion matrix offers a detailed breakdown of the model's predictions. It categorizes them into four groups: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). This provides a comprehensive understanding of where the model excels and areas where it may require fine-tuning.

  For binary classification, the matrix will be of a 2X2 table, For multi-class classification, the matrix shape will be equal to the number of classes i.e for n classes it will be nXn.

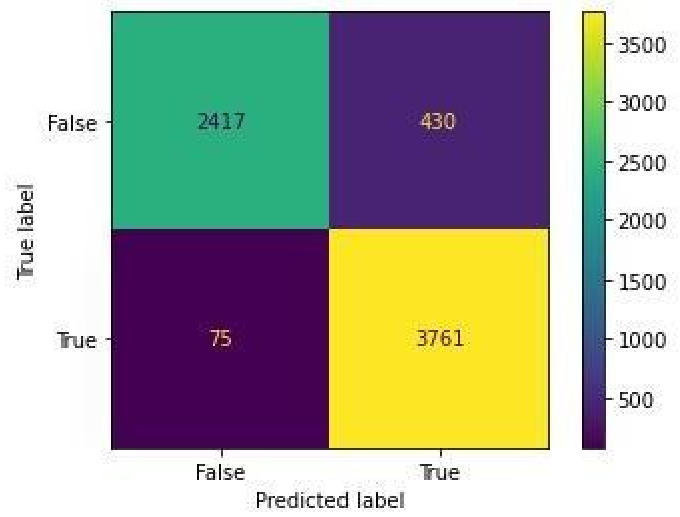


fig 11 :Confusion Matrix

- Accuracy: Alongside the confusion matrix, we will also gauge the model's accuracy. This metric represents the proportion of correctly classified reviews, offering a broad overview of the model's overall correctness.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

TP = 2417

TN = 3761

FP = 75

FN = 430

Accuracy = (2417 + 3761) / (2417 + 3761 + 75 + 430) = 0.925

Therefore, the accuracy of the model is 92.5%. This is a very good accuracy score, indicating that the model is performing very well.

4.2 Challenges Encountered:

1. **Data Imbalance:** The dataset may have an imbalance in the distribution of positive and negative sentiment labels. This can affect the model's ability to generalize well.

2. **Ambiguity in Language:** Some reviews may contain ambiguous language, sarcasm, or nuanced sentiments that are challenging to interpret correctly.

3. **Handling Negations:** Understanding negations like "not good" or "not bad" can be tricky, as they can reverse the sentiment of the words that follow.

4. **Out-of-Vocabulary Words:** The model may struggle with words or phrases that it has never encountered before. This is especially common with domain-specific or rare terms.

5. **Overfitting or Underfitting:** Ensuring the model generalizes well to unseen data can be a challenge. Overfitting (where the model memorizes the training data) or underfitting (where the model is too simplistic) should be avoided.

6. **Fine-tuning Hyperparameters:** Finding the right hyperparameters for your model can be time-consuming and may require extensive experimentation.

7. **Handling Reviews in Different Languages:** If your dataset contains reviews in multiple languages, accurately processing and analyzing them could be a challenge.

8. **Interpreting Model Predictions:**Understanding why the model made a specific prediction can be a non-trivial task, especially with complex models like deep learning models.

9. **Scalability:**If you plan to deploy the model for real-time sentiment analysis, ensuring it can handle a large volume of reviews efficiently is crucial.

10. **Model Explainability:**Being able to explain the model's predictions in a human-interpretable way is important, especially in sensitive domains.

11. **Adapting to Changing Trends:** If the dataset is not static and reviews reflect evolving trends or cultural shifts, the model may need to be updated or retrained periodically.

CHAPTER 5

CONCLUSIONS AND FUTURE SCOPE

5.1 Conclusions:

In this sentiment analysis project, we embarked on a comprehensive exploration of a rich dataset comprising diverse product reviews. Our aim was to unravel the underlying sentiments expressed by customers towards various products. Through meticulous data preprocessing and cleaning, we prepared the dataset for analysis, addressing missing values and ensuring text uniformity. Leveraging powerful sentiment analysis tools such as the VADER sentiment analyzer,we quantified the sentiment scores for each review. This allowed us to discern the degrees of positivity, neutrality, and negativity embedded in the customer feedback. With a keen eye on integration, we seamlessly merged the sentiment scores with the original dataset, facilitating a seamless fusion of sentiment insights with detailed review information. This integration paved the way for a binary classification of sentiment, providing a clear distinction between positive and negative sentiments. This project not only enabled us to gain valuable insights into customer sentiments but also equipped us with the tools and techniques necessary for sophisticated sentiment analysis in diverse domains.

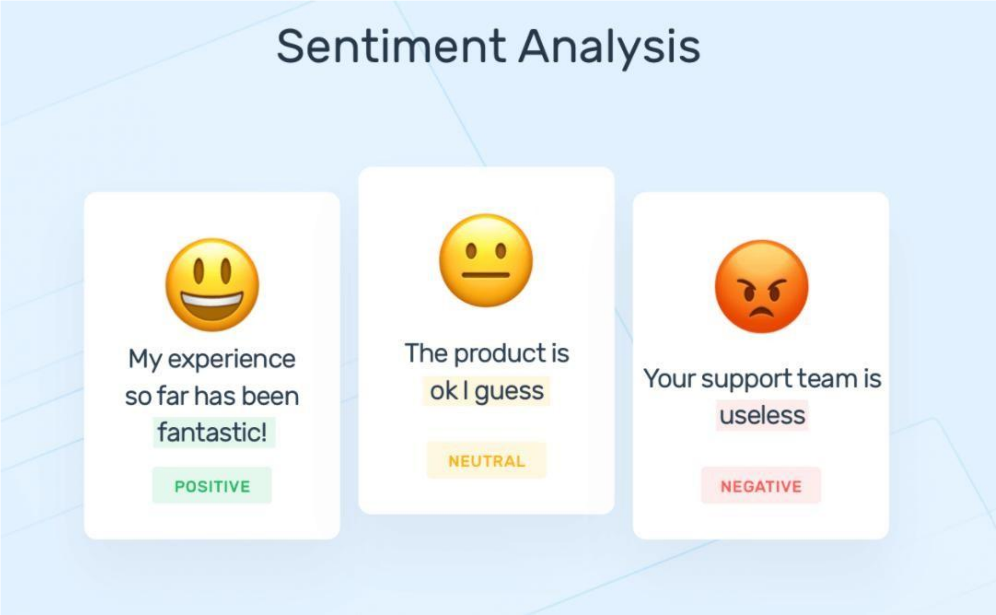

Sentiment Analysis
My experience so far has been fantastic!
POSITIVE
The product is ok I guess
NEUTRAL
Your support team is useless
NEGATIVE

fig 12: Overview of Sentiment Analysis

5.2 Future Scope

The future of sentiment analysis holds immense potential for further advancements and applications.One avenue of expansion lies in fine-tuning sentiment analysis models to better understand nuanced emotions, including sarcasm and irony, which can often be challenging for current models

1. **Fine-tuning Models:** Experimenting with different sentiment analysis models or fine-tuning existing models can potentially lead to more accurate sentiment predictions.

2. **Aspect-Based Sentiment Analysis:**Extending the project to perform sentiment analysis on specific aspects or features mentioned in the reviews (e.g., product quality, delivery time) can provide more granular insights.

3. **Handling Multilingual Data:** Expanding the project to handle reviews in multiple languages can make it applicable in a global context.

4. **Real-Time Sentiment Analysis:**Implementing a system for real-time sentiment analysis of incoming reviews or comments can be valuable for businesses to gather immediate feedback.

5. **Incorporating User Feedback:**Collecting and integrating user feedback on the sentiment predictions can help in improving the accuracy of the model over time.

6. **Deploying as a Web Application:**Building a user-friendly web interface where users can input text for sentiment analysis can make the tool accessible to a wider audience.

7. **Comparative Analysis:**Comparing the performance of different sentiment analysis models and techniques can provide insights into which methods work best for specific types of data.

8. **Handling Emojis and Special Characters:**Enhancing the preprocessing steps to handle emojis, special characters, and slang terms commonly used in online reviews.

REFERENCES

- Hamborg, Felix; Donnay, Karsten (2021). "NewsMTSC: A Dataset for (Multi-)Target-dependent Sentiment Classification in Political News Articles". "Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume"
- Sharma, Raksha; Somani; Kumar; Bhattacharyya (2017). "Sentiment Intensity Ranking among Adjectives Using Sentiment Bearing Word Embeddings" (PDF). Association for Computational Linguistics: 547–552.
- Kim, S. M.; Hovy, E. H. (2006). "Identifying and Analyzing Judgment Opinions." (PDF). Proceedings of the Human Language Technology / North American Association of Computational Linguistics conference (HLT-NAACL 2006). New York, NY. Archived from the original (PDF) on 2011-06-29.
- Gottschalk, Louis August, and Goldine C. Gleser. “The measurement of psychological states through the content analysis of verbal behavior”. Univ of California Press, 1969.
- Turney, Peter (2002). "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews". Proceedings of the Association for Computational Linguistics. pp. 417–424. arXiv:cs.LG/0212032.
- Pang, Bo; Lee, Lillian; Vaithyanathan, Shivakumar (2002). "Thumbs up? Sentiment Classification using Machine Learning Techniques". Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 79–86.

APPENDIX-A

```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('ggplot')
import nltk

# Read the dataset from 'Reviews.csv'
df = pd.read_csv('Reviews.csv')

# Print the shape of the original dataset
print(df.shape)

# Select first 500 rows for analysis
df = df.head(500)

# Print the shape of the modified dataset
print(df.shape)

# Display the first few rows of the dataset
df.head()

# Plot a bar chart to visualize the distribution of review scores
```

```
ax = df['Score'].value_counts().sort_index() \
    .plot(kind='bar',
          title='Count of Reviews by Stars',
          figsize=(10, 5))
ax.set_xlabel('Review Stars')
plt.show()

# Download the VADER Lexicon for sentiment analysis
nltk.download('vader_lexicon')

# Initialize the SentimentIntensityAnalyzer
from nltk.sentiment import SentimentIntensityAnalyzer
from tqdm.notebook import tqdm
sia = SentimentIntensityAnalyzer()

# Perform sentiment analysis for the entire dataset
res = {}
for i, row in tqdm(df.iterrows(), total=len(df)):
    text = row['Text']
    myid = row['Id']
    res[myid] = sia.polarity_scores(text)

# Display the sentiment analysis results
res
# Create a DataFrame from the sentiment analysis results
vaders = pd.DataFrame(res).T
```

```
# Reset the index and rename the columns
vaders = vaders.reset_index().rename(columns={'index': 'Id'})

# Merge the sentiment analysis DataFrame with the original dataset
vaders = vaders.merge(df, how='left')

# Display the first few rows of the merged DataFrame
vaders.head()

# Create subplots for sentiment analysis visualization
fig, axs = plt.subplots(1, 4, figsize=(12, 3))
sns.barplot(data=vaders, x='Score', y='pos', ax=axs[0])
sns.barplot(data=vaders, x='Score', y='pos', ax=axs[1])
sns.barplot(data=vaders, x='Score', y='neu', ax=axs[2])
sns.barplot(data=vaders, x='Score', y='neg', ax=axs[3])
axs[0].set_title('compound')
axs[1].set_title('Positive')
axs[2].set_title('Neutral')
axs[3].set_title('Negative')
plt.tight_layout()
plt.show()

# Create sentiment labels based on compound score
pos_neg1 = []
for i in range(len(vaders['compound'])):
    if vaders['compound'][i] >= 0:
        pos_neg1.append(1)  # 1 for positive sentiment
    else:
        pos_neg1.append(0)  # 0 for negative sentiment

# Add sentiment labels to the DataFrame
vaders['sentiment_label'] = pos_neg1

# Download NLTK stopwords for text processing
nltk.download('stopwords')

# Import necessary libraries for word cloud generation
from nltk.corpus import stopwords
from wordcloud import WordCloud

# Consolidate text for word cloud generation
consolidated = ' '.join(
    word for word in vaders['Text'][vaders['sentiment_label'] ==
1].astype(str))

# Generate and display the word cloud
wordCloud = WordCloud(width=1600, height=800,
                                          random_state=21, max_font_size=110)
plt.figure(figsize=(15, 10))
plt.imshow(wordCloud.generate(consolidated), interpolation='bilinear')
plt.axis('off')
plt.show()

# Import necessary libraries for further data processing and modeling
```

```
   import re

   import seaborn as sns

   from sklearn.feature_extraction.text import TfidfVectorizer

   import matplotlib.pyplot as plt


   # Initialize the TF-IDF Vectorizer

   cv = TfidfVectorizer(max_features=2500)


   # Transform the text data into numerical features

   X = cv.fit_transform(vaders['Text']).toarray()


   # Split the dataset into training and testing sets

   from sklearn.model_selection import train_test_split

   X_train, X_test, y_train, y_test = train_test_split(X,
vaders['sentiment_label'],

   test_size=0.33,
        stratify=vaders['sentiment_label'],
        random_state = 42)


   # Import necessary libraries for model evaluation

   from sklearn.metrics import accuracy_score

   from sklearn.tree import DecisionTreeClassifier


   # Initialize and train the Decision Tree Classifier

   model = DecisionTreeClassifier(random_state=0)

   model.fit(X_train, y_train)


   # Test the model on the training set and print accuracy
```

```
pred = model.predict(X_train)
print(accuracy_score(y_train, pred))

# Import necessary libraries for confusion matrix visualization
from sklearn import metrics
cm = confusion_matrix(y_train, pred)

# Display the confusion matrix
cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix=cm,

    display_labels=[False, True])
cm_display.plot()
plt.show()
```