SENTIMENT ANALYSIS ON PRODUCT REVIEWS

A Course Based Project Report in Data Engineering

Submitted in partial fulfillment of the requirements for the award of the degree of

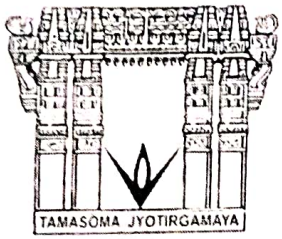
BACHELOR OF TECHNOLOGY IN
CSE (Internet of Things)

Submitted by

Akhila	(22071A6927)
Ruchitanjani	(22071A6930)
Abhi Sree	(22071A6936)
Srinayani	(22071A6940)

Under the guidance of

Naga Durga Saile. K
Assistant Professor,
(Department of CSE-AIML & IoT, VNR VJIET)



Department of CSE

Artificial Intelligence and Machine Learning & Internet of Things

Vallurupalli Nageswara Rao Vignana Jyothi Institute of

Engineering And Technology

(An Autonomous Institute, NAAC Accredited with 'A++' Grade NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT & B. Tech Courses Approved by AICTE, New Delhi, Affiliated to JNTUH Recognized as "College with Potential for Excellence" by UGC) ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institute, NAAC Accredited with 'A++' Grade, NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT & B. Tech Courses, Approved by AICTE, New Delhi. Affiliated to JNTUH, Approved as "Potential for Excellence" by UGC) ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

DEPARTMENT OF CSE

Artificial Intelligence and Machine Learning & Internet of Things

CERTIFICATE

This is to certify that the course based project report entitled "**Sentiment Analysis on Product Reviews**" is bonafide work done under our supervision and is being submitted by Akhila (22071A6927),Ruchitanjani(22071A6930),AbhiSree(22071A6936),Srinayani(22071A6940) in partial fulfillment for the award of the degree of Bachelor of Technology in CSE(Internet of Things), of the VNRVJIET, Hyderabad during the academic year 2024-2025. Certified further that to the best of our knowledge the work presented in this thesis has not been submitted to any other University or Institute for the award of any Degree or Diploma.

Sagar Yeruva.

Naga Durga Saile. K
Assistant Professor
Dept. of CSE (AIML & IoT)
VNR VJIET

Dr. Sagar Yeruva
Associate Professor & HoD
Dept. of CSE (AIML & IoT)
VNR VJIET

DECLARATION

We declare that the major project work entitled "**Sentiment Analysis on Product Reviews**" submitted in the department of CSE(Internet of Things), Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology, Hyderabad, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **CSE(Internet of Things)** is a bonafide record of our own work carried out under the supervision of, Naga Durga Saile. K, Assistant Professor Department of CSE(AIML & IoT), VNRVJIET. Also, we declare that the matter embodied in this thesis has not been submitted by us in full or in any part thereof for the award of any degree/diploma of any other institution or university previously.

Akhila	(22071A6927) Akhi
Ruchitanjani	(22071A6930) Ruchitanjani
Abhi Sree	(22071A6936) Abhi
Srinayani	(22071A6940) Srinayani

ACKNOWLEDGEMENT

Firstly, we would like to express our immense gratitude towards our institution VNR Vignana Jyothi Institute of Engineering and Technology, which created a great platform to attain profound technical skills in the field of Computer Science, thereby fulfilling our most cherished goal.

We are very much thankful to our Principal, **Dr. Challa Dhanunjaya Naidu,** and our Head of Department, **Dr. Sagar Yeruva**, Associate Professor for extending their cooperation in doing this project within the stipulated time.

We extend our heartfelt thanks to our guide **Naga Durga Saile**. K, Assistant Professor for enthusiastic guidance throughout the course of our project.

Finally, our appreciable obligation also goes to all the staff members of the Computer Science & Engineering (AIML & IoT) department and to our fellow classmates who directly or indirectly helped us.

Akhila	(22071A6927)
Ruchitanjani	(22071A6930)
Abhi Sree	(22071A6936)
Srinayani	(22071A6940)

ABSTRACT

Customer reviews play a pivotal role in shaping business strategies and guiding consumer decisions in today's digital marketplace. This project leverages sentiment analysis to extract valuable insights from Flipkart product reviews using natural language processing (NLP) and machine learning techniques. The dataset, comprising textual reviews and numerical ratings, undergoes preprocessing to remove noise, tokenize text, and eliminate stopwords, ensuring effective feature extraction. Key features are derived using TF-IDF vectorization, which captures the significance of terms within the context of the reviews, transforming them into numerical representations for analysis.

A Decision Tree Classifier is trained to classify reviews into positive, neutral and negative sentiments, demonstrating high accuracy in predicting customer sentiment. Visual tools like word clouds provide an intuitive understanding of commonly used terms in positive, neutral and negative reviews, enhancing the model's interpretability. The project also incorporates a real-time sentiment prediction system, enabling instant analysis of new reviews for practical applications in e-commerce, customer support, and feedback systems, thus offering a robust solution for understanding and improving customer satisfaction.

LIST OF FIGURES

CONTENTS

1. INTRODUCTION

1.1 SENTIMENT ANALYSIS AND ITS ROLE IN E-COMMERCE

In the era of digital transformation, e-commerce platforms have become integral to the global marketplace. A critical aspect of these platforms is the ability to gather and analyze customer feedback through product reviews. These reviews provide valuable insights into customer satisfaction, product quality, and service efficiency. However, the vast volume of customer feedback poses a significant challenge to manual analysis.

Sentiment analysis, a subfield of natural language processing (NLP), offers a solution by automating the process of understanding the sentiment expressed in textual data. By classifying reviews as positive, neutral, or negative, sentiment analysis helps businesses gain actionable insights at scale. For example, identifying recurring complaints about a product's build quality or highlighting features like excellent battery life can directly influence product development and marketing strategies.

In this context, sentiment analysis is not just a tool for understanding customer opinions but a competitive advantage in improving customer satisfaction, retention, and brand loyalty. With advancements in machine learning and artificial intelligence, automated sentiment analysis has become more accurate and reliable, making it an essential component of the e-commerce industry.

1.2 OBJECTIVE

The objective of this project is to develop a sentiment analysis model that classifies product reviews into positive, neutral or negative sentiments based on textual feedback and associated ratings. This project leverages machine learning techniques, specifically a Decision Tree Classifier, to analyze and categorize reviews efficiently.

The primary goals are as follows:

Data Preparation: Preprocess the dataset by cleaning and normalizing the text, removing noise, and extracting meaningful features using TF-IDF vectorization.

Sentiment Classification: Train a machine learning model to classify reviews with ratings of 5 as positive, ratings of 4 as neutral and lower ratings as negative.

Insights Generation: Visualize common terms associated with each sentiment category through word clouds and analyze trends in customer feedback.

Real-time Prediction: Develop a framework to predict the sentiment of new reviews, enabling practical application in monitoring customer satisfaction and identifying areas for improvement.

2. LITERATURE SURVEY

[1] Now a day's internet is the most valuable source of learning, getting ideas, reviews for a product or a service. Everyday millions of reviews are generated in the internet about a product, person or a place. Because of their huge number and size it is very difficult to handle and understand such reviews. Sentiment analysis is such a research area which understands and extracts the opinion from the given review and the analysis process includes natural language processing (NLP), computational linguistics, text analytics and classifying the polarity of the opinion. In the field of sentiment analysis there are many algorithms exist to tackle NLP problems. Each algorithm is used by several applications. In this paper we have shown the taxonomy of various sentiment analysis methods. This paper also shows that Support vector machine (SVM) gives high accuracy compared to Naïve bayes and maximum entropy methods.

[2] In this paper, they present a novel approach to identify feature-specific expressions of opinion in product reviews with different features and mixed emotions. The objective is realized by identifying a set of potential features in the review and extracting opinion expressions about those features by exploiting their associations. Capitalizing on the view that more closely associated words come together to express an opinion about a certain feature, dependency parsing is used to identify relations between the opinion expressions. The system learns the set of significant relations to be used by dependency parsing and a threshold parameter that allows them to merge closely associated opinion expressions. The data requirement is minimal as this is a one-time learning of the domain-independent parameters. The associations are represented in the form of a graph that is partitioned to finally retrieve the opinion expression describing the user-specified feature. They show that the system achieves a high accuracy across all domains and performs at par with state-of-the-art systems despite its data limitations.

[3] Sentiment analysis is used for Natural language Processing, text analysis, text preprocessing, Stemming etc. are the major research field in current time. Sentiment analysis using different techniques and tools for analyze the unstructured data in a manner that objective results can be generated from them. Basically, these techniques allow a computer to understand what is being said by humans. Sentiment analysis uses different techniques to determine the sentiment of a text or sentence. The Internet is a large repository of natural language. People share their thoughts and experiences which are subjective in nature. Many a time, getting suitable information about a product can became tedious for customers. Companies may not be fully aware of customer requirements. Product reviews can be analyzed to understand the sentiment of the people towards a particular topic. However, these are voluminous; therefore a summary of positive and negative reviews needs to be generated. In this paper, the main focus is on the review of algorithms and techniques used for extract feature wise summary of the product and analyzed them to form an authentic review. Future work will include more product reviews websites and will focus on higher level natural language processing tasks. Using best and new techniques or tool for more accurate result in which the system except only those keywords which are in dataset rest of the words are eliminated by the system.

[4] Online customer reviews are a vital source of market insights, influencing purchase decisions and helping businesses understand customer preferences and experiences. However, extracting meaningful insights from large volumes of reviews is a significant challenge. This research analyzes 28,995 Flipkart reviews on handicraft products to classify them into positive, neutral, and negative sentiments using machine learning algorithms such as Logistic Regression (LR), K-Nearest Neighbors (KNN),

Multilayer Perceptron (MLP), and Support Vector Machine (SVM). Employing TF-IDF and Count Vectorizer, SVM demonstrated the highest accuracy compared to other classifiers, highlighting its effectiveness in sentiment analysis. This study provides insights into customer perceptions of handicraft products on e-commerce platforms.

[5] E-commerce platforms and sellers constantly seek consumer feedback in the current digital retail environment, particularly regarding the customers' experiences with the product. The volume of customer evaluations and product comments posted online has significantly increased as a result of this new trend, which reflects a rising preference for online buying via the well-known Indian e-commerce site Flipkart (https://www.flipkart.com). The opinions and experiences of other customers are frequently relied upon by prospective buyers, therefore the sentiments conveyed in these evaluations have a considerable impact on their decisions. We focused our investigation on iPhone product reviews. The Bag-of-Words methodology was used to convert iPhone product reviews into numerical representations to start the process. Additionally, a word cloud was used to show word frequency throughout the evaluations to graphically demonstrate the text-based data. The investigation was then carried out using a range of machine learning techniques, including decision trees and logistic regression. Impressively, when examining the dataset made up of reviews of iPhone products and the Flipkart website, both machine learning models displayed outstanding accuracy. Notably, With a 99% exceptional accuracy rate, the Decision Tree model surpassed the Logistic Regression model, which was only 92% accurate at the time.

2.1 SYSTEM STUDY

The sentiment analysis system for Flipkart reviews employs a combination of machine learning algorithms to classify customer reviews into positive, negative, or neutral sentiments. This study provides an overview of the algorithms used, their working principles, and suitability for the project.

2.1.1 Decision Tree Classifier

A Decision Tree is a non-linear, supervised machine learning algorithm commonly used for classification and regression tasks. It works by splitting the data into subsets based on the most significant features that best separate the data points. The splits continue recursively, forming a tree-like structure, which ends in leaf nodes that represent the class label.

Working:

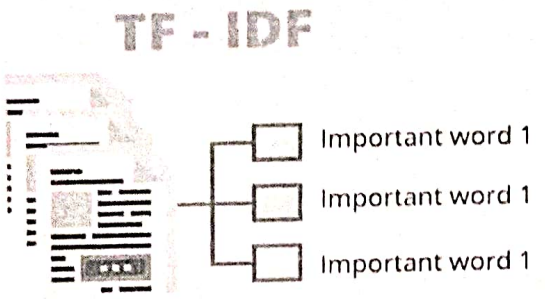
In this project, a **Decision Tree Classifier** is used to predict whether a review is positive, neutral or negative based on the TF-IDF transformed features (the text data). Here's a simplified flow of how it works:

- **Training Phase**: The Decision Tree model is trained on the TF-IDF transformed reviews (X_train) and their associated sentiment labels (y_train). It uses features (words/terms) extracted from the reviews to find the best splits. The model finds features (words) that most effectively separate positive, neutral and negative ones.

- **Prediction Phase**: Once the model is trained, you can use it to predict sentiment for unseen data (in this case, new reviews). The model evaluates which path (split) a new review falls into based on the words in the review and assigns it to either the positive, neutral or negative sentiment class.

2.1.2 TF-IDF Vectorization

TF-IDF (Term Frequency-Inverse Document Frequency) is a statistical measure used to evaluate the importance of a word in a document relative to a corpus. It is used to convert the raw text into numerical features (a vector), which can then be fed into machine learning models like Decision



Trees.

Fig1:TF-IDF vectorization

- Term Frequency (TF) measures how frequently a term appears in a document.
- Inverse Document Frequency (IDF) measures how important a term is across the entire dataset. It decreases the weight of terms that appear frequently across all documents, indicating that those words are less significant.

Working:

1. **TF Calculation**: For each document (review), TF counts the frequency of each word.
2. **IDF Calculation**: IDF computes how rare or common a word is across all documents in the dataset. If a word appears in almost every document, it gets a lower weight, as it doesn't provide much information for classification.
3. **TF-IDF Calculation**: The final weight of a word in a document is calculated as the product of its TF and IDF values. Words with higher TF-IDF scores are considered more important because they appear frequently in specific documents but not across many documents.
4. **Feature Extraction**: The TfidfVectorizer is used to convert the review text into a sparse matrix of TF-IDF features, which represents the importance of each word in each review. This matrix is then used as input data for the machine learning model.

3. SYSTEM REQUIREMENTS

3.1 REQUIREMENT SPECIFICATION

3.1.1 Functional Requirements

Every software application must fulfill a large number of functions in order for it to operate correctly. These functions are nothing more than different tasks that are carried out during each stage of the application development process. This step is included in the best practices for creating applications. Together, functional and non-functional requirements establish a set of rules for the efficient operation of an application. They also assist the user and developer in identifying the software and hardware requirements necessary to operate the application. Functional Requirements that are required to run this application are:

1. Load a CSV file (flipkart_data.csv) containing reviews and ratings into a Pandas DataFrame.
2. Create visualizations, including count plots for rating distributions and word clouds for positive and negative reviews.
3. Preprocess text data by removing punctuation, converting to lowercase, and eliminating stopwords.
4. Assign binary labels to reviews based on their ratings (1 for positive, 0 for neutral and -1 for negative).
5. Convert preprocessed text reviews into numerical feature representations using TF-IDF vectorization.
6. Train a decision tree classifier on the vectorized data.
7. Allow predictions for new reviews by preprocessing them and applying the trained model.

3.1.2 Non Functional Requirements

Non-functional criteria are used to set conditions for tracking an application's performance characteristic. It explains the working of a specific application function. They also determine the overall quality of the project and hence it is a very important aspect in any software development process.

The Non-Functional Requirements include:

1.Performance:Ensure that preprocessing, visualization, and model training are completed efficiently for datasets up to 100,000 entries. Use libraries like pandas and numpy for optimized data handling and ensure machine learning tasks are executed promptly. Profile and benchmark the pipeline to identify bottlenecks and optimize for speed.

2. Scalability:The solution should handle larger datasets by processing data in batches and leveraging distributed computing tools like Dask or Spark. Replace memory-intensive operations with disk-based alternatives for scalability. Use cloud environments or GPUs for faster processing of large-scale data.

3. Usability:Provide intuitive visualizations like histograms and word clouds with clear labels and legends. Ensure the sentiment prediction function is user-friendly, with simple inputs and well-documented outputs. Incorporate detailed error messages to help users troubleshoot easily.

4. Reliability:Use robust libraries like nltk and scikit-learn for consistent and reliable text processing and modeling. Ensure all code is well-tested and includes error handling for common issues, such as missing data or invalid input. Regularly update dependencies to maintain compatibility.

5. Efficience:Optimize performance by leveraging vectorized operations for preprocessing and

parallel processing for computationally intensive tasks. Cache intermediate results to minimize redundant computations. Profile code with tools like cProfile to identify and eliminate inefficiencies.

6. Portability:Ensure compatibility across Windows, macOS, and Linux by avoiding OS-specific paths and functions. Use relative paths or environment variables for file handling and provide a requirements.txt for dependency management. Test the solution on multiple platforms to guarantee portability.

7. Maintainability:Write modular code with separate functions for preprocessing, visualization, and prediction to simplify updates. Use clear comments and documentation for better readability. Modular design ensures easy integration of new features or updates without affecting the existing functionality.

3.1.3 Software

Requirements Software:

1. Operating System:Windows 10/11, Linux, or macOS.
2. Programming Environment:Python 3.8 or higher.
3. Required Libraries:

pandas: For data manipulation and analysis.
seaborn: For creating count plots and visualizations.
matplotlib: For plotting word clouds and other graphics.
nltk: For text preprocessing, including stopwords and tokenization.
wordcloud: For generating word clouds.
scikit-learn: For TF-IDF vectorization, model training, and prediction.
tqdm: For tracking the progress of preprocessing tasks.

4. Development Tools:

Any Python-compatible IDE or text editor (e.g., Jupyter Notebook, PyCharm, or Visual Studio Code)

3.1.4 Hardware

Requirements Hardware:

Processor: Intel I3 processor
Storage Space: 500 GB
Devices Required: Mouse and a Keyboard
Minimum Ram: 4GB and a good Internet connection

3.2 METHODOLOGY

1. Data Loading

The Flipkart reviews dataset is imported as a CSV file using the pandas library. The dataset contains key attributes such as review (textual feedback) and rating (numerical scores). Exploratory data analysis is performed to understand the structure, identify missing values, and determine the distribution of ratings. The head() function is used to preview the dataset, and unique ratings are listed for reference.

2. Data Preprocessing

Text reviews are cleaned to prepare them for machine learning. This involves:

Balancing Dataset: Reduced the over represented rating 5 by randomly selected 1700 reviews with rating of 5.

Removing Punctuation: All special characters and punctuation are eliminated using regular expressions.

Lowercasing: Text is converted to lowercase to ensure uniformity.

Stopword Removal: Common words like "and" or "the" are removed using the nltk stopword list to reduce noise.

Additionally, numerical ratings are converted into binary labels: ratings = 5 are labeled as positive (1), ratings=4 are labeled as neutral (0), while others are labeled as negative (-1). This prepares the dataset for classification.

3. Feature Extraction

Preprocessed text reviews are transformed into numerical vectors using the TF-IDF (Term Frequency-Inverse Document Frequency) technique. This captures the importance of words within the reviews while minimizing the impact of frequently occurring but unimportant words. The TfidfVectorizer from scikit-learn is configured to extract a maximum of 2,500 features, ensuring a balance between dimensionality and computational efficiency.

4. Model Training

A machine learning model is trained to classify reviews as positive, neutral or negative. The dataset is split into training and testing subsets using an 80:20 ratio to ensure proper evaluation. A decision tree classifier is implemented using scikit-learn, which is trained on the TF-IDF features. The model's accuracy is evaluated, and performance metrics are computed to ensure reliability.

5. Visualization

Count Plots: Seaborn is used to plot the distribution of ratings, showing the frequency of each label.

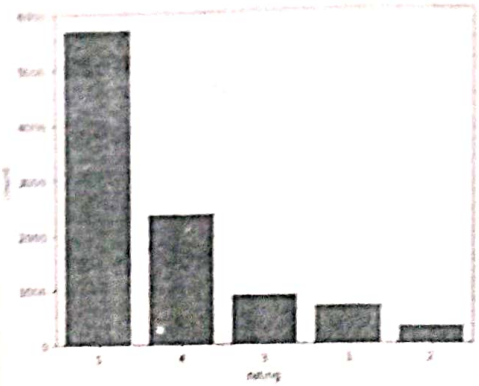


Fig2:Rating count plot

Word Clouds: Positive, neutral and negative reviews are combined separately to generate word clouds using the wordcloud library, highlighting the most frequent terms in each category. These visualizations make the data interpretable and allow for a better understanding of patterns.

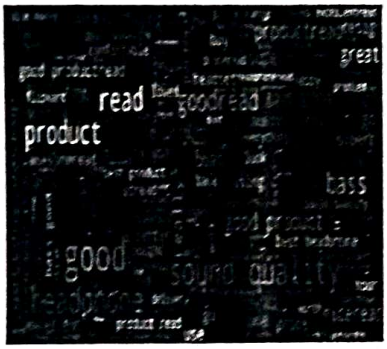


Fig3: Positive Wordclouds

Fig4: Negative Wordclouds

Fig5: Neutral WordClouds

6. Sentiment Prediction

A dedicated function is implemented for predicting the sentiment of new reviews. The function preprocesses the input review text (removing punctuation, lowercasing, and removing stopwords) and converts it into numerical features using the trained TF-IDF vectorizer. The trained decision tree model then predicts the sentiment as either positive, neutral or negative. This ensures the solution is reusable and scalable for real-world applications.

WORK FLOWCHART:

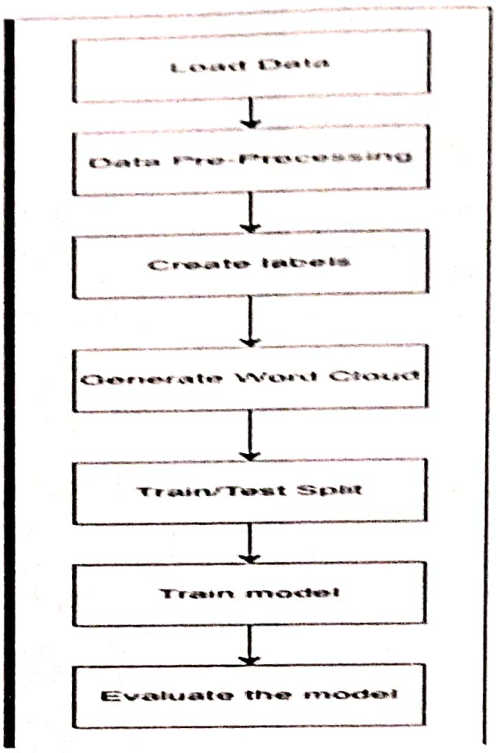


Fig 6: Work flowchart

4.DATASET

4.1Dataset source: Kaggle

4.2 Description

The dataset used for this analysis contains 5,105 records and 2 columns:

- review: Contains customer reviews in text, providing feedback on various products.
- rating: Numerical ratings given by customers, ranging from 1 to 5.
- label: Sentiment labels assigned to each review:
    - 1: Positive (rating = 5)
    - 0: Neutral (rating = 4)
    - -1: Negative (rating ≤ 3)

Summary of the Dataset:

- Shape: (5105, 3), indicating 5,105 rows and 2 columns.
- Missing Values: None in any of the columns.

This balanced dataset provides a combination of textual and numerical data for multi-class sentiment analysis, making it well-suited for customer feedback classification.

5.UML DIAGRAMS

UML diagrams are used to understand the system in a better and simple way. The elements are like components which can be associated in different ways to make a complete UML picture, which is known as a diagram. There are different types of UML Diagrams.

5.1 Sequence Diagram:

A Sequence Diagram is a key component of Unified Modeling Language (UML) used to visualize the interaction between objects in a sequential order. It focuses on how objects communicate with each other over time, making it an essential tool for modeling dynamic behavior in a system. Sequence diagrams illustrate object interactions, message flows, and the sequence of operations, making them valuable for understanding use cases, designing system architecture, and documenting complex processes.

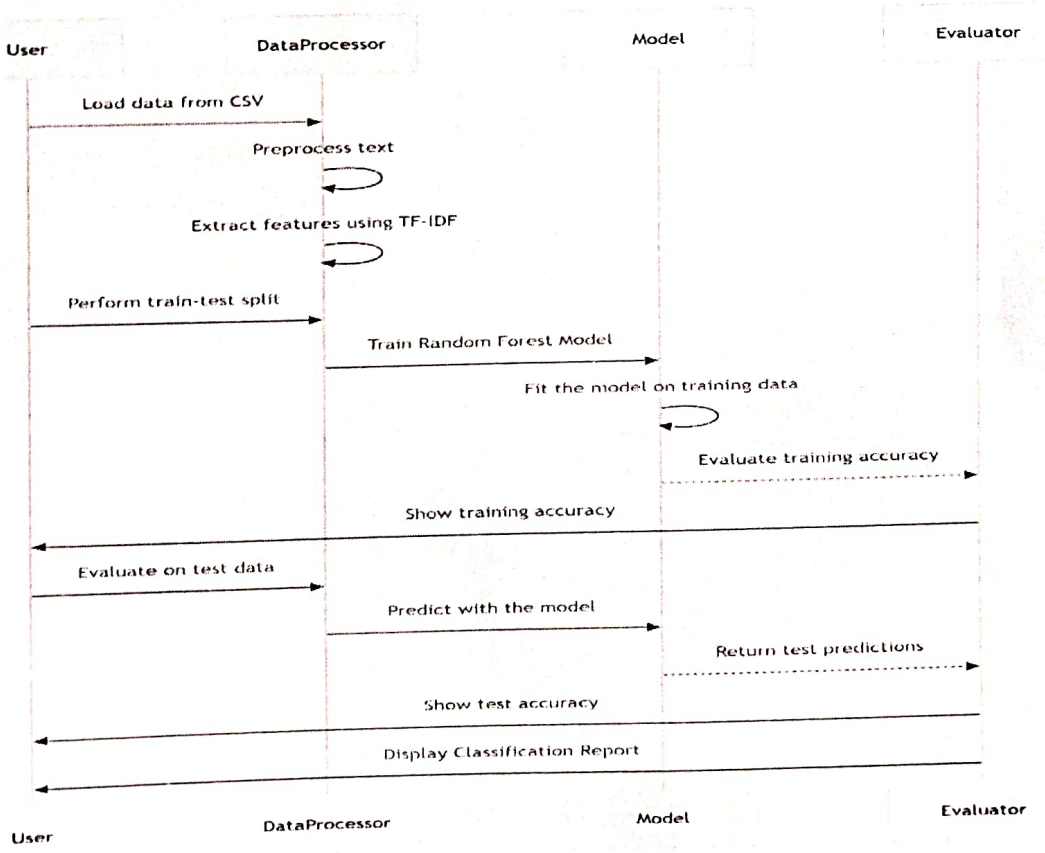


Fig 7: Sequence Diagram

5.2 Activity Diagram:

An application's control flow or sequence is shown in an activity diagram. In fact, we may utilise the activity diagram to validate each action depicted in the use-case diagram. It also shows the process steps. They basically show how the software application's workflows are done. It also places emphasis on the order in which tasks should be completed and the conditions that must be met for each activity to be successful. In this manner, it informs us about the factors that influence a specific task.

As a result, we can now see the application at a high level .The primary goals are:

- Depict the activity flow
- Describe the sequence (branched or sequential)

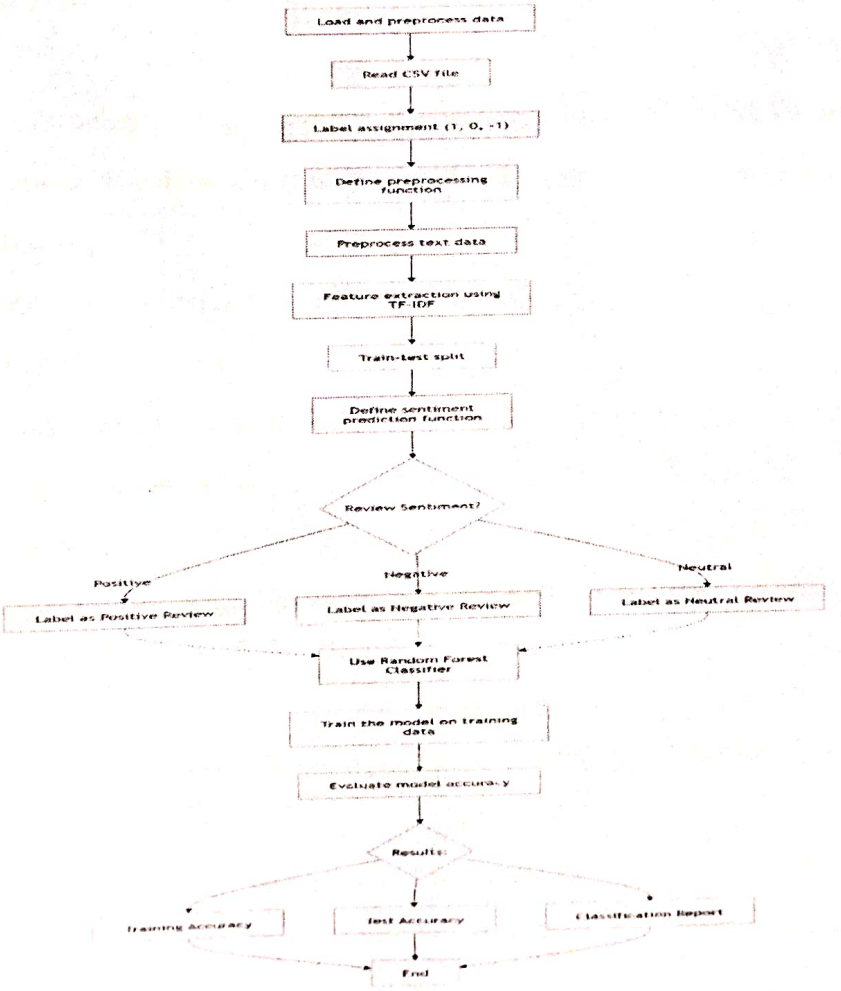


Fig 8: Activity Diagram

6. IMPLEMENTATION

6.1 MODULES

The proposed system has 3 modules:

1. Data Pre-Processing
2. Training Model Module
3. Testing Module

6.1.1 Data Pre-Processing Module

Description:

The goal of this module is to clean and prepare the dataset for further analysis. The dataset contains reviews and ratings, which are transformed into structured numerical data for the classification task.

Steps:

1. **Import Libraries**: Essential libraries like pandas, matplotlib, and NLTK are imported.
2. **Load Dataset**: The dataset is read from a CSV file into a pandas DataFrame.
3. **Data Cleaning**:
    - Balanced the dataset.
    - Removed punctuations.
    - Converted all text to lowercase.
    - Removed stopwords.
4. **Label Creation**:
    - Ratings are classified as positive (1) for ratings 5 and above, neutral (0) for ratings 4 and negative (-1) otherwise.
5. **Visualization**:
    - Word clouds are generated for positive, neutral and negative reviews to identify frequently used words.

Importing Libraries and Data Preparation

```
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import re
import seaborn as sns
from sklearn.feature_extraction.text import TfidfVectorizer
import matplotlib.pyplot as plt
```

```
from wordcloud import WordCloud
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from tqdm import tqdm
# Download necessary NLTK data
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
# Load dataset
data = pd.read_csv("C:/Users/kondu/Downloads/flipkart_data.csv")
```

- Libraries Import: Essential libraries like pandas for data manipulation, re for regular expressions, nltk for text processing, and matplotlib for data visualization are imported.
- Warnings Suppression: Warnings are ignored using warnings.filterwarnings('ignore') to avoid cluttering output with unnecessary messages.
- NLTK Downloads: NLTK's required resources (stopwords, punkt, and wordnet) are downloaded for stopword removal, tokenization, and lemmatization tasks.
- Dataset Loading: The dataset flipkart_data.csv is loaded into a pandas DataFrame for further processing.Creating Labels for Sentiment Classification.

Reduce the Over-represented Rating 5

```
reduced_positive_df = df[df['rating'] == 5].sample(n=1700, random_state=42)
other_ratings_df = df[df['rating'] != 5]
balanced_df = pd.concat([reduced_positive_df, other_ratings_df], ignore_index=True)
balanced_df.to_csv("reduced_rating5_dataset.csv", index=False)
```

- Downsample Rating 5: Randomly sample 1700 reviews with a rating of 5.
- Keep Other Ratings: Keep all reviews for ratings other than 5.
- Combine and Save: Combine the reduced 5 ratings with other ratings and save the new balanced dataset. Downsample Rating 5: Randomly sample 1700 reviews with a rating of 5.
- Keep Other Ratings: Keep all reviews for ratings other than 5.
- Combine and Save: Combine the reduced 5 ratings with other ratings and save the new balanced dataset.

Creating Labels for Sentiment Classification

```
data['label'] = data['rating'].apply(lambda x: 1 if x == 5 else (0 if x == 4 else -1))
```

1. Label Creation: A new column label is added to classify reviews based on ratings.

2. Rating Threshold: Reviews with a rating of 5 are labeled as positive (1), a rating of 4 as neutral (0), and ratings ≤ 3 as negative (-1).
3. Multi-Class Classification: This transforms the ratings into a three-class sentiment classification suitable for training a more nuanced sentiment analysis model.
4. Efficient Use of apply: The lambda function efficiently assigns the sentiment label based on the rating for each review.

Text Preprocessing

```
# Preprocess review text
def preprocess_text(text_data):
    preprocessed_text = []
    for sentence in tqdm(text_data):
        sentence = re.sub(r'[^\w\s]', '', sentence)  # Remove punctuations
        sentence = ' '.join(word.lower() for word in word_tokenize(sentence)
                            if word not in stopwords.words('english'))
        preprocessed_text.append(sentence)
    return preprocessed_text
data['review'] = preprocess_text(data['review'])
```

1. Text Cleaning: The function removes punctuation using re.sub() and tokenizes the sentence using word_tokenize().
2. Lowercasing: Converts all words in the review text to lowercase to standardize the text.
3. Stopword Removal: Uses NLTK's stopwords to eliminate common words (like "the", "is", "and") that don't contribute much to sentiment analysis.
4. Text Transformation: The function is applied to the review column to preprocess all reviews in the dataset.

Visualizing Word Clouds

```
# Visualize word clouds for positive, neutral, and negative reviews
for label, title in zip([1, 0, -1], ['Positive Review', 'Neutral Review', 'Negative Review']):
    text = ' '.join(data['review'][data['label'] == label].astype(str))
    wordcloud = WordCloud(width=1600, height=800, max_font_size=110).generate(text)
    plt.figure(figsize=(15, 10))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.title(f'{title} Word Cloud')
    plt.show()
```

Text Separation: Positive, neutral and negative reviews are separated based on the label column (1 for positive, 0 for neutral, -1 for negative).

Word Cloud Generation: WordCloud() is used to generate word clouds for positive, neutral and negative reviews.

Visualization: The word clouds are displayed using imshow() to create an interactive visualization without axis.

Max Font Size: The word clouds are designed to display the most frequent words with a large font size (max_font_size=110).

6.1.2 Training Model Module

Description:

In this module, the cleaned data is split into training and testing datasets. The machine learning model is trained using the **Decision Tree Classifier**.

Steps:

1. **Feature Extraction**: Convert text into numerical vectors using **TF-IDF Vectorizer**.
2. **Train-Test Split**: The dataset is divided into 80% training and 20% testing sets using train_test_split.
3. **Model Training**: A Decision Tree Classifier is trained on the training set.

Feature Extraction Using TF-IDF

```
# Feature extraction using TF-IDF
cv = TfidfVectorizer(max_features=2500)
X = cv.fit_transform(data['review']).toarray()
y = data['label']
```

1. TF-IDF Vectorization: TfidfVectorizer() transforms the review text into a matrix of numerical features using the TF-IDF method.
2. Max Features: Limits the features to the top 2500 terms based on their TF-IDF scores (max_features=2500).
3. Text-to-Numerical Conversion: The reviews are converted to a numerical matrix X and the sentiment labels to y.
4. Feature Matrix: X contains the transformed feature vectors of the reviews, and y holds the sentiment labels.

Splitting the Dataset into Training and Testing Sets

```
# Split dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, stratify=y, random_state=42)
```

1. Data Split: The dataset is split into training (67%) and testing (33%) sets using train_test_split().
2. Stratified Split: Ensures that positive, neutral and negative reviews are evenly distributed in both training and test sets (stratify=y).

3. Reproducibility: The random_state=42 ensures that the split is consistent across different runs.
4. Balanced Dataset: The stratified split maintains the proportion of positive, neutral and negative reviews in both sets.

Training the Decision Tree Model

```
# Train the Decision Tree Classifier
model = DecisionTreeClassifier(random_state=0)
model.fit(X_train, y_train)
 Evaluate the training accuracy
from sklearn.metrics import accuracy_score
train_predictions = model.predict(X_train)
print("Training Accuracy:", accuracy_score(y_train, train_predictions))
```

1. Model Initialization: A DecisionTreeClassifier is created and trained using the fit() method with the training data (X_train and y_train).
2. Training Process: The model learns from the training data to create decision rules for sentiment classification.
3. Model Evaluation: The training accuracy is calculated by comparing predictions (train_predictions) with actual labels (y_train).
4. Accuracy Output: The training accuracy is printed to evaluate how well the model has learned from the training data.

7.TESTING

- Using logistic regression:

Logistic Regression has been implemented and 77% accuracy score is obtained as shown in fig8

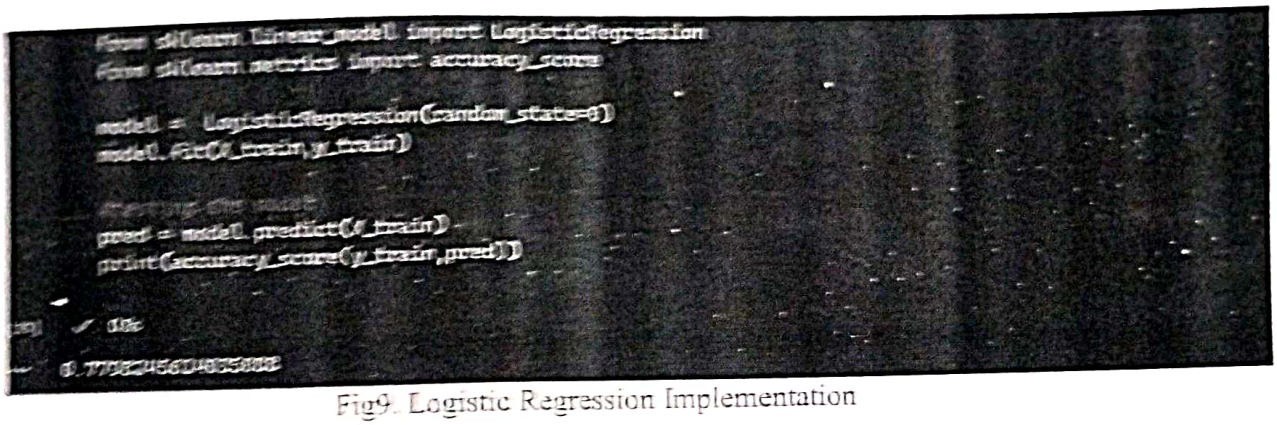


Fig9. Logistic Regression Implementation

- Using Decision Tree Classifier:

Decision Tree algorithm has been implemented and 96% accuracy score is obtained as shown in fig 9

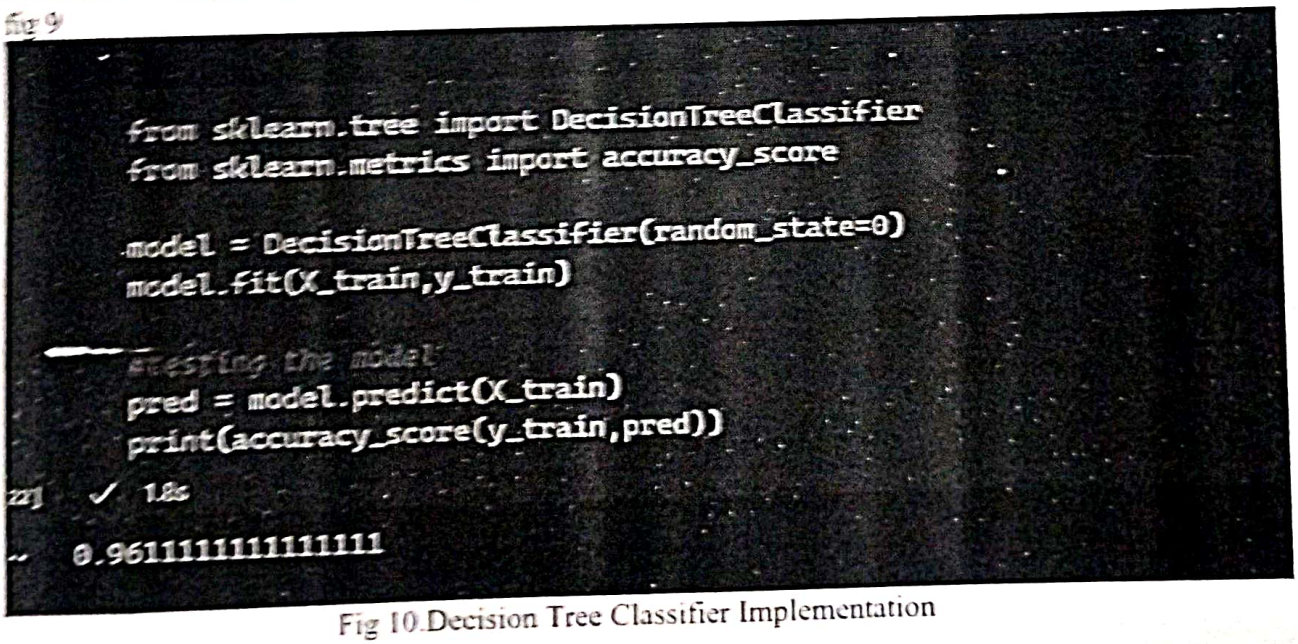


Fig 10.Decision Tree Classifier Implementation

- Using Random Forest Classifier:

Random Forest Classifier has been implemented and 57% accuracy score is obtained as shown in fig10

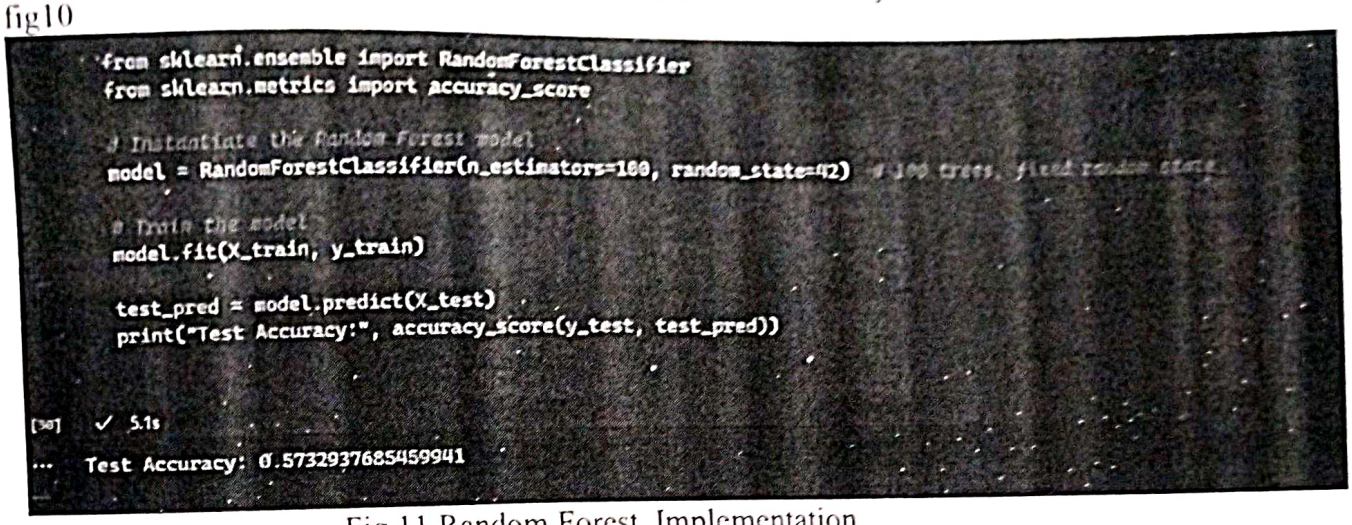


Fig 11.Random Forest Implementation

- Using SVM Algorithm:

SVM Algorithm has been implemented and 77% accuracy score is obtained as shown in fig11.

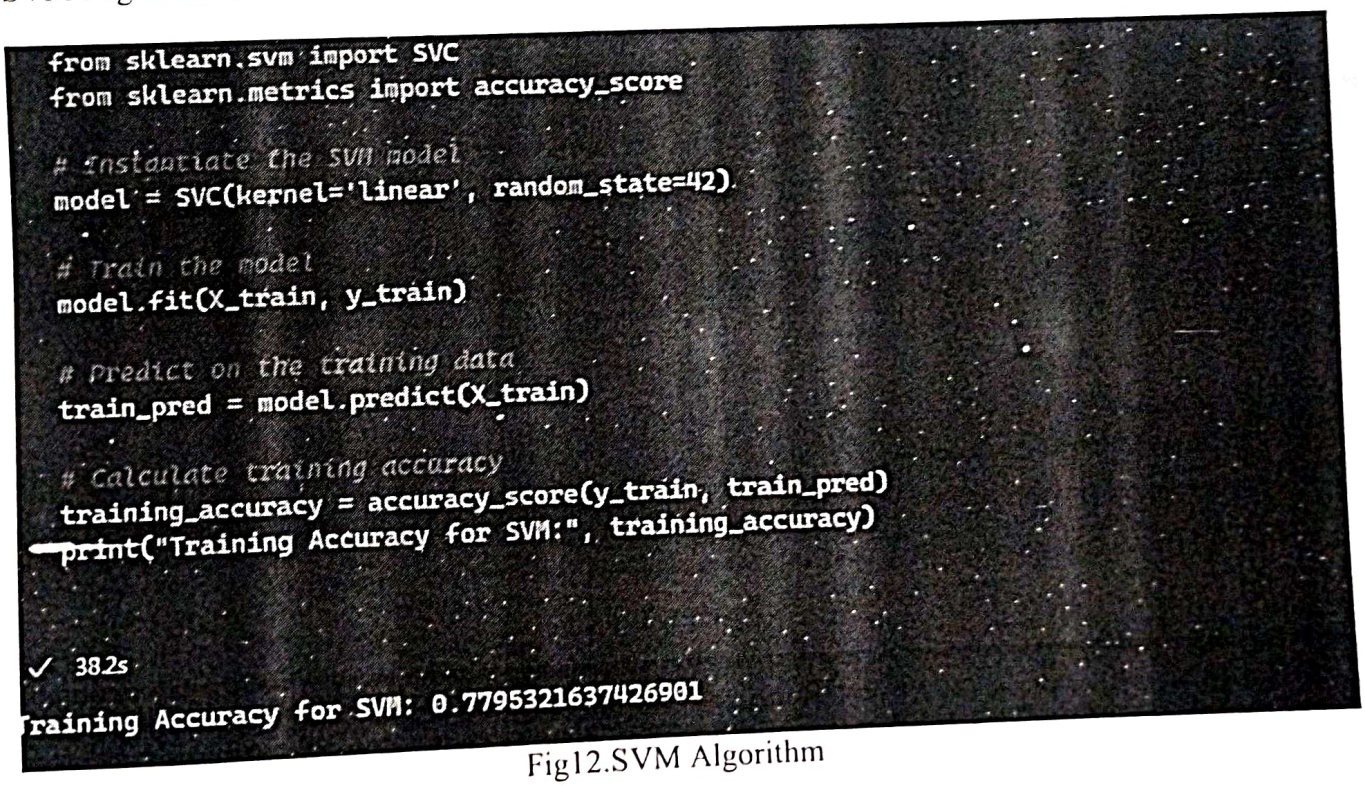


Fig12.SVM Algorithm

8. RESULTS

The sentiment analysis project successfully classified Flipkart product reviews as positive, neutral or negative using a Decision Tree Classifier, achieving an impressive training accuracy of 96%. This high accuracy demonstrates the model's ability to capture patterns in the data. Word clouds for positive, neutral and negative reviews provided insights into frequently used sentiment-related vocabulary. However, the absence of testing on unseen data raises concerns about potential overfitting, suggesting the need for further evaluation using metrics like test accuracy, precision, and recall. The TF-IDF vectorization technique effectively transformed text into numerical features. While the model performed well, class imbalance in the dataset might have influenced predictions, necessitating techniques such as oversampling or under-sampling to address this. Comparing the Decision Tree with other algorithms, including Logistic Regression, SVM and Random Forest could help identify the most suitable approach for this task. Despite these limitations, the project highlights the utility of machine learning in sentiment analysis and provides a foundation for further refinements.

Model	Test Accuracy
Logistic Regression	77%
Decision Tree Classifier	96%
SVM Algorithm	57%
Random Class Classifier	77%

Table 1: Resulting Accuracy of various Algorithms

9.CONCLUSION

The sentiment analysis of Flipkart reviews using a Decision Tree classifier demonstrated the ability to classify customer feedback as either positive, neutral or negative based on review ratings. Data preprocessing, including punctuation removal, lowercase conversion, and stopword elimination, enhanced the text quality for analysis. Word clouds visually showcased the most frequent terms in positive, neutral and negative reviews, revealing key customer sentiments. The TF-IDF vectorizer transformed text into numerical features, allowing the model to learn sentiment patterns effectively. The Decision Tree classifier achieved good accuracy on the training data, but further validation with unseen data is necessary for generalization. A custom function was developed to predict the sentiment of new reviews, demonstrating the model's real-world applicability. For future improvements, exploring advanced algorithms, cross-validation, and contextual embeddings like BERT could enhance the model's performance and accuracy.

10.Bibliography

[1] Shivaprasad, T. K., and Jyothi Shetty. "Sentiment analysis of product reviews: A review." 2017 International conference on inventive communication and computational technologies (ICICCT). IEEE, 2017.

[2] Mukherjee, Subhabrata, and Pushpak Bhattacharyya. "Feature specific sentiment analysis for product reviews." International conference on intelligent text processing and computational linguistics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[3] Chauhan, Chhaya, and Smriti Sehgal. "Sentiment analysis on product reviews." 2017 International Conference on Computing, Communication and Automation (ICCCA). IEEE, 2017.

[4] Kanakamedala, Venkata Chaitanya, Sonal Hukampal Singh, and Rishitha Talavani. "Sentiment analysis of online customer reviews for handicraft product using machine learning: A case of Flipkart." 2023 International Conference for Advancement in Technology (ICONAT). IEEE, 2023.

[5] Nasar, Mohammad, Mohammad Abu Kausar, and Ajeet Singh. "Sentiment classification based on machine learning approaches in flipkart product reviews." Artificial Intelligence and Information Technologies. CRC Press, 2024. 400-406.

[6] Vyas, Vishal, and V. Uma. "Approaches to sentiment analysis on product reviews." *Sentiment Analysis and Knowledge Discovery in Contemporary Business*. IGI global, 2019. 15-30.

[7] Agarwal, Basant, et al. "Machine learning approach for sentiment analysis." *Prominent feature extraction for sentiment analysis* (2016): 21–45.

[8] Neethu, M. S., and R. Rajasree. "Sentiment analysis in twitter using machine learning techniques." *2013 fourth international conference on computing, communications and networking technologies (ICCCNT)*. IEEE, 2013.

[9] Hasan, Ali, et al. "Machine learning-based sentiment analysis for twitter accounts." *Mathematical and computational applications* 23.1 (2018): 11.

[10] Ahmad, Munir, et al. "Machine learning techniques for sentiment analysis: A review." *Int. J. Multidiscip. Sci. Eng* 8.3 (2017): 27.