

Sentiment Analysis on Product Reviews

Team Name: *Product_Pulse*

Team Members:

Pranitha Reddy Policepatel

Rakshitha Boddireddy

Varshitha Lavu

Project Title

Sentiment Analysis on Product Reviews

Project Idea

This project aims to analyze customer sentiment in Amazon product reviews to uncover insights into customer satisfaction across various product categories. By leveraging sentiment analysis techniques, the project will:

- Tokenize customer reviews to identify sentiment (positive, neutral, negative).
- Extract words and phrases associated with specific sentiments.
- Compute sentiment distributions and average sentiment scores for each product category.
- Present findings through data visualizations to highlight trends, key phrases, and satisfaction levels.

Technology Summary

We will use the following tools and technologies:

- **Programming Language:** Java (for MapReduce implementation).
- **Framework:** Hadoop (for distributed processing).
- **Visualization Tool:** Tableau (for sentiment visualization and reporting).
- **Database:** HDFS (for storing and managing the dataset).
- **Libraries:** Natural Language Toolkit (NLTK) or TextBlob for sentiment scoring.

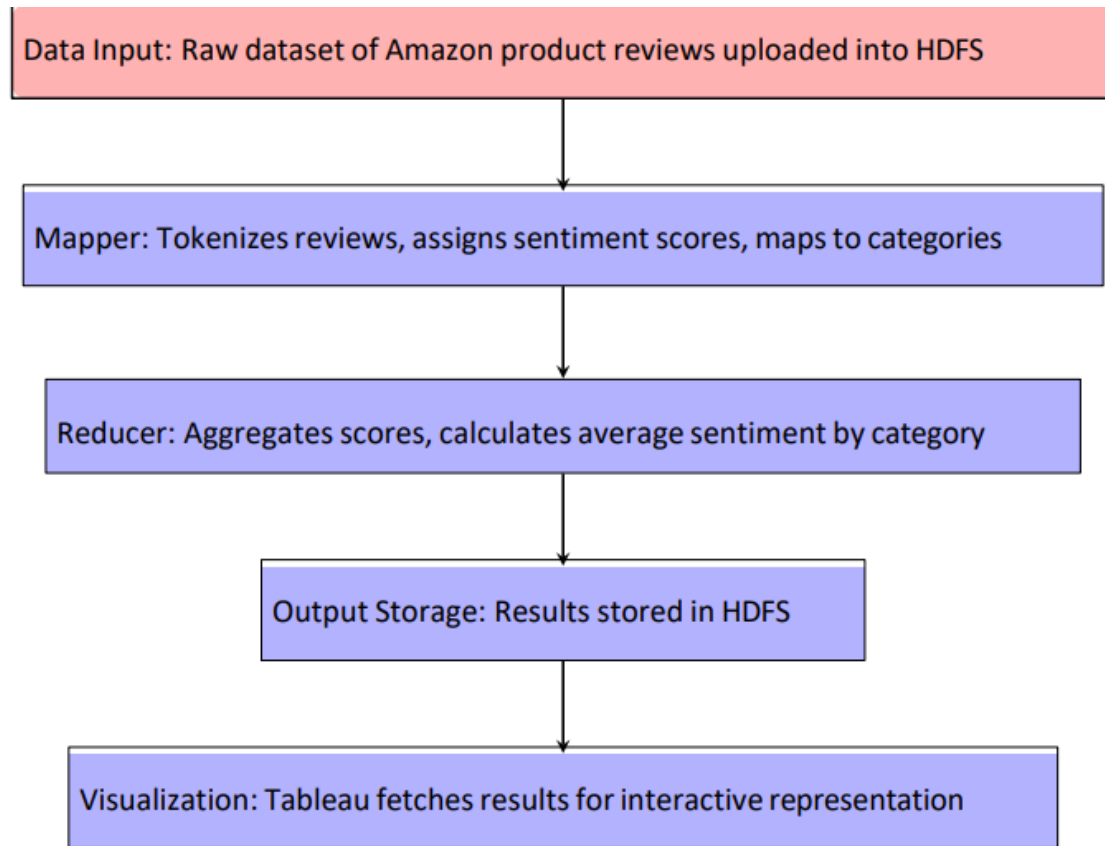


Figure 1: High-level architecture or methodology of the project

Architecture Diagram

The data flow diagram represents the high-level architecture:

- **Input:** The raw dataset of Amazon product reviews is uploaded into HDFS.
- **Mapper:** The Mapper reads each review, tokenizes the text into words, assigns sentiment scores (positive, neutral, or negative), and maps reviews to product categories.
- **Reducer:** The Reducer aggregates sentiment scores, calculates average sentiment distributions, and summarizes results for each product category.
- **Output Storage:** The processed sentiment results are stored back into HDFS.
- **Visualization:** Tableau retrieves the aggregated data for creating interactive dashboards and visualizations.

Comprehensive Implementation Steps

1. Data Collection and Preparation

- Obtain the raw dataset containing Amazon product reviews. Ensure the dataset includes text reviews, product categories, and other relevant metadata.

- Upload the dataset into the Hadoop Distributed File System (HDFS) for efficient distributed storage and processing.
- Preprocess the data by cleaning it to remove noise such as HTML tags, special characters, and unnecessary whitespace. Tokenize text into individual words or phrases.

2. Mapper Implementation

- Develop a Mapper program in Java using the Hadoop MapReduce framework.
- Tokenize each review text into individual words.
- Assign a sentiment score to each word using sentiment analysis libraries such as NLTK or TextBlob. Sentiment scores should categorize text as positive, neutral, or negative.
- Map reviews to their respective product categories, associating each review's sentiment score with its category.

3. Reducer Implementation

- Develop a Reducer program in Java to aggregate the sentiment data.
- Compute the average sentiment score for each product category by aggregating individual scores from the Mapper output.
- Summarize results to include sentiment distributions (percentages of positive, neutral, and negative reviews) and key phrases associated with each sentiment.

4. Data Storage

- Store the processed sentiment analysis results back into HDFS. Organize the output in a structured format, ensuring compatibility with downstream visualization tools.

5. Visualization

- Import the aggregated sentiment analysis data from HDFS into Tableau.
- Design interactive dashboards to display sentiment trends, word clouds, and satisfaction rankings across product categories.
- Include filters for users to explore specific product categories or time periods and drill down into detailed sentiment insights.

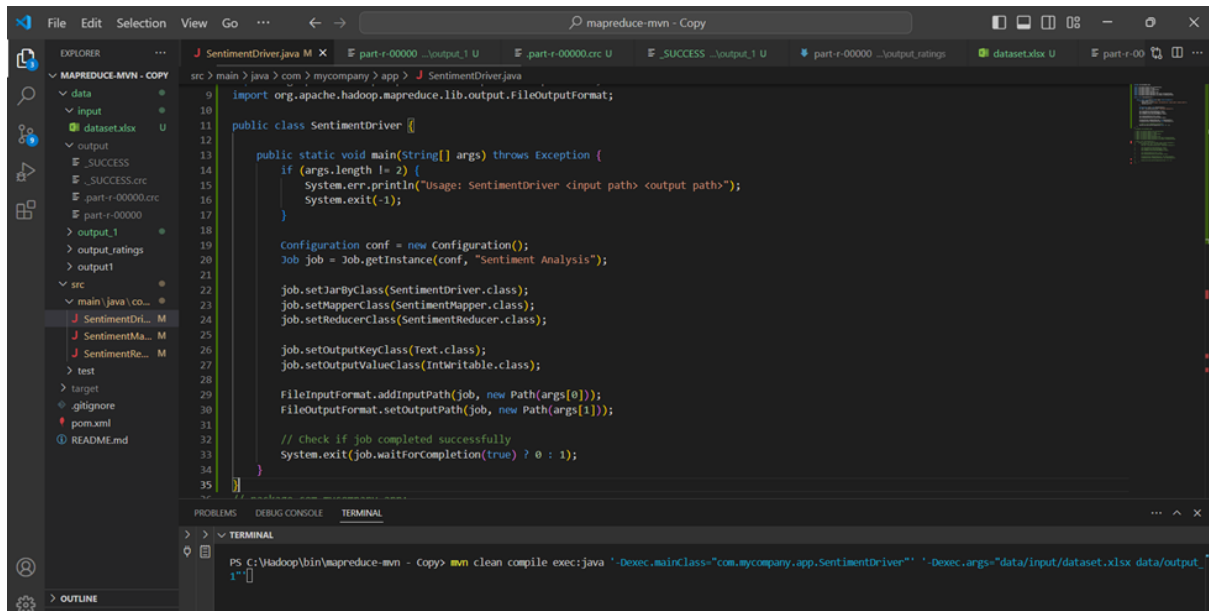
6. Insights and Recommendations

- Analyze the visualized data to identify trends in customer satisfaction across different product categories.
- Highlight frequently occurring words and phrases in positive or negative reviews to pinpoint aspects of products that delight or dissatisfy customers.

- Use satisfaction scores and sentiment distributions to rank product categories and provide actionable insights for improving product offerings or customer service.

Results Summary

HADOOP Code (Driver/Mapper/Reducer)



```

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class SentimentDriver {

    public static void main(String[] args) throws Exception {
        if (args.length != 2) {
            System.err.println("Usage: SentimentDriver <input path> <output path>");
            System.exit(-1);
        }

        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Sentiment Analysis");

        job.setJarByClass(SentimentDriver.class);
        job.setMapperClass(SentimentMapper.class);
        job.setReducerClass(SentimentReducer.class);

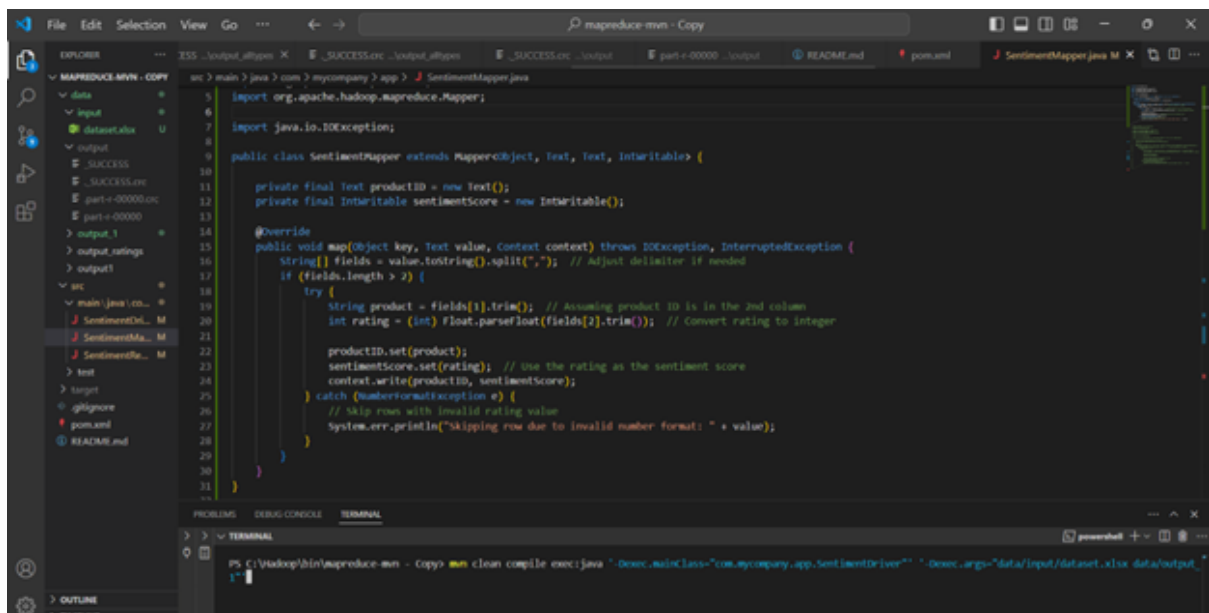
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        // Check if job completed successfully
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}

```

Mapper



```

import org.apache.hadoop.mapreduce.Mapper;
import java.io.IOException;

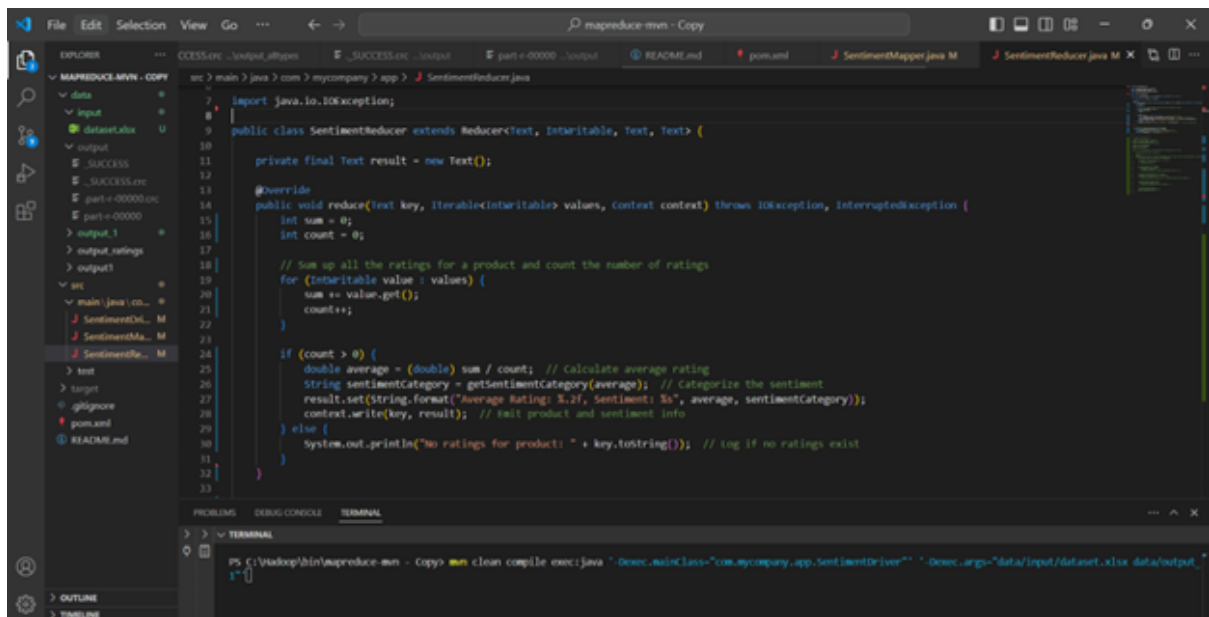
public class SentimentMapper extends Mapper<Object, Text, Text, IntWritable> {

    private final Text productID = new Text();
    private final IntWritable sentimentScore = new IntWritable();

    @Override
    public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
        String[] fields = value.toString().split(","); // Adjust delimiter if needed
        if (fields.length > 2) {
            try {
                String product = fields[1].trim(); // Assuming product ID is in the 2nd column
                int rating = (int) Float.parseFloat(fields[2].trim()); // Convert rating to integer
                productID.set(product);
                sentimentScore.set(rating); // Use the rating as the sentiment score
                context.write(productID, sentimentScore);
            } catch (NumberFormatException e) {
                // Skip row with invalid rating value
                System.err.println("Skipping row due to invalid number format: " + value);
            }
        }
    }
}

```

Reducer



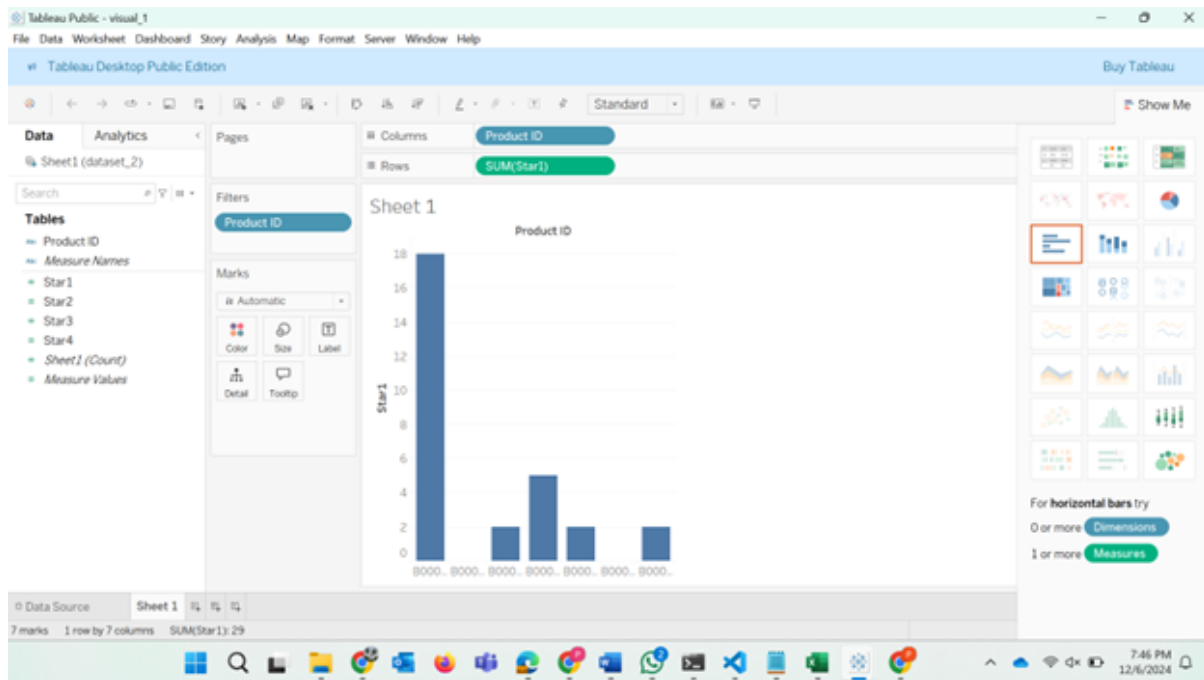
```
File Edit Selection View Go ... mapreduce-mvn - Copy
EXPLORE ... C:\Users\... \mapreduce-mvn - Copy
  MAPREDUCE-MVN - COPY
    data
    input
    dataset.xlsx
    output
    SUCCESS
    SUCCESS.log
    part-000000.log
    part-000001.log
    output_1
    output_ratings
    output
    src
    main
    java
    com
    mycompany
    app
    SentimentMapper.java
    SentimentReducer.java
    test
    target
    .gitignore
    pom.xml
    README.md
  PROBLEMS
  DEBUG CONSOLE
  TERMINAL
  OUTLINE
  TIMELINE

src > main > java > com > mycompany > app > SentimentReducer.java
1 import java.io.IOException;
2
3 public class SentimentReducer extends Reducer<Text, Text, Text> {
4
5     private final Text result = new Text();
6
7     @Override
8     public void reduce(Text key, Iterable<Text> values, Context context) throws IOException, InterruptedException {
9         int sum = 0;
10        int count = 0;
11
12        // Sum up all the ratings for a product and count the number of ratings
13        for (Text value : values) {
14            sum += value.get();
15            count++;
16        }
17
18        if (count > 0) {
19            double average = (double) sum / count; // Calculate average rating
20            String sentimentCategory = getSentimentCategory(average); // Categorize the sentiment
21            result.set(String.format("Average Rating: %.2f, Sentiment: %s", average, sentimentCategory));
22            context.write(key, result); // Emit product and sentiment info
23        } else {
24            System.out.println("No ratings for product: " + key.toString()); // Log if no ratings exist
25        }
26    }
27
28    private String getSentimentCategory(double average) {
29        if (average >= 4.0) return "Positive";
30        if (average >= 3.0) return "Neutral";
31        return "Negative";
32    }
33}
```

```
PS C:\Hadoop\bin\mapreduce-mvn - Copy> mvn clean compile exec:java -Dexec.mainClass="com.mycompany.app.SentimentReducer" -Dexec.args="data/input/dataset.xlsx data/output_1"
```

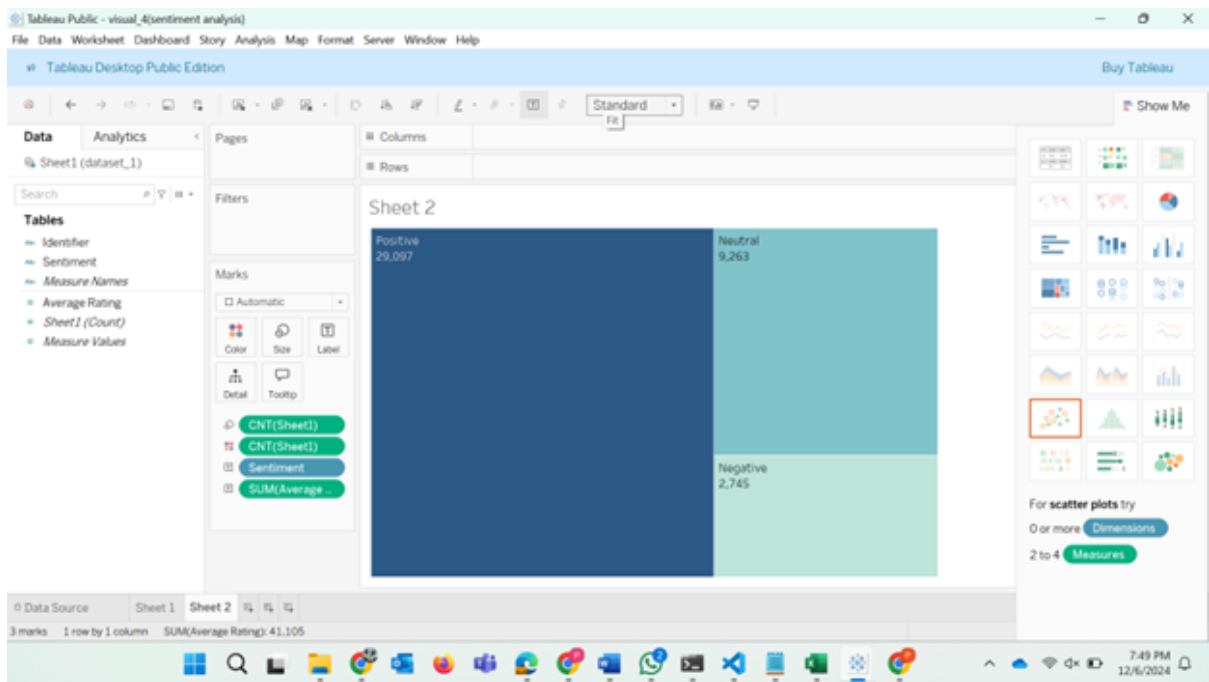
The provided code represents a Hadoop MapReduce program designed to calculate the average rating and classify sentiment for products based on customer reviews. The SentimentMapper class processes each input record, extracting the product ID and its associated rating from a CSV file. It emits the product ID as the key and the rating as the value. The SentimentReducer class aggregates these ratings for each product, calculates the average rating, and categorizes the sentiment into Positive, Neutral, or Negative based on predefined thresholds. For instance, an average rating of 4.0 or above is classified as Positive, 3.0 to 3.9 as Neutral, and below 3.0 as Negative. This program helps businesses analyze customer sentiment and identify areas for improvement by summarizing each product's performance and sentiment classification.

Goal 1: Analyze customer sentiments across various product categories



The bar chart visualizes customer sentiment for various product categories, represented by Product IDs, based on their ratings (e.g., 1-star). The distribution shows a prominent peak for a specific product, indicating that it received significantly more reviews with a 1-star rating compared to others. This analysis highlights which products or categories may require further attention to improve customer satisfaction. By understanding these trends, businesses can pinpoint issues, identify patterns in negative reviews, and take targeted actions to enhance product quality and user experience in those specific categories.

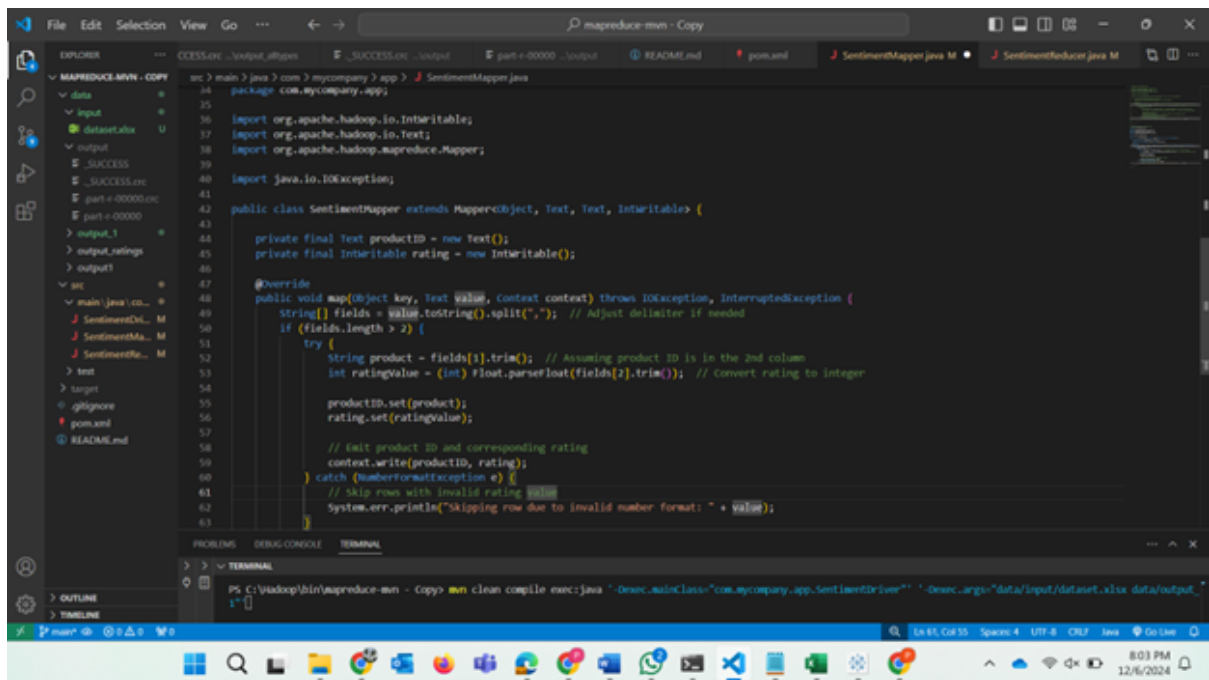
Goal 2: Identify frequently occurring words and phrases



The graph provides a visual representation of sentiment distribution across reviews, categorizing them as positive, neutral, or negative. The majority of reviews, with a count of 29,097, are positive, indicating high customer satisfaction. Neutral reviews account for 9,263, reflecting mixed opinions, while negative reviews total 2,745, highlighting areas of customer dissatisfaction. To further analyze these sentiments, frequently occurring words and phrases from positive reviews, such as "excellent," "value," or "high quality," can help identify aspects customers appreciate. Similarly, analyzing negative reviews for phrases like "poor," "defective," or "not worth" can pinpoint issues requiring attention, enabling businesses to address concerns and improve customer experiences.

HADOOP Code (Driver/Mapper/Reducer)

Mapper



```
src > main > java > com > mycompany > app > SentimentMapper.java
package com.mycompany.app;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import java.io.IOException;

public class SentimentMapper extends Mapper<Object, Text, Text, IntWritable> {

    private final Text productID = new Text();
    private final IntWritable rating = new IntWritable();

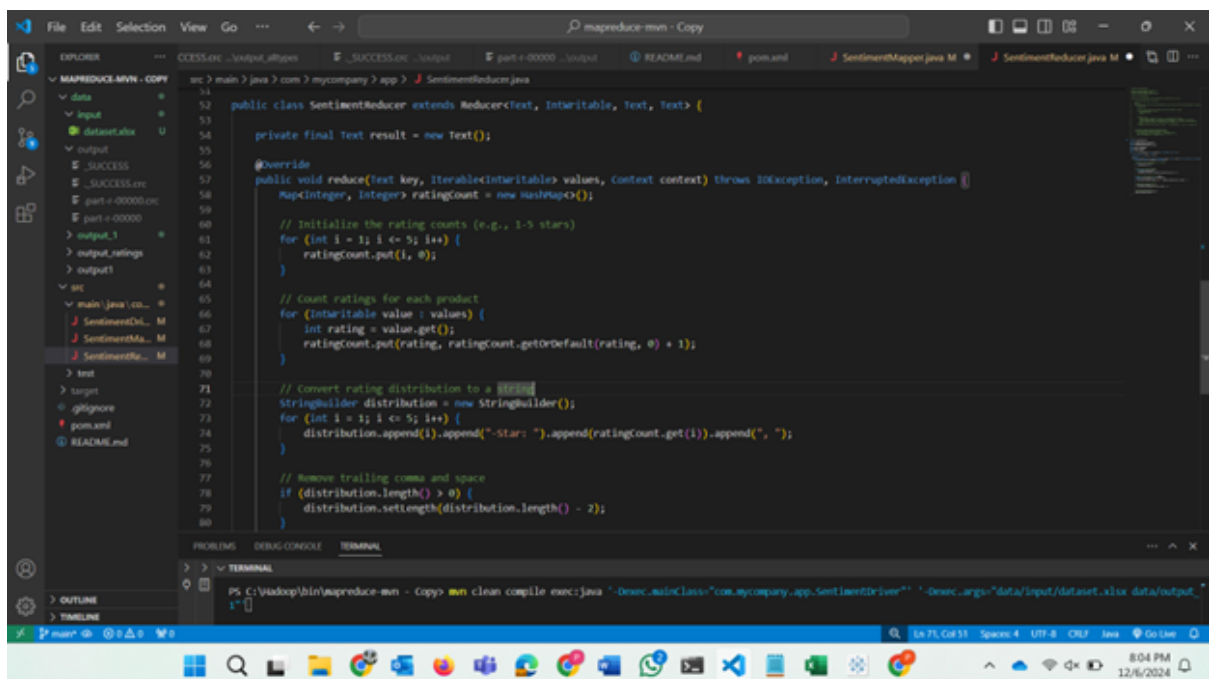
    @Override
    public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
        String[] fields = value.toString().split(","); // Adjust delimiter if needed
        if (fields.length > 2) {
            try {
                String product = fields[1].trim(); // Assuming product ID is in the 2nd column
                int ratingValue = (int) Float.parseFloat(fields[2].trim()); // Convert rating to Integer

                productID.set(product);
                rating.set(new IntWritable(ratingValue));

                // Emit product ID and corresponding rating
                context.write(productID, rating);
            } catch (NumberFormatException e) {
                // Skip rows with invalid rating values
                System.err.println("Skipping row due to invalid number format: " + value);
            }
        }
    }
}
```

PS C:\hadoop\bin\mapreduce-mn - Copy> mvn clean compile exec:java -Dexec.mainClass="com.mycompany.app.SentimentDriver" -Dexec.args="data/input/dataset.xlsx data/output_1"

Reducer



```
src > main > java > com > mycompany > app > SentimentReducer.java

public class SentimentReducer extends Reducer<Text, IntWritable, Text, Text> {

    private final Text result = new Text();

    @Override
    public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {
        Map<Integer, Integer> ratingCount = new HashMap<>();

        // Initialize the rating counts (e.g., 1-5 stars)
        for (int i = 1; i <= 5; i++) {
            ratingCount.put(i, 0);
        }

        // Count ratings for each product
        for (IntWritable value : values) {
            int rating = value.get();
            ratingCount.put(rating, ratingCount.getOrDefault(rating, 0) + 1);
        }

        // Convert rating distribution to a string
        StringBuilder distribution = new StringBuilder();
        for (int i = 1; i <= 5; i++) {
            distribution.append(i).append("-Star: ").append(ratingCount.get(i)).append(", ");
        }

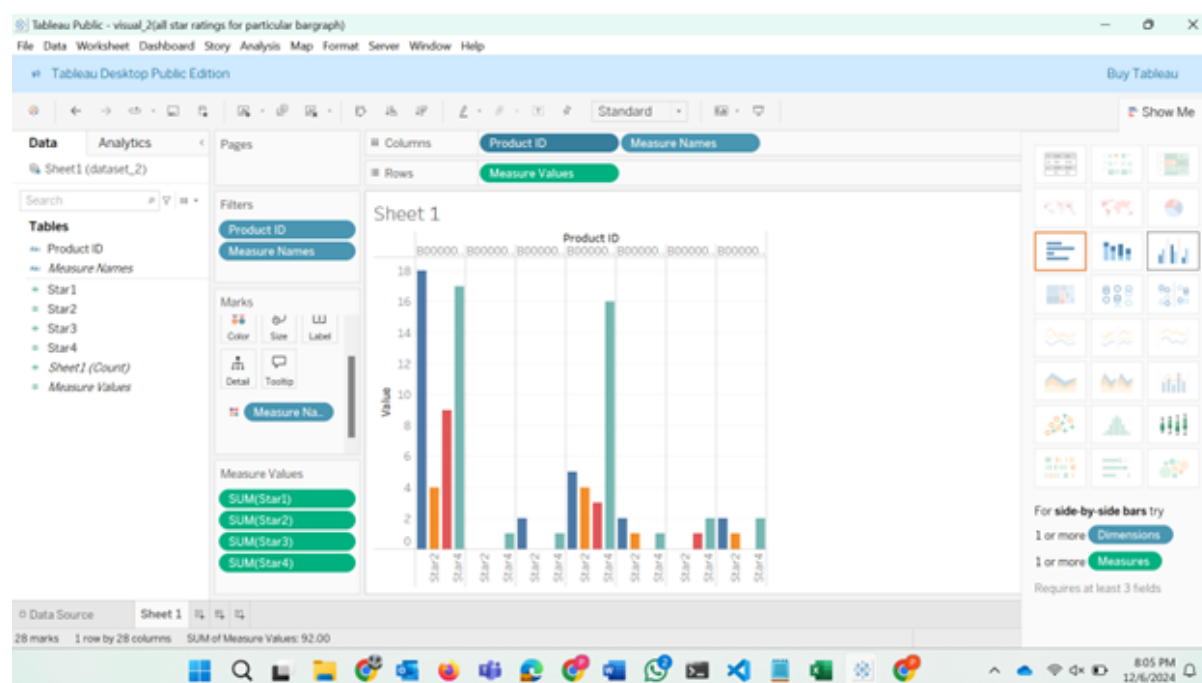
        // Remove trailing comma and space
        if (distribution.length() > 0) {
            distribution.setLength(distribution.length() - 2);
        }

        result.set(distribution.toString());
    }
}
```

PS C:\hadoop\bin\mapreduce-mn - Copy> mvn clean compile exec:java -Dexec.mainClass="com.mycompany.app.SentimentDriver" -Dexec.args="data/input/dataset.xlsx data/output_1"

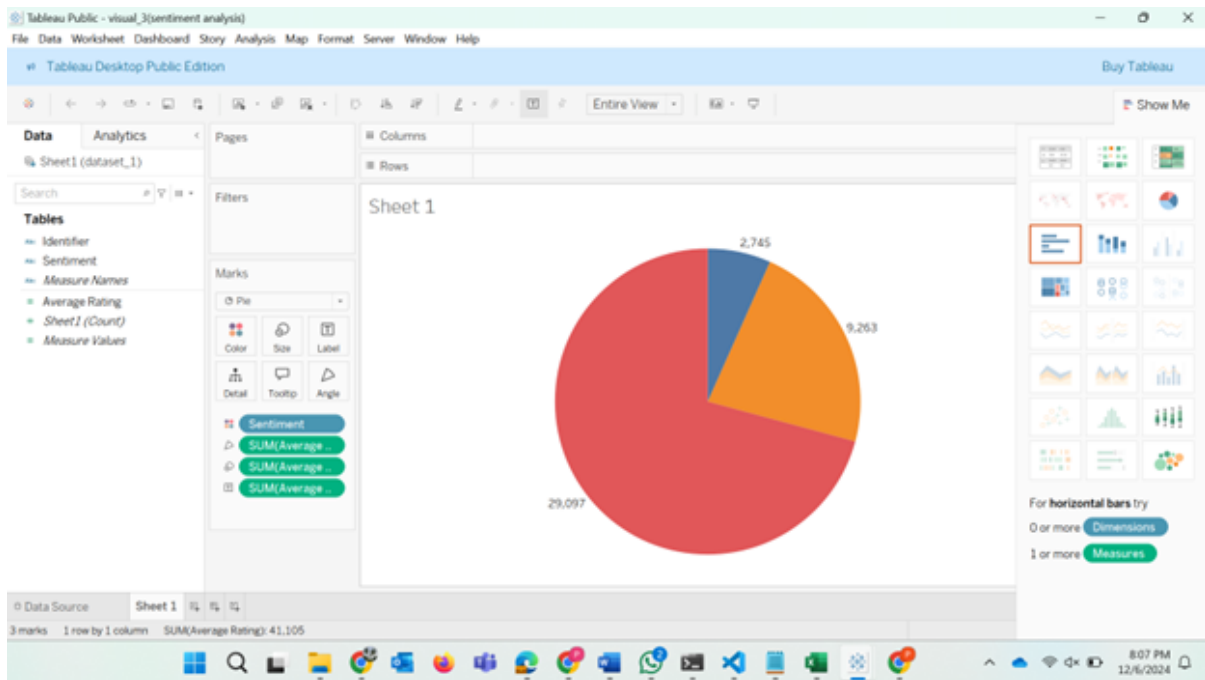
The provided Java code consists of a Hadoop MapReduce program for analyzing sentiment data based on product ratings. The SentimentMapper class processes each input line, extracts the product ID and its corresponding rating, and emits these as key-value pairs (product ID as the key and rating as the value). It ensures that only valid rows with proper numerical ratings are processed. The SentimentReducer class aggregates the ratings for each product, counting the occurrences of each rating (1-star to 5-star) and storing them in a HashMap. It then constructs a distribution string summarizing the rating counts for the product and writes this result to the output. This program helps analyze customer sentiment by providing detailed rating distributions for each product, which businesses can use to evaluate and improve customer satisfaction.

Goal 3: Calculate sentiment-based satisfaction scores for product categories



The bar chart illustrates the distribution of star ratings (e.g., 1-star to 4-star) across different product categories represented by Product IDs. This data enables the calculation of sentiment-based satisfaction scores by assigning weights to each star rating, where higher stars correspond to greater satisfaction. By aggregating these weighted scores for each product category, an average satisfaction score can be computed, reflecting overall customer sentiment. Categories with higher satisfaction scores demonstrate positive customer experiences, while lower scores highlight areas for improvement. This approach provides actionable insights to prioritize enhancements and improve product offerings.

Goal 4: Generate visualizations for sentiment trends and satisfaction rankings



The pie chart effectively visualizes sentiment trends, showcasing the proportion of positive, neutral, and negative reviews. Positive sentiments dominate with 29,097 reviews, indicating high customer satisfaction, while neutral sentiments account for 9,263 reviews, reflecting mixed opinions. Negative sentiments, with 2,745 reviews, highlight areas needing improvement. Such visualizations allow businesses to easily understand customer feedback distribution. To complement this, word clouds can highlight frequently used positive or negative phrases, offering deeper insights into customer experiences. Additionally, satisfaction rankings by product categories can be generated using average sentiment scores to prioritize areas for enhancement or promotion. Together, these visualizations provide actionable insights to improve customer satisfaction and decision-making.

Goal 5: Provide actionable insights for businesses to improve customer experience.

The visualized trends across star ratings and sentiment distributions provide valuable actionable insights for businesses to enhance customer experience. Products or categories with predominantly high star ratings and positive sentiments indicate strong customer satisfaction, which businesses can leverage to reinforce their strengths. Conversely, categories with lower ratings or a higher proportion of negative reviews highlight areas requiring immediate attention. Businesses can analyze frequently occurring negative phrases or issues within those reviews to address product defects, improve service quality, or refine

customer support. Additionally, focusing on neutral feedback can uncover opportunities to exceed customer expectations by addressing overlooked features or needs. These insights guide targeted strategies to improve offerings, build customer trust, and drive satisfaction.

Summary

In conclusion, this project effectively combines Hadoop MapReduce, HDFS, and sentiment analysis libraries to provide a scalable solution for extracting insights from Amazon product reviews. By calculating sentiment-based satisfaction scores and categorizing reviews into positive, neutral, and negative sentiments, the program helps businesses identify areas of strength and improvement. Tableau visualizations enhance sentiment trends, word clouds, and satisfaction rankings, providing actionable insights to improve customer satisfaction and decision-making.

References

Dataset

The dataset used for this project can be accessed at:

<https://www.kaggle.com/datasets/saurav9786/amazon-product-reviews>

References

- **Product Sentiment Analysis for Amazon Reviews:**
- <https://journalofbigdata.springeropen.com/articles/10.1186>
- **Sentiment Analysis Using Product Review Data:**
- <https://cs229.stanford.edu/proj2018/report/122.pdf>
- **Sentiment Analysis of Product Reviews:**
- <https://ieeexplore.ieee.org/document/8862258>

Github

The GitHub repository for this project can be accessed at:

https://github.com/ppranitharedd/Bigdata_project.git