

A Project Report
On
Sentiment Analysis and Opinion Mining on E-commerce Reviews
Submitted in partial fulfillment of the
requirement for the award of the degree of
BACHELOR OF COMPUTER APPLICATION



Session 2023-24
in
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

By
Shashank Singh
21SCSE1430015
Vaibhav Singh
21SCSE1430024
Satyam Anand
21SCSE1430027

Under the guidance of
Ms. Nitin Sondhi
Assistant Professor

SCHOOL OF COMPUTER APPLICATION AND TECHNOLOGY
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
April, 2024



SCHOOL OF COMPUTER APPLICATION AND
TECHNOLOGY
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled “Sentiment Analysis and Opinion Mining on E-commerce Reviews” in partial fulfillment of the requirements for the award of the BCA (Bachelor of Computer Application) submitted in the School of Computer Application and Technology of Galgotias University, Greater Noida, is an original work carried out during the period of September 2023 to April 2024, under the supervision of Mr. Nitin Sondhi Assistant Professor, Department of School of Computer Application and Technology, Galgotias University, Greater Noida.

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Shashank Singh, 21SCSE1430015

Vaibhav Singh, 21SCSE1430024

Satyam Anand, 21SCSE1430027

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mr. Nitin Sondhi

Assistant Professor

CERTIFICATE

This is to certify that Project Report entitled “Sentiment Analysis and Opinion Mining on E-commerce Reviews” which is submitted by Shashank Singh (21SCSE1430015), Vaibhav Singh(21SCSE1430024), Satyam Anand (21SCSE1430027) in partial fulfillment of the requirement for the award of degree BCA. in Department of SCAT of School of Computer Application and Technology, Galgotias University, Greater Noida, India is a record of the candidate own work carried out by him/them under my supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Signature of Examiner(s)

Signature of Supervisor(s)

Date: April, 2024

Place: Greater Noida

Abstract

Nowadays, the world is becoming more digital and customers are relying on online products which helps them to simplify their living. Therefore, the reviews provided on a product becomes an important aspect to attract and maintain customers and build a novel strategy to acquire higher position in the market. The purpose of this study is to investigate the different processes and techniques used in gathering requirements, designing, implementing and testing the reviews provided on a particular product to gain insights on customer experiences. It can be attained by using sentiment analysis and opinion mining which is defined is the computational study of people's opinions, sentiments, attitudes, and emotions expressed in written language. It is one of the most active research areas in natural language processing and text mining in recent years...

This study helps to analyse the sentiments and the opinions of people based on their reviews on a product or service of E-commerce. It uses a dataset of the Ecommerce reviews and then built a model to predict the sentiment of the comment given the comment declaration by using Python and deep learning algorithms.

Table of Contents

TITLE		PAGE NO.
Candidate Declaration		1
Certificate		2
Abstract		3
List of Figures		6
Acronyms		5
Chapter - 1	Introduction	
	1.1	Introduction
	1.1.1	Sentiment Analysis
	1.1.2	Machine learning
	1.1.5	Python
	1.3	Numpy, matplotlib, seaborn
Chapter - 2	Literature Survey	
	2.1	Multimedia semantic analysis
	2.2	Dictionary- based analysis
	2.3	Movie-Reviews
	2.4	Language-based

Chapter - 3	Functionality/Working		
	3.1	Introduction	18
	3.2	System Architecture	21
	3.3	Required Tools	22
	3.4	dataset	23
	3.5	Methodology	24
		3.5.1	Data collection
		3.5.2	Data Preprocessing
		3.5.3	Pre-processing of Data
		3.5.4	correlation
		3.5.5	Word Cloud
	3.6	Algorithm	
		3.6.1	Support Vector Machine
		3.6.2	SGD classifier
Chapter-4	Implementation		40
Chapter - 5	Result & Discussion		
	5.1	Result	49
		5.1.1	Comparative study

List of figures

Figure name	Page No.
Use case diagram	23
Algorithm	23
No. of stars given in the reviews dataset	27
correlation on anonymous/named users	28
Wordcloud	29
Accuracy chart on train data	48
Accuracy chart on test data	48
Confusion matrix	49
Comparative study	50

Acronyms

B.Tech.	Bachelor of Technology
M.Tech.	Master of Technology
BCA	Bachelor of Computer Applications
MCA	Master of Computer Applications
B.Sc. (CS)	Bachelor of Science in Computer Science
M.Sc. (CS)	Master of Science in Computer Science
SCSE	School of Computing Science and Engineering

CHAPTER-1

Introduction

As the commercial sites are becoming popular and people are trading different products from different e-commerce sites. Therefore, reviewing products before buying is a common aspect to come to a conclusion for the purchase. The business teams analyse the product and service reviews over the Internet to strengthen their strategies and make a profitable and novel decisions to enhance their business. Similarly, it is easy for governments to get public feedback on their policies and learn about important events in other countries. However, with the proliferation of various websites, detecting and monitoring opinion online and extracting information from them remains a difficult task. The average human reader would have trouble identifying relevant websites, extracting and summarizing comments from them. Therefore, an automated sentiment analysis system is needed.

Sentiment analysis allows large-scale processing of data in an efficient and cost-effective manner. Its popularity is mainly due to two reasons. First, it has a wide range of applications because opinions are central to almost all human activities and are key influencers of our behavior. It can be used to rank people's attitudes on topics, comments from Facebook posts, tweets, product reviews, political agendas, review sites, and more. An important feature of opinions is that they are subjective, unlike factual information. Sentiment analysis involves analyzing the opinions of a population group to obtain some sort of opinion insights and apply that information to have profitable responses. Opinion information is very important for businesses and manufacturers. They often want to know in time what consumers and the public think of their products and services. However, it is not realistic to manually read every post on the website and extract useful viewpoint information from it. If you do it manually, there is too much data.

Sentiment analysis methodology is used which basically relies on machine learning and Natural language processing and our major focus will be performing sentiment analysis methodology

on Amazon reviews using python. In this, the algorithm recognizes positive and negative feedback, visualizes the content via his word cloud, predicts

ratings from naive Bayesian evaluation and logistic regression models, and then compares the accuracy of various machine learning algorithms.

In our model, both active and manual approach is taken to label the dataset. Linear SVC Algorithm and SGD classifiers are used to provide accurate results without loss of valuable information. From the processed dataset, these algorithms worked upon training and test datasets separately to distinguish the gap between the accuracies.

1.2 Machine Learning

Machine learning is a subfield of artificial intelligence (AI) that involves building algorithms that can learn from data and improve their performance over time. The goal of machine learning is to enable computers to make predictions, decisions, and identify patterns without being explicitly programmed.

Machine learning algorithms typically rely on large datasets to train models that can make accurate predictions or decisions on new, unseen data. The process of training a machine learning model involves selecting an appropriate algorithm, choosing relevant features, and using optimization techniques to adjust the model's parameters and improve its performance.

There are three main types of machine learning: supervised learning, unsupervised learning, and reinforcement learning.

Supervised learning: In supervised learning, the algorithm is trained on a labeled dataset, where each input data point is associated with a corresponding output label or value. The goal of the algorithm is to learn a mapping function that can predict the output for new, unseen inputs.

Unsupervised learning: In unsupervised learning, the algorithm is trained on an unlabeled

dataset, where there are no predefined output labels or values. The goal of the algorithm is to identify patterns and relationships in the data, such as clustering similar data points together or identifying important features.

Reinforcement learning: In reinforcement learning, the algorithm learns through trial-and-error interactions with an environment. The algorithm receives feedback in the form of rewards or punishments based on its actions, and it adjusts its behavior to maximize the expected reward.

Machine learning is used in a wide range of applications, such as image and speech recognition, natural language processing, recommender systems, fraud detection, and predictive modeling, among others

1.3 PYTHON

Python is a high-level programming language that is widely used for various applications, including web development, scientific computing, data analysis, artificial intelligence, and more. Python was first released in 1991 and has since become one of the most popular programming languages in the world.

Some of the key features of Python include:

- 1.Easy to learn: Python has a simple and intuitive syntax that makes it easy to learn, even for beginners.
- 2.Large standard library: Python comes with a large standard library that provides a wide range of modules and functions for various tasks, such as web development, data analysis, and scientific computing.

3.Platform-independent: Python can run on different operating systems, including Windows, Linux, and macOS, making it a versatile language.

4.Interpreted language: Python is an interpreted language, which means that code can be executed directly without the need for compilation, making it easier to test and debug code.

5.Object-oriented: Python supports object-oriented programming, which allows developers to write reusable code and create complex data structures.

6.Dynamic typing: Python is dynamically typed, which means that data types are determined at runtime rather than being declared in advance.

Python has a large and active community of developers who contribute to its development and create open-source libraries and tools that extend its functionality. Some of the popular libraries in Python include NumPy for scientific computing, Pandas for data analysis, Django for web development, and TensorFlow for machine learning.

PANDAS

Pandas is a Python library for data manipulation and analysis. It provides data structures for efficiently storing and processing large datasets, and a wide range of tools for working with data. Here are some of the key features and functionalities of Pandas:

Data structures: Pandas provides two main data structures: Series and DataFrame. A Series is a one-dimensional array-like object that can hold any data type, while a DataFrame is a two-dimensional table-like data structure that can hold data of different types in columns.

Data manipulation: Pandas provides a wide range of tools for data manipulation, including filtering, sorting, merging, grouping, pivoting, reshaping, and transforming data.

Data cleaning: Pandas provides tools for data cleaning, such as handling missing values, transforming data types, and removing duplicates.

Data visualization: Pandas provides integration with Matplotlib, a popular data visualization library, for creating plots and visualizations of data.

Input/output: Pandas provides tools for reading and writing data in various formats, including CSV, Excel, SQL, and JSON.

Integration with other libraries: Pandas can be integrated with other Python libraries, such as NumPy, Scikit-learn, and TensorFlow, for data analysis and machine learning tasks.

Overall, Pandas is a powerful tool for data analysis and manipulation in Python, and is widely used in data science and machine learning workflows. Its flexible data structures and rich functionality make it a valuable tool for working with large datasets and performing complex data transformations.

MATPLOTLIB

Matplotlib is a Python library for creating data visualizations. It provides a wide range of tools for creating different types of plots and charts, such as line plots, scatter plots, bar plots, histograms, and heatmaps. Here are some key features and functionalities of Matplotlib:

Plotting functions: Matplotlib provides a variety of plotting functions that can be used to create different types of plots. These functions take data and formatting parameters as input and output a plot.

Customizability: Matplotlib allows for a high degree of customization of plots, including axis labels, legends, titles, color schemes, and plot styles.

Integration with Pandas: Matplotlib integrates with Pandas, a popular data manipulation library, allowing for easy creation of plots from Pandas DataFrames.

Compatibility with Jupyter notebooks: Matplotlib is compatible with Jupyter notebooks, a popular environment for data analysis and exploration.

Support for multiple output formats: Matplotlib can output plots in a variety of formats, including PNG, PDF, SVG, and EPS

PLOTLY

Plotly is a popular open-source library for creating interactive visualizations in Python. With Plotly, you can create a wide variety of charts and graphs, from simple scatter plots to complex 3D visualizations.

With the help of Plotly library we will take a look at the variable “ Score ” to see if majority of the customer ratings are positive or negative.

NUMPY

NumPy (short for Numerical Python) is a Python library that provides support for large, multi-dimensional arrays and matrices, as well as a large collection of mathematical functions to operate on these arrays. NumPy is a fundamental library for scientific computing with Python.

Here are some of the main features and benefits of using NumPy:

Multidimensional array support: NumPy provides support for multidimensional arrays and matrices, which allows for efficient computation and manipulation of large datasets.

Mathematical functions: NumPy provides a large collection of mathematical functions that can be applied to arrays and matrices, such as linear algebra functions, Fourier transform, random number generation, and statistical analysis.

Broadcasting: NumPy supports broadcasting, which allows for efficient computation on arrays of different shapes and sizes.

Integration with other libraries: NumPy integrates well with other scientific computing libraries, such as SciPy, Pandas, and Matplotlib, providing a powerful toolkit for scientific computing with Python.

SEABORN

Seaborn is a Python data visualization library built on top of matplotlib that provides a high-level interface for creating informative and attractive statistical graphics. It is designed to work well with Pandas and NumPy data structures, and provides a variety of visualization techniques to

explore and understand relationships within datasets. Here are some of the main features and benefits of using Seaborn:

Easy to use: Seaborn provides a simple and intuitive interface for creating complex visualizations with minimal code.

Beautiful default styles: Seaborn provides aesthetically pleasing default styles for visualizations, making it easy to create attractive graphics without much customization.

CHAPTER-2

Literature Survey

Adams W, Iyengar G, Lin CY, Naphade MR, Neti C, Nock HJ, Smith JR presented a learning-based approach for semantic indexing of multimedia content using cues derived from audio, image, and text features. We approach this problem by developing a set of pre-defined lexicons statistical models. New concepts are then mapped using the concepts in the glossary. We use multiple modalities, namely audio, video and text capabilities, to achieve robust concept recognition.

In another study of the subjectivity of English word sense shows that subjective aspects of an entity are traits that are better delineated at the semantic level than at the traditional word level. In this article, we examine whether sensibility coordinated across languages consistently exhibits this property, and if so, how to exploit this property automatically. First, we perform a manual annotation survey to assess whether subjective properties of sensations can be reliably transmitted across language boundaries.

In a study called comprehensive study of sentimental analysis, it is found that sentiment analysis is the task of extracting and analyzing people's opinions, feelings, attitudes, perceptions, etc. about various entities such as subjects, products, and services. The rapid development of Internet-based applications such as websites, social networks, and blogs has allowed people to generate vast amounts of opinions and reviews about products, services, and everyday activities.

Bopin et al. [5] examined the effectiveness of classifying documents by their overall sentiment using machine learning techniques. Experiments have shown that machine learning techniques outperform human-crafted foundations in sentiment analysis of movie review data. The experimental setup consists of a movie review corpus containing 700 randomly selected positive mood reviews and 700 negative mood reviews.

Classification, and Support Vector Machines were used. Conclusions of Pan et al. Machine learning techniques outperform human baselines for sentiment classification. On the other hand, the accuracy achieved with sentiment classification is much lower than with topic-based classification.

Xiaowen Ding et al. [15] proposed a holistic dictionary-based approach that uses external cues and linguistic conventions of natural language expressions to determine the semantic orientation of opinions. The advantage of this approach is that it can easily handle context-sensitive opinion terms. The algorithm used uses the language's patterns to handle special words and phrases. Based on this technique, the researcher built a system called his Opinion Observer. Experimenting with the product reviews dataset is very effective. It has been shown to efficiently handle multiple conflicting sentence terms. This system shows superior performance at compared to existing methods.

Alekh Agarwal et al., [8] proposed a machine learning method that incorporates linguistic knowledge gathered through synonym graphs for effective opinion classification. This approach demonstrates the extent to which document relationships affect sentiment analysis. This is achieved by using graph-cutting techniques and opinion words derived from Wordnet's synonym graph. The proposed approach also improves prediction accuracy in classification tasks. Experiments with this system have produced results with greater than 90% accuracy, reduced processing time, and minimal differences in final accuracy. The expected usefulness of the ratings is used for ranking, which is also based on the expected effect on sales. The proposed method identifies reviews with the greatest impact. For feature-oriented products, reviews that verify that the information in the product description is used, as well as subjective reviews are useful. Proposed method uses econometric analysis with text mining techniques and subjective analysis.

CHAPTER 3

FUNCTIONALITY

SENTIMENT ANALYSIS

Sentiment analysis is a computational process used to determine the emotional tone or attitude expressed in a piece of text, such as a review, social media post, or news article. The purpose of sentiment analysis is to automatically categorize the sentiment of the text as positive, negative, or neutral.

Sentiment analysis is typically achieved using natural language processing (NLP) techniques, which involve parsing the text and analyzing its linguistic features, such as the use of certain words, phrases, or emoticons, as well as the context in which they are used.

Sentiment analysis can be useful for a variety of applications, such as monitoring customer feedback, tracking brand reputation, identifying emerging trends, or predicting market behavior. It can also be combined with other NLP techniques, such as named entity recognition, topic modeling, or summarization, to gain deeper insights into the underlying themes and issues in the text.

ADVANTAGES

Sentiment analysis has several advantages that make it a valuable tool for businesses, researchers, and individuals who work with text data. Some of the key advantages of sentiment analysis include:

Understanding customer sentiment: Sentiment analysis can be used to analyze customer feedback, such as reviews, social media posts, and customer service interactions, to gain insights into customer sentiment and preferences.

Monitoring brand reputation: Sentiment analysis can be used to track brand reputation and public perception by analyzing online mentions of a brand or product.

Identifying emerging trends: Sentiment analysis can be used to identify emerging trends or topics by analyzing patterns in online discussions or social media activity.

Improving customer service: Sentiment analysis can be used to identify areas where customer service can be improved by analyzing customer feedback and identifying common issues.

Quantifying subjective data: Sentiment analysis can be used to quantify subjective data, such as opinions, attitudes, and emotions, making it easier to analyze and compare across different sources. Overall, sentiment analysis provides a powerful tool for gaining insights from text data, enabling businesses and researchers to make data-driven decisions and better understand their audiences.

LIMITATIONS

While sentiment analysis can be a powerful tool for analyzing text data, it also has several limitations. Some of the key limitations include:

Contextual ambiguity: Sentiment analysis models can struggle with detecting sarcasm, irony, or other forms of subtle language use, which can lead to misinterpretations of the sentiment of a text.

Domain-specific knowledge: Sentiment analysis models are often trained on general language patterns, but they may not be able to accurately detect sentiment in specialized domains, such as technical jargon or slang.

Cultural and linguistic variations: Sentiment analysis models can perform differently across different cultures and languages, which can lead to inaccurate predictions in cross-cultural or multilingual settings.

Data bias: Sentiment analysis models can be biased based on the training data used to develop them, which can lead to inaccurate predictions or reinforce existing social biases.

Limited emotional range: Sentiment analysis models are typically limited to detecting positive, negative, or neutral sentiment, which can oversimplify the complexity of human emotions and attitudes expressed in text.

Sentiment polarization: Sentiment analysis models may classify sentiment in a binary way, leading to sentiment polarization, where the analysis fails to capture the nuances and complexities of mixed or ambiguous sentiment.

It's important to keep these limitations in mind when using sentiment analysis and to combine it with other methods of analysis to gain a more complete understanding of the text data.

REQUIRED TOOLS

HARDWARE

- Processor- i3
- Hard disk- 4 Gb
- Memory – 2 Gb ram
- Internet connection
- Server

SOFTWARE

VS code

Google colab

DATASET

A dataset is a collection of data that is organized and structured in a specific way. Datasets can take many forms, including tables, spreadsheets, databases, text files, and more. They can be used for a variety of purposes, such as data analysis, machine learning, and statistical modeling.

Datasets typically consist of rows and columns, with each row representing an individual observation or example, and each column representing a variable or feature of that observation. For example, in a dataset of customer information, each row might represent a different customer, and the columns might include information such as age, gender, income, and purchase history.

Datasets can be collected through a variety of means, including manual data entry, web scraping, surveys, and more. They can also be publicly available or privately held, depending on the source and intended use.

In machine learning, datasets are often used for training and testing models. A dataset is typically split into a training set, used to train the model, and a testing set, used to evaluate the model's performance. The quality and size of the dataset can have a significant impact on the performance of a machine learning model, making the selection and preparation of the dataset an important step in the machine learning process.

The dataset used in this study basically consists of tables where few information about the product is given like product name, manufacturer and date of manufacturing.

Apart from that, It consists of reviews like ratings on the product, If they would recommend the product further as a true/false criteria, review statements, source URLs and reviewers details like name and city.

DESIGN

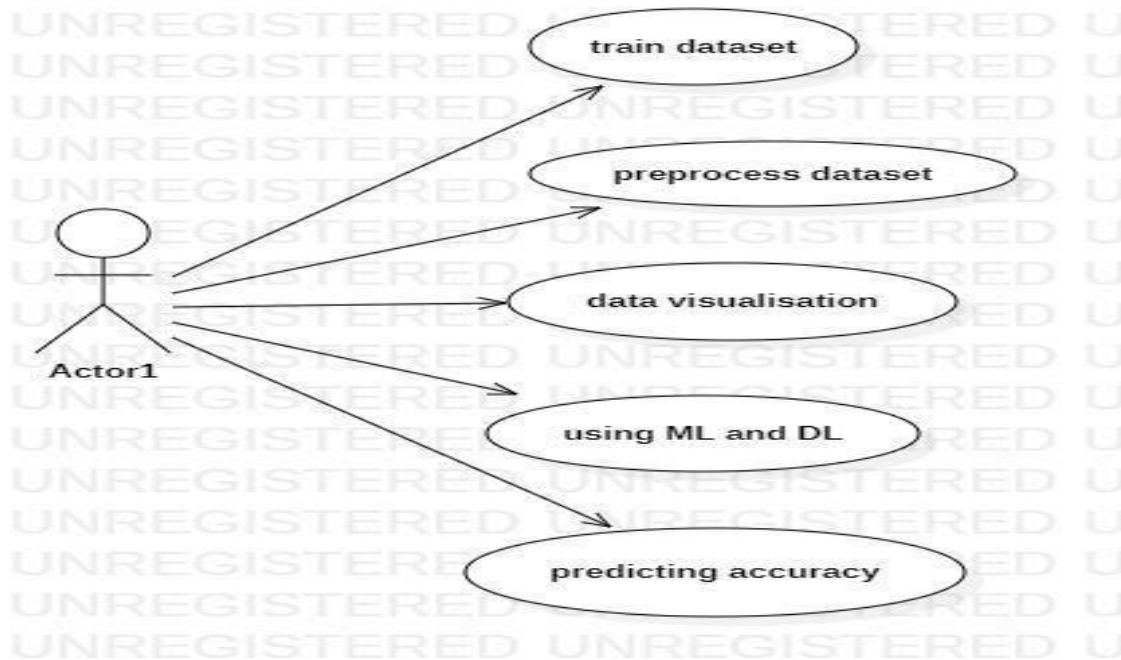


Fig 4: Use Case diagram

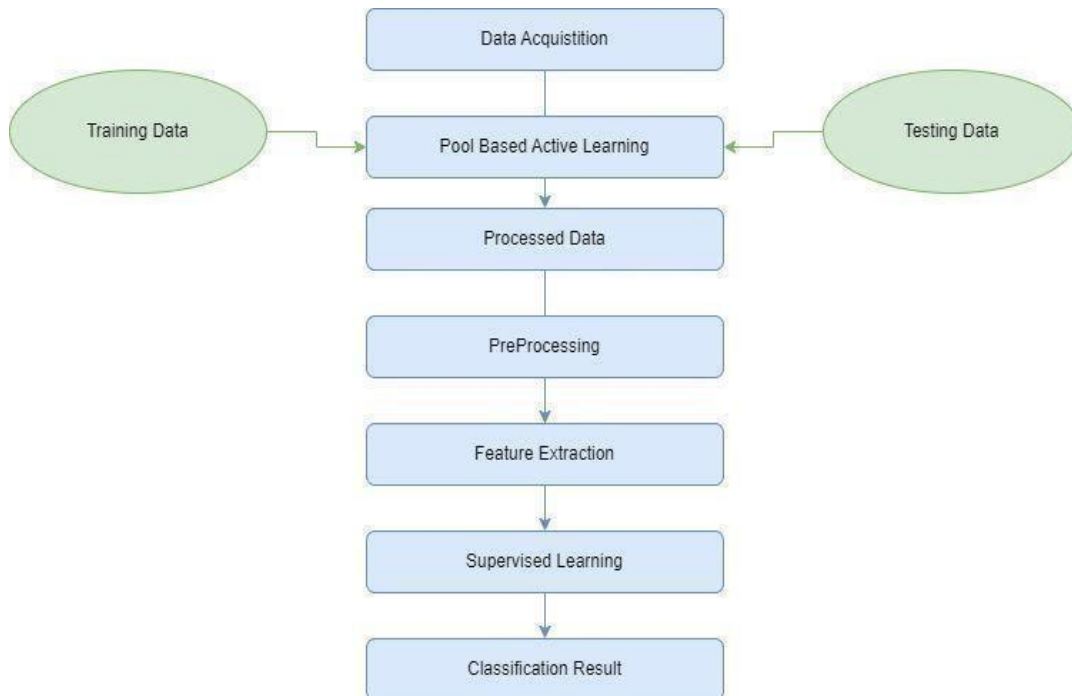


Fig 5: Algorithm diagram

METHODOLOGY

A. DATA COLLECTION

Data collection refers to the process of gathering and collecting data from various sources for analysis or processing. Data collection can be done manually or using automated methods, depending on the type of data, the volume of data, and the resources available.

The data collection process typically involves the following steps:

Defining the research question or problem: The first step in data collection is to clearly define the research question or problem that the data will be used to address. **Identifying the data sources:** Once the research question or problem is defined, the next step is to identify the sources of data that will be used to answer the question or address the problem. These sources can include public databases, surveys, interviews, social media, and other sources.

Designing the data collection method: Depending on the type of data, different methods may be used to collect the data. For example, surveys may be conducted to gather information from a large number of people, while interviews may be used to gather more in-depth information from a smaller group of people.

Collecting the data: Once the data collection method is designed, the data can be collected. This may involve sending out surveys, conducting interviews, or using automated tools to collect data from social media or other online sources.

Cleaning and organizing the data: After the data is collected, it needs to be cleaned and organized to ensure that it is accurate, complete, and ready for analysis.

Analyzing the data: Once the data is cleaned and organized, it can be analyzed to extract insights and draw conclusions.

Data collection is an important step in the data analysis process, as the quality of the data collected can impact the accuracy and reliability of the insights that are generated. It is important to carefully plan and execute the data collection process to ensure that the data collected is relevant, accurate, and complete.

The major source of data for this study is the E-commerce product reviews sheet online.

This can be acquired from kaggle where the dataset based on reviews for different products is present. Users rate different products according to their accountability and satisfaction. It can also provide the range of issue based on durability, sizes, prices etc.

B. DATA PREPARATION

Data preparation refers to the process of cleaning, transforming, and organizing raw data to make it suitable for analysis. Data preparation is a critical step in the data analysis process, as the quality of the data and the way it is organized can have a significant impact on the accuracy and reliability of the insights that are generated.

The data preparation process typically involves the following steps:

Data cleaning: This involves identifying and correcting errors or inconsistencies in the data, such as missing values, outliers, or incorrect data types.

Data integration: If the data is coming from multiple sources, it may need to be integrated into a single dataset to facilitate analysis.

Data transformation: This involves converting the data into a format that is suitable for analysis. For example, categorical data may need to be converted into numerical data, or data may need to be standardized or normalized.

Data reduction: If the dataset is very large, it may be necessary to reduce the size of the dataset by selecting a subset of the data or summarizing the data using statistical methods.

Data sampling: If the dataset is too large to analyze in its entirety, a sample of the data may be selected for analysis.

Data formatting: This involves organizing the data in a format that is suitable for analysis. For example, data may need to be arranged in rows and columns or aggregated by a certain variable.

Data validation: This involves checking the quality and accuracy of the data to ensure that it is suitable for analysis.

By preparing the data properly, analysts can improve the accuracy and reliability of the insights generated from the data. Good data preparation practices can also help to save time and resources by minimizing the need for data cleaning or re-analysis later on.

The dataset should be accurate to perform machine learning algorithms in order to produce high quality results. For serving this purpose, the datasets needs to be cleaned and prepared to work accordingly.

LEXICAL- It helps in distributing the labelled words according to the sentiments using a vocabulary based technique which measures the mild and extreme nature of words.

STRUCTURAL-It helps in breaking up the reviews into simpler arrays of necessary words,normalizing the data and eliminating the unnecessary occurrences.

TEXT REPRESENTATION- It helps in finding the frequency of common words used in the reviews and making a word cloud with those words to fetch it to the algorithm.

C. CORRELATION

Correlation refers to the statistical relationship between two or more variables. When two variables are correlated, changes in one variable are associated with changes in the other variable. Correlation can be measured using a statistical metric known as a correlation coefficient.

There are two main types of correlation: positive correlation and negative correlation. Positive correlation occurs when two variables increase or decrease together, while negative correlation occurs when one variable increases while the other variable decreases. The strength of a correlation can be measured using a correlation coefficient, which ranges from -1 to +1. A correlation coefficient of -1 indicates a perfect negative correlation, a correlation coefficient of +1 indicates a perfect positive correlation, and a correlation coefficient of 0 indicates no correlation.

Correlation can be useful in many areas of analysis, such as finance, marketing, and social sciences. For example, in finance, analysts may use correlation to identify relationships between stock prices or economic indicators. In marketing, correlation can be used to identify patterns in customer behavior or preferences. In social sciences, correlation can be used to identify relationships between different variables, such as income and education.

It is important to note that correlation does not necessarily imply causation. Just because two variables are correlated does not mean that one variable causes the other. Other factors, such as confounding variables, may be influencing the relationship between the variables. Therefore, it is important to use caution when interpreting correlations and to consider other factors that may be influencing the relationship between the variables.

After the filtration and preparation of data, the processed data was correlated with the frequency of stars given as a review for a particular product.

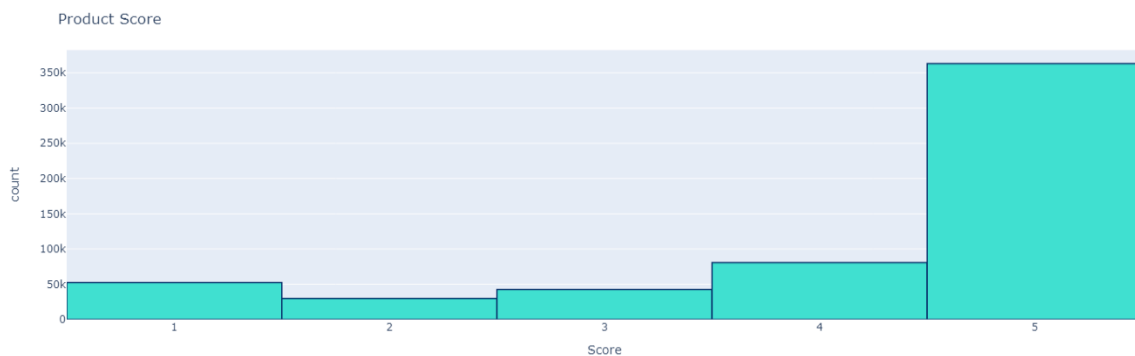


Fig 1: No. of stars given in the reviews dataset

It is also correlated by the anonymous and named reviewers to get an insight of true and false values.

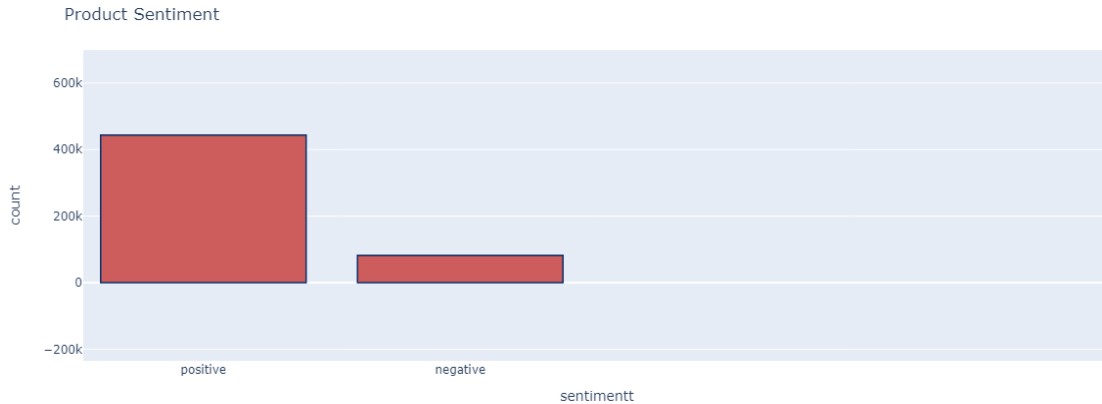


Fig 2: correlation on anonymous/named users

D. WORD CLOUD

A word cloud is a visual representation of text data, where the size of each word represents its frequency or importance in the text. In a word cloud, the most frequently used words in the text are displayed in larger font sizes, while less frequently used words are displayed in smaller font sizes.

Word clouds are often used to provide a quick and easy way to visualize patterns or trends in text data. They can be generated using various tools or software, such as Wordle, TagCrowd, or Python libraries like wordcloud or matplotlib.

Word clouds can be useful for a variety of applications, such as:

- Identifying the most common themes or topics in a body of text

- Visualizing customer feedback or reviews to identify common issues or areas of satisfaction

- Summarizing the key points of a document or presentation

- Analyzing social media conversations or news articles to identify trending topics or sentiments

Word clouds should be used with caution, however, as they can be influenced by factors such as the size of the text data, the method used to generate the word cloud, and the words chosen for inclusion in the word cloud.

It is important to consider the context and content of the text data when interpreting a word cloud, and to use additional analysis methods to gain a deeper understanding of the patterns and trends in the text.

each class. SVM can also handle non-linearly separable data by transforming the feature space into a higher dimensional space using a kernel function.

ADVANTAGES

Support Vector Machines (SVM) is a popular supervised learning algorithm used for classification and regression analysis. SVM works by finding a hyperplane (a boundary that separates two classes) in the feature space that maximizes the margin between the two classes. The margin is defined as the distance between the hyperplane and the closest data points from each class. SVM can also handle non-linearly separable data by transforming the feature space into a higher dimensional space using a kernel function.

Support Vector Machines (SVM) is a popular supervised learning algorithm used for classification and regression analysis. SVM has several advantages, including:

Effective for high-dimensional data: SVM can handle a large number of features, making it effective for high-dimensional datasets.

Robust to outliers: SVM is robust to outliers as it focuses on the data points that lie on the margin or near the margin.

Good generalization performance: SVM has good generalization performance, which means that it can accurately classify new, unseen data.

Versatility: SVM can be used for both classification and regression tasks, and it can handle non-linearly separable data using kernel functions.

Less prone to overfitting: SVM is less prone to overfitting than other algorithms, such as decision trees, as it uses the margin to find the best boundary rather than trying to fit the data exactly.

Can handle large datasets: SVM is able to handle large datasets efficiently due to its ability to work with sparse data.

Tunable: SVM provides several parameters that can be tuned for better performance, including the choice of kernel function, the regularization parameter, and the kernel coefficient.

Widely used: SVM is a widely used algorithm and has been applied successfully in various applications, such as image classification, text classification, and bioinformatics.

Overall, SVM is a powerful and versatile algorithm that can be useful in a variety of applications.

LIMITATIONS

Support Vector Machines (SVM) is a popular supervised learning algorithm used for classification and regression analysis. While SVM has several advantages, it also has some limitations, including:

Sensitivity to parameter settings: SVM's performance can be sensitive to the choice of hyperparameters, such as the regularization parameter and kernel function. Selecting the wrong parameters can lead to poor performance.

Computationally intensive: SVM can be computationally intensive for large datasets, especially when using non-linear kernels. This can lead to longer training times and higher computational costs.

Difficult to interpret: SVM does not provide easily interpretable models, making it difficult to understand how the algorithm is making its predictions.

Limited applicability to multi-class problems: SVM was originally designed for binary classification problems and may not be as effective for multi-class problems. It requires the use of one-versus-one or one-versus-all techniques to extend it to multi-class problems.

Not suited for non-numerical data: SVM is designed to work with numerical data and cannot handle non-numerical data without preprocessing.

Can be affected by imbalanced data: SVM can be affected by imbalanced data, where one class has significantly more samples than the other, leading to biased predictions.

Can be affected by noisy data: SVM can be affected by noisy data, as outliers can significantly impact the location and orientation of the decision boundary.

Overall, while SVM is a powerful and widely used algorithm, it may not always be the best choice for every problem. It is important to carefully consider the limitations and potential challenges of SVM when deciding whether to use it for a particular problem.

HYPERPLANE

In machine learning, a hyperplane is a geometric object that separates data into different classes or groups. In a two-dimensional space, a hyperplane is a line that separates the space into two regions, and in a three-dimensional space, it is a plane that separates the space into two regions.

In the context of SVM (Support Vector Machines), a hyperplane is used as a decision boundary to separate data points into different classes. The goal of SVM is to find the hyperplane that maximizes the margin between the two classes, which is the distance between the hyperplane and the nearest data points from each class.

When the data is linearly separable, there is a unique hyperplane that perfectly separates the data into two classes. However, when the data is not linearly separable, a hyperplane can still be used by mapping the data into a higher-dimensional feature space, where a linear boundary may exist.

In summary, a hyperplane is a fundamental concept in machine learning and is often used as a decision boundary to separate data points into different classes. SVM is one algorithm that utilizes hyperplanes to find an optimal decision boundary that maximizes the margin between the classes.

If the hyperplane is linear then, SVC is used. In this, Margins are used which separates the line to the closest points.

Good Margin- All the points have same distance and maximum to hyperplane

Bad Margin- close to -1 or +1 vectors of hyperplane.

Soft Margin-if the points are not linearly separable, some points are allowed even if they are on the incorrect side of hyperplane.

Hard Margin- If data is separated linearly, then two parallel lines are considered and largest distance between them.

SGD CLASSIFIER

The SGD (Stochastic Gradient Descent) classifier is a linear classification algorithm that is widely used in machine learning. It is particularly useful for large datasets since it can learn from data incrementally, which makes it efficient for online learning.

The SGD classifier works by minimizing the loss function, which measures the difference between the predicted class and the true class label. The algorithm updates the model parameters using the gradient of the loss function, which is calculated for each training example.

One of the advantages of the SGD classifier is its ability to handle large datasets. This is because it updates the model parameters after processing each training example, which makes it more efficient than other batch learning algorithms that require the entire dataset to be processed before updating the model parameters.

Another advantage of the SGD classifier is its ability to handle sparse data, which is common in natural language processing applications. It achieves this by using sparse representations of the input data and computing the gradient only for the non-zero elements.

The SGD classifier is also flexible in terms of the loss function used for optimization. It can be used with a variety of loss functions, including logistic regression, linear regression, and support vector machines.

However, the SGD classifier can be sensitive to the choice of hyperparameters, such as the learning rate and regularization term, which can affect its performance. Additionally, the stochastic nature of the algorithm can make it prone to fluctuations in the optimization process, which can result in slower convergence and reduced accuracy.

Overall, the SGD classifier is a powerful algorithm that can be useful in a variety of applications, particularly for large datasets and sparse data. However, it requires careful tuning of hyperparameters to achieve optimal performance.

It finds the parameters that reduces the cost functions and predict the gap between the points. It calculates the loss of single point on the hyperplane that can result in the loss of accuracy.

It is used with different machine learning algorithms to predict, recall and find the F1 score of the dataset.

ADVANTAGES

The SGD (Stochastic Gradient Descent) algorithm has several advantages, including:

Efficiency: The SGD algorithm is more efficient than batch gradient descent because it updates the model parameters using only one sample or a small batch of samples at a time. This makes it more suitable for large datasets and online learning.

Flexibility: The SGD algorithm can be used with a variety of loss functions and is not limited to a specific type of model. This makes it a versatile algorithm that can be used for different types of problems, including classification, regression, and neural networks.

Suitable for non-convex optimization: The SGD algorithm can be used for non-convex optimization problems, which cannot be solved using batch gradient descent. It can find local minima in the loss function, which can be useful for problems with multiple solutions.

Memory efficiency: The SGD algorithm requires less memory than batch gradient descent because it only stores one sample or a small batch of samples at a time. This makes it more suitable for problems with limited memory resources.

Adaptivity: The SGD algorithm can adapt to changes in the data distribution over time because it updates the model parameters after each sample or batch of samples. This makes it suitable for online learning and real-time applications.

Handles noisy data: The SGD algorithm can handle noisy data and outliers because it updates the model parameters after each sample or batch of samples. This means that noisy samples are eventually "forgotten" as the model updates itself.

Overall, the SGD algorithm is a powerful optimization algorithm that has several advantages, including efficiency, flexibility, and memory efficiency. It is particularly useful for large datasets and real-time applications, and it can handle noisy data and non-convex optimization problems.

LIMITATIONS

Despite its advantages, the SGD (Stochastic Gradient Descent) algorithm also has some limitations. Here are some of them:

Sensitivity to learning rate: The performance of the SGD algorithm can be sensitive to the choice of learning rate. A learning rate that is too high can cause the algorithm to diverge, while a learning rate that is too low can cause slow convergence.

Limited convergence: The SGD algorithm does not always converge to the global minimum of the loss function, especially in non-convex optimization problems. It may converge to a local minimum, which may not be the optimal solution.

Inability to handle non-differentiable functions: The SGD algorithm requires that the loss function be differentiable, which means it cannot be used with non-differentiable functions.

May require more iterations: Because the SGD algorithm updates the model parameters after each sample or batch of samples, it may require more iterations to converge compared to batch gradient descent.

Prone to noise: The SGD algorithm is more prone to noise and outliers than batch gradient descent, especially when the batch size is small. This is because the gradient computed on a small batch of samples may not be representative of the true gradient.

Hyperparameter tuning: The performance of the SGD algorithm can be sensitive to the choice of hyperparameters, such as the learning rate, batch size, and regularization term. These hyperparameters need to be tuned carefully to achieve optimal performance.

Overall, the SGD algorithm is a powerful optimization algorithm that has several limitations. It requires careful tuning of hyperparameters and may not always converge to the global minimum of the loss function. It is also sensitive to noise and may require more iterations to converge compared to batch gradient descent.

SGD THEOREM

The SGD (Stochastic Gradient Descent) algorithm is an optimization algorithm used to minimize a loss function in a machine learning model. Here is a mathematical explanation of how the SGD algorithm works:

Suppose we have a set of training data, consisting of input features X and corresponding target values y . We want to find the parameters w of a model that can predict the target values y given the input features X .

The SGD algorithm starts with an initial guess for the model parameters w , and iteratively updates the parameters using the following steps:

Choose a random sample from the training data (or a small batch of samples).

Compute the gradient of the loss function with respect to the model parameters w , evaluated at the chosen sample.

Update the model parameters w by taking a step in the direction of the negative gradient, scaled by a learning rate α .

This process is repeated until the algorithm converges to a set of parameters that minimize the loss function.

The mathematical expression for the SGD algorithm can be written as follows:

$$w := w - \alpha * \text{gradient}(w, x_i, y_i)$$

where w is the current set of model parameters, α is the learning rate, $\text{gradient}(w, x_i, y_i)$ is the gradient of the loss function with respect to the model parameters w , evaluated at the sample (x_i, y_i) , and $:=$ denotes the assignment operator.

The gradient of the loss function can be computed using backpropagation in a neural network, or by taking the partial derivatives of the loss function with respect to the model parameters in a linear regression or logistic regression model.

The SGD algorithm can be used with different types of loss functions, such as mean squared error for regression problems or cross-entropy loss for classification problems. It is a powerful optimization algorithm that is well-suited for large datasets and online learning. However, it requires careful tuning of hyperparameters and may not always converge to the global minimum of the loss function.

Chapter 4 Implementation

TRAINING DATASET

In machine learning, a training dataset is a set of data used to train a machine learning algorithm to make predictions or classifications. The training dataset is a subset of the overall dataset that is used to train the model by adjusting its parameters until it can accurately predict the target variable. The training dataset is usually created by partitioning the overall dataset into a training set and a testing set. The training set is used to fit the model, while the testing set is used to evaluate the model's performance. This process is known as supervised learning, where the algorithm learns from labeled examples

The training dataset typically includes input features, or independent variables, and the corresponding output, or dependent variable. The goal is to find a model that can predict the output variable given the input features.

It's important to note that the training dataset should be representative of the overall dataset and should have enough variability to ensure that the model is robust and can generalize well to new data. If the training dataset is too small or biased, the model may overfit or underfit the data, leading to poor performance on new, unseen data.

TESTING DATASET

In machine learning, a training dataset is a set of data used to train a machine learning algorithm to make predictions or classifications. The training dataset is a subset of the overall dataset that is used to train the model by adjusting its parameters until it can accurately predict the target variable.

The training dataset is usually created by partitioning the overall dataset into a training set and a testing set.

The training set is used to fit the model, while the testing set is used to evaluate the model's performance. This process is known as supervised learning, where the algorithm learns from labeled examples.

The training dataset typically includes input features, or independent variables, and the corresponding output, or dependent variable. The goal is to find a model that can predict the output variable given the input features.

It's important to note that the training dataset should be representative of the overall dataset and should have enough variability to ensure that the model is robust and can generalize well to new data. If the training dataset is too small or biased, the model may overfit or underfit the data, leading to poor performance on new, unseen data.

In machine learning, a testing dataset is a set of data used to evaluate the performance of a trained machine learning model. The testing dataset is separate from the training dataset and is used to assess the accuracy and generalization capability of the model on new, unseen data.

The testing dataset is created by partitioning the overall dataset into a training set and a testing set. The training set is used to train the model, while the testing set is used to evaluate the model's performance. The testing dataset should have a similar distribution to the training dataset and should not be used in any way during the training process to prevent overfitting.

When evaluating a model's performance using a testing dataset, common metrics include accuracy, precision, recall, F1 score, and AUC-ROC. The choice of metric depends on the specific problem being solved and the desired trade-offs between different types of errors. It's important to note that the testing dataset should not be used to adjust the model's parameters, as this can lead to overfitting and poor generalization to new data. Instead, hyperparameter tuning should be performed on a separate validation dataset.

STOPWORDS

In natural language processing, stopwords are words that are commonly used in a language but do not carry significant meaning or value in the context of a specific text. These words are often removed from the text before performing any analysis or processing, as they can cause noise and affect the accuracy of the results.

Examples of stopwords in English include "the", "a", "an", "in", "on", "at", "and", "of", "to",

"for", "which", "whose", "such", "there", "you", "your", "yours", "they", "their", "theirs" etc

Removing stopwords from text can help reduce the number of features in the dataset and improve the efficiency of machine learning algorithms. However, it's important to note that some stopwords may carry contextual meaning and should be retained in certain analyses.

Additionally, the list of stopwords may vary depending on the language and the specific context of the text being analyzed.

PRECISION AND RECALL

Precision and recall are two performance metrics commonly used in machine learning for evaluating the performance of a binary classification model. They are calculated based on the values in a confusion matrix.

Precision measures the proportion of true positives (TP) among all instances predicted as positive by the model. It is calculated as:

$$\text{precision} = \text{TP} / (\text{TP} + \text{FP})$$

In other words, precision measures how accurate the positive predictions made by the model are.

A high precision value indicates that the model is making few false positive predictions, which is desirable in many applications such as spam filtering.

Recall measures the proportion of true positives (TP) among all instances that belong to the positive class. It is calculated as:

$$\text{recall} = \text{TP} / (\text{TP} + \text{FN})$$

In other words, recall measures how well the model is able to identify all instances that belong to the positive class. A high recall value indicates that the model is making few false negative predictions, which is desirable in many applications such as disease diagnosis.

The choice between precision and recall depends on the specific application and the trade-offs between false positives and false negatives. For example, in a spam filtering application, it may be more important to have a high precision value to minimize false positives, even if it means some spam messages are missed (resulting in a lower recall value). On the other hand, in a disease diagnosis application, it may be more important to have a high recall value to minimize false negatives, even if it means some healthy patients are misdiagnosed (resulting in a lower precision value).

F1 SCORE AND SUPPORT

F1 score is another performance metric commonly used in machine learning for evaluating the performance of a binary classification model. It is a weighted average of precision and recall, and it is calculated as:

$$\text{F1 score} = 2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$$

The F1 score provides a balance between precision and recall and can be a useful metric for comparing the performance of different models.

Support is the number of instances in each class in the dataset. It is often used in conjunction with precision, recall, and F1 score to provide additional context about the performance of the model. For example, if the support for the positive class is very small, then a high precision value may be less meaningful because it may be due to chance. In such cases, it may be more appropriate to look at the recall or F1 score, which take into account both true positives and false negatives.

STEPS

- *Importing the needed python libraries*

```
!pip install scikit-plot
from wordcloud import WordCloud, STOPWORDS

from sklearn.metrics import accuracy_score, classification_report

from sklearn.linear_model import SGDClassifier
```

- **Uploading the dataset into the environment.**

```
dataset = pd.read_csv('/content/245_1.csv')
```

- **Reading the dataset using count, head, tail functions.**

```
dataset.head()
dataset.shape
dataset.isnull().sum()
```

```
dataset = dataset[['brand','manufacturer','reviews.didPurchase','reviews.rating', 'reviews.text']]
```

- **Cleansing the data for prediction.**

```
dataset['reviews.didPurchase'] = dataset['reviews.didPurchase'].fillna('Not Avialable')  
dataset = dataset.dropna()
```

```
dataset.isnull().sum()
```

- **Visualizing data using matplotlib and seaborn libraries**

```
data = dataset['reviews.rating'].value_counts()  
data = dataset['reviews.rating'].value_counts()  
sns.barplot(x=data.index, y=data.values)
```

```
ax_plt = sns.countplot(dataset['reviews.didPurchase'])  
ax_plt.set_xlabel(xlabel="User's Reviews",fontsize=12)  
ax_plt.set_ylabel(ylabel='No. of Reviews',fontsize=12)  
ax_plt.axes.set_title('Accurate No. of Reviews',fontsize=12)
```

```
ax_plt.tick_params(labelsize=11)
```

- **Preparing the stopword cloud according to the dataset**

```
stopwords = set(STOPWORDS)  
def wordcloud(data, title = None):  
wordcloud = WordCloud(  
    background_color='white',
```

```
stopwords=stopwords,  
    max_words=250,  
    max_font_size=30,  
    scale=2,
```

```
random_state=5 #chosen a andom by flipping a coin; it was heads  
) .generate(str(data))
```

```
fig = plt.figure(1, figsize=(15, 15))
```

- **Dividing the dataset into training and test sets upon which the algorithm will work**

```
X_train, X_test, y_train, y_test = train_test_split(train_features,  
y_target, test_size=0.3, random_state=101, shuffle=True)
```

- **Using LinearSVC algorithm to predict the accuracy**

```
lsvm = LinearSVC(class_weight='balanced')  
lsvm = lsvm.fit(X_train, y_train)
```

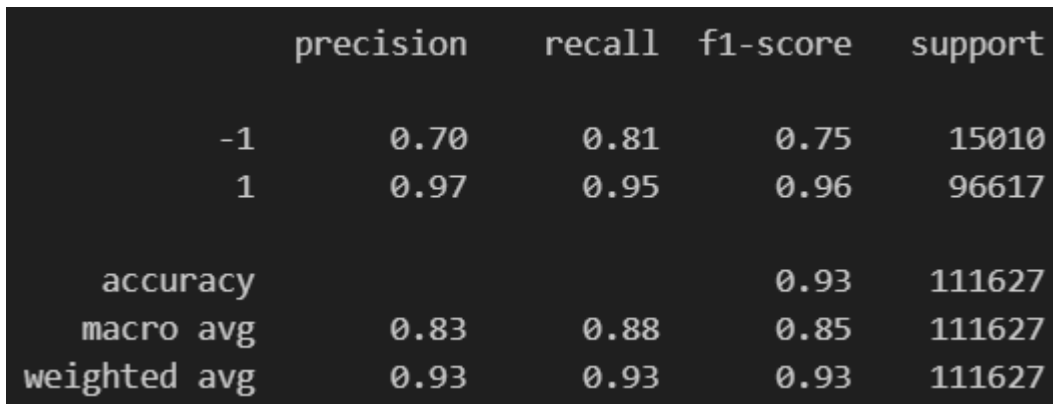
- **Predicting accuracy of the training dataset**

```
pred_train = lsvm.predict(X_train) print("Accuracy Train:  
{:.format(accuracy_score(y_train, pred_train))
```

```
"Accuracy Train: {}".format(accuracy_score(y_train,pred_train))
print(classification_report(y_train,pred_train))
```

- **Predicting accuracy of the test dataset**

```
print("Accuracy Test : {}".format(accuracy_score(y_test,pred_test)))
print(classification_report(y_test,pred_test))
skplt.metrics.plot_confusion_matrix(y_test, pred_test, normalize=True)
plt.show()
```



	precision	recall	f1-score	support
-1	0.70	0.81	0.75	15010
1	0.97	0.95	0.96	96617
accuracy			0.93	111627
macro avg	0.83	0.88	0.85	111627
weighted avg	0.93	0.93	0.93	111627

Fig 8: Accuracy chart on test data

COMPARATIVE STUDY

Different researches have been done on sentiment analysis using different machine learning and natural language processing techniques.

So, In this section, the comparison between previous research and our study is given for proper implications.

- LIMITATIONS OF PREVIOUS WORKS:

- 1 Data preparation is not done correctly which results in loss of data.
- 2 The splitting of train and test data was improperly distributed so the accuracy differs on both the aspects

The prototyping has been started by making the architectural plan and a code to analyse the data fetched.

This research used SVC and SGD classifier algorithm providing the accuracy of 93%.

CHAPTER - 5

CONCLUSION

We have used python and pandas operations to perform the sentiment analysis of the dataset obtained. It provides an easy-to-use visual representation of the dataset, working environment and building the predictive analytics.

We also have used matplotlib and seaborn for good visualization of the dataset for better and easy understanding as well as deep learning neural networks.

REFERENCES

1Marouane Birjali, Mohammed Kasri Abderrahim, Beni- Hssane“A comprehensive survey on sentiment analysis: Approaches, challenges and trends”, in conference 2008

2Adams W, Iyengar G, Lin CY, Naphade MR, Neti C, Nock HJ, Smith JR, “Semantic indexing of multimedia content using visual, audio, and text cues”,Journal of engineering 2003

3Asghar MZ, Khan A, Ahmad S, Qasim M, Khan IA, “Lexicon-enhanced sentiment analysis framework”, Journal 2017

4Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan, “Thumbs up? Sentiment classification using machine learning techniques”, conference for engineering,2015

5Alekha Agarwal and Pushpak Bhattacharyya, “Sentiment analysis”: A new approach for effective use of linguistic knowledge” ,journal of machine learning,2017

6 Anindya Ghose, Panagiotis G. Ipeirotis, “Designing Novel Review Ranking Systems: Predicting Usefulness and Impact of Reviews”, PhD,2015

7Kontopoulos E, Berberidis C, Dergiades T al (2013) Ontology-based sentiment analysis of twitter posts. *Expert Syst Appl* 40(10):4065–4074

8Li L, S, Cao D, et al (2017) SentiNet: Mining visual sentiment from scratch. In: *Advances Computational Intelligence Systems*, 309–317

9Liu H, Jou , Chen T, et al (2016) Complura: Exploring and leveraging a large-scale multilingual visual sentiment ontology. In: *Proceedings the 2016 ACM International Conference on Multimedia Retrieval*, 417–420

10Ravi K, Ravi (2015) A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowl-Based Syst* 89(C):14–46

11Tang D, Wei F, et al (2014) Learning sentiment-specific word embedding for Twitter sentiment classification. In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics (ACL’14)*, 1555–1565

12Tang D, Qin B, et al (2015) User modelling with neural network for review rating prediction. In: *Proceedings of 24th International Conference on Artificial Intelligence (IJCAI’15)*, 1340–1346

13Wang X, Wei F, et al (2011) Topic sentiment analysis in Twitter: a graph-based hashtag sentiment classification approach. In: *Proceedings of the 20th ACM international conference on Information and knowledge management*, 1031–1040

14Wang J, Fu J, Xu Y, et al (2016) Beyond object recognition: Visual sentiment analysis with deeply coupled adjective and noun neural networks. In: *Proceedings of the 25th International Joint Conference on Artificial Intelligence* , 3484–3490