Project Report

On

E-Commerce Sentiment Analysis

Submitted in partial fulfillment for the award of

Post Graduate Diploma in Big Data Analytics (PG-DBDA)

From Know-IT(Pune)

Guided by:

Anay Tamhankar

Prasad Deshmukh

Submitted By:

Ajay Yadav (230343025003)

Nomesh Verma (230343025035)

Swarup Shinde (230343025044)

Sanket Zargad (230343025057)

CERTIFICATE

TO WHOMSOEVER IT MAY CONCERN

This is to certify that

Ajay Yadav (230343025003)

Nomesh Verma (230343025035)

Swarup Shinde (230343025044)

Sanket Zargad (230343025057)

Have successfully completed their project on

E-Commerce Sentiment Analysis

Under the guidance of **Anay Tamhankar sir and Prasad Deshmukh Sir**

ACKNOWLEDGEMENT

This project "E-Commerce Sentiment Analysis" was a great learning experience for us and we are submitting this work to CDAC Know-IT (Pune).

We all are very glad to mention the name of Anay Tamhankar Sir and Prasad Deshmukh Sir for his valuable guidance to work on this project. His guidance and support helped us to overcome various obstacles and intricacies during the course of project work.

We are highly grateful to Mr. Vaibhav Inamdar Manager (Know-IT), C-DAC, for his guidance and support whenever necessary while doing this course Post Graduate Diploma in Big Data

Analytics (PG-DBDA) through C-DAC ACTS, Pune.

Our most heartfelt thanks go to Mrs. Bakul Joshi (Course Coordinator-DBDA) who gave all the required support and kind coordination to provide all the necessities like required hardware, internet facility and extra Lab hours to complete the project and throughout the course up to the last day here in C-DAC Know-IT, Pune.

From:

Ajay Yadav (230343025003)

Nomesh Verma (230343025035)

Swarup Shinde (230343025044)

Sanket Zargad (230343025057)

TABLE OF CONTENTS

Abstract

This project aims to analyze sentiment of customers with the help of Deep Machine Learning (NLP). This can help business to understand customer's response and will help to make decisions effectively.

The project will involve collecting Amazon product data from various sources, such as Kaggle. We processed ETL on this Dataset in Spark.

Next, the project will use machine learning algorithms such as Deep Machine Learning (NLP), to analyze sentiment for identifying customer's responses. The project will use Apache Spark, to analyze large datasets efficiently.

The project will also include data visualization techniques to present the sentiment results in an intuitive and easy-to-understand format. This will help business to quickly identify sentiment of customer and take effective decision.

1. INTRODUCTION

- Sentiment analysis using Deep Machine Learning (NLP) and Spark involves building a sentiment analysis and to analyze customer opinion effectively.
- This Sentiment analysis can be done by analyzing text review of customers data and building a model that can predict what customer is texting about project.
- The project will use machine learning algorithms such as NLP and Sentiment Analyzer to analyze the data and identify customers response to the product. The project will use Apache Spark to analyze large datasets efficiently.
  - For Sentiment Analysis, features such as id, name, review text, sentiment, review rating has been used and applied NLP on features.

Datasets and features:

Data used in the project is structured in nature. It was collected from www.kaggle.com. The main goal of the analysis is to understand sentiment of customer's review text. This research uses Deep Machine Learning (NLP) and Spark.

2. SYSTEM REQUIREMENTS

Hardware Requirements:

- Platform – Windows
- RAM – 8 GB of RAM,
- Peripheral Devices – Mouse, Keyboard, Monitor
- A network connection for data recovering over network.

Software Requirements:

- Python 3
- PySpark
- Tableau
- GoogleColab and Jupyter
- OS – Windows

FUNCTIONAL REQUIREMENTS

(1) Python 3:

- Python is a general purpose and high-level programming language.
- It is use for developing desktop GUI applications, websites, and web applications.
- Python allows to focus on core functionality of the application by taking care of common programming tasks.
- Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small Talk, and Unix shell and other scripting languages.

(2) Apache Spark:

- What is Spark: Apache Spark is an open-source distributed computing system designed forprocessing large volumes of data.
- Key Features: Spark provides several key features that make it well-suited for processing big data, including in-memory processing, support for various data sources and formats, fault- tolerance, and scalability.
- Spark also provides a range of APIs, including SQL, streaming, machine learning, and graph processing, making it a versatile platform for a wide range of use cases.

(3) Tableau:

- Data visualization is the graphical representation of information and data.
- It helps create interactive elements like charts, graphs, and maps, data visualization tools providean accessible way to see and understand trends, outliers, and patterns in data.
- Tableau is widely used for Business Intelligence but is not limited to it.
- It helps create interactive graphs and charts in the form of dashboards and worksheets to gainbusiness insights.

- All of this is made possible with gestures as simple as drag and drop.

(4) Pyspark:

- PySpark is the Python library and API for Apache Spark, an open-source, distributed computing framework designed to process large datasets efficiently across clusters of computers.
- It provides a powerful and flexible platform for big data processing, machine learning, graph computation, and more.
- PySpark enables developers and data scientists to work with data in a distributed and parallel manner, leveraging the capabilities of Spark's underlying architecture.

Why PySpark:

PySpark's distributed computing, scalability, in-memory processing, machine learning capabilities, and integration with the larger Spark ecosystem make it a preferred choice for sentiment analysis tasks that involve processing and analyzing large volumes of text data.

Data Cleaning Process:

Fig: Data Cleaning Process

Data preprocessing is a crucial step in a sentiment analysis project as it helps clean and prepare the text data for analysis. Here's a general outline of the data preprocessing steps we take for our sentiment analysis project using PySpark:

(1) Data Collection and Understanding:

- Collect the raw reviews data from kaggle.
- Understand the structure of the data, including the format, columns, and labels if available

(2) Text Cleaning:

- Remove any irrelevant information, such as URLs, hashtags, and mentions.
- Remove special characters, punctuation, and numbers that may not contribute to sentiment analysis.
- Convert the text to lowercase to ensure consistent analysis.

(3) Tokenization:

- Tokenize the text into individual words or tokens.
- PySpark's Tokenizer can be used to split text into tokens.

(4) Stop Word Removal:

- Remove common stop words (e.g., "and," "the," "is") that do not carry significant sentiment information.
- PySpark's "StopWordsRemover" can help remove stop words from the tokenized text.

(5) Stemming or Lemmatization:

- Reduce words to their base or root form to normalize the vocabulary.
- PySpark's SnowballStemmer or external libraries like NLTK can be used for stemming.

(6) Handling Negations and Contractions:

- Identify and handle negations (e.g., "not good" should be treated differently from "good").
- Expand contractions (e.g., "I'm" to "I am") to ensure proper tokenization.

(7) Exploratory Data Analysis (EDA):

- Perform EDA to gain insights into the distribution of sentiments, most frequent words, and patterns in the text data.

3. SYSTEM ARCHITECTURE

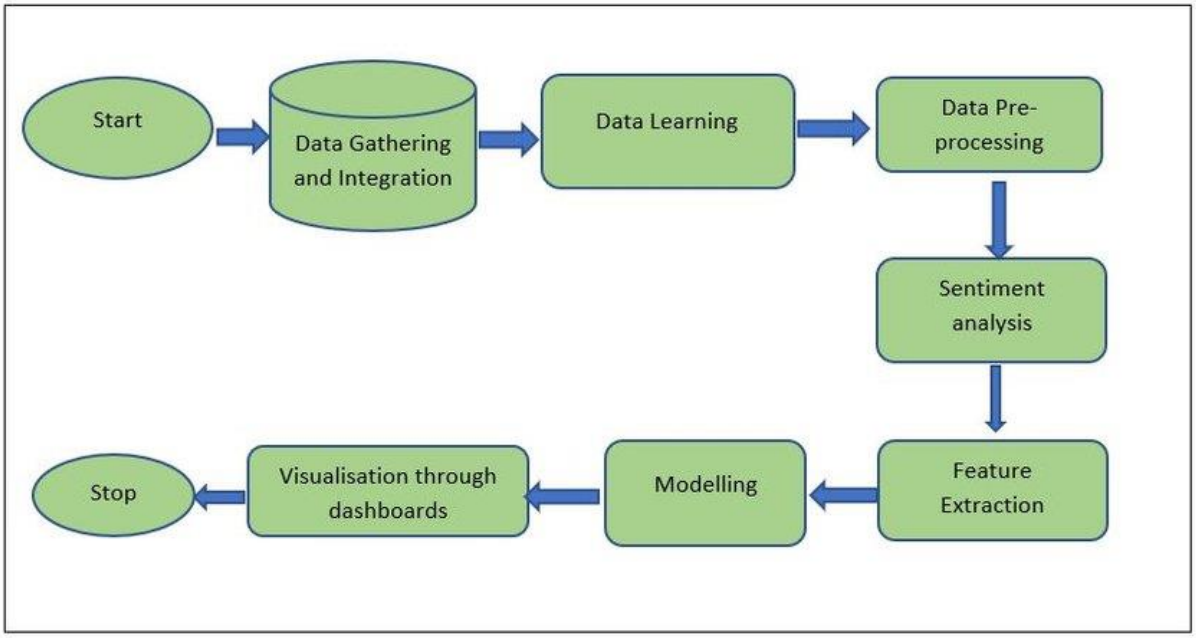

Fig: System Architecture of Sentiment Analysis

4. METHODOLOGY

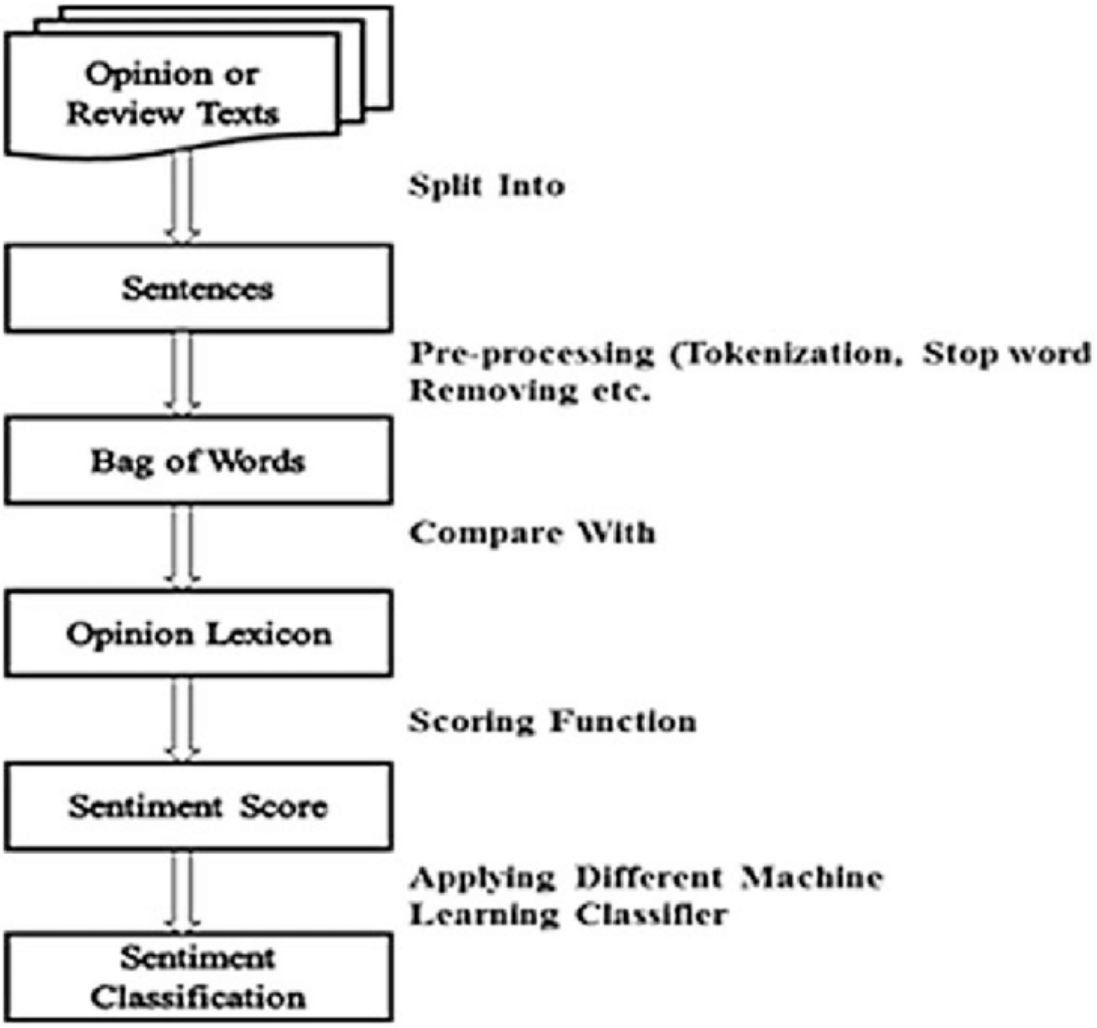

Fig: Sentiment Analysis Methodology

5.MACHINE LEARNING ALGORITHMS

In this project we apply various types of Deep Machine Learning (NLP) such as NLTK, Sentiment Analyzer. During the implementation we analyze the sentiment of customer review text.

Machine learning is the research that explores the development of algorithms that can learn from data and provide predictions based on it. Works that study flight systems are increasing the usage of machine learning methods. The methods used are NLTK, Sentiment Analyzer.

1. NLTK:

NLTK (Natural Language Toolkit) is a Python library integral to deep machine learning for natural language processing (NLP). It provides a comprehensive suite of tools for tasks like tokenization, stemming, part-of-speech tagging, and sentiment analysis. With its extensive collection of linguistic data and algorithms, NLTK simplifies complex NLP processes, enabling researchers and developers to build and train deep learning models that understand and generate human language effectively.

Pros:

- Comprehensive NLP Toolbox: NLTK offers a broad set of tools for tasks like tokenization, tagging, and parsing, streamlining text analysis. Makes no prior assumption of the data
- Rich Linguistic Resources: NLTK includes a collection of linguistic resources like corpora (text datasets) and lexical resources (wordlists, thesauri). These resources are valuable for training and testing NLP models and conducting linguistic research.

Cons:

- Performance: NLTK can sometimes be slower compared to more optimized libraries like spaCy, especially when dealing with large text datasets. Certain tasks might take longer to execute, which could be a concern for real-time applications or processing massive amounts of data.

- Resource Intensive: Some NLTK operations, such as parsing and complex feature extraction, can be memory-intensive and may not scale well for very large datasets or constrained environments.

- Complexity: NLTK's extensive functionality can sometimes lead to a steeper learning curve, particularly for newcomers to NLP or programming. The wide range of options might make it overwhelming for those looking for quick and simple solutions.

2. Sentiment Analysis:

The 'SentimentIntensityAnalyzer' is a component of the Natural Language Toolkit (NLTK) library in Python. It is designed for sentiment analysis, a subfield of natural language processing (NLP) that aims to determine the sentiment or emotional tone of a piece of text. The `SentimentIntensityAnalyzer` utilizes a **lexicon-based** approach to assign sentiment polarity scores to text, indicating the degree of **positivity, negativity,** or **neutrality** in the text's sentiment.

Key Features and Functions:

1. Lexicon-Based Approach: The SentimentIntensityAnalyzer uses a pre-built lexicon called **VADER** (Valence Aware Dictionary and Sentiment Reasoner). VADER contains a vast collection of words and phrases with assigned sentiment scores.

2, Polarity Scores: The primary output of the SentimentIntensityAnalyzer is a set of sentiment polarity scores, including:

- Positive Score (`pos`): The probability of the text conveying a positive sentiment.
- Negative Score (`neg`): The probability of the text conveying a negative sentiment.
- Neutral Score (`neu`): The probability of the text being neutral in sentiment.
- Compound Score (`compound`): An aggregated score that combines the positive, negative, and neutral scores to provide an overall sentiment polarity. The compound score ranges from -1 (most negative) to 1 (most positive).

Pros:

- Ease of Use: The SentimentIntensityAnalyzer provides a straightforward way to perform sentiment analysis without requiring extensive training data or complex models.
- Quick Insights: The polarity scores can give you a quick overview of the sentiment in a piece of text, making it useful for social media monitoring, customer reviews analysis, and more.
- No Training Needed: Unlike machine learning-based sentiment analysis models, the 'SentimentIntesityAnalyzer' does not require training on large datasets. It is based on a lexicon, which means you can use it out of the box.

Cons:

- Sensitivity to Context: The SentimentIntensityAnalyzer might struggle with understanding sarcasm, irony, or complex linguistic nuances, as it does not capture context as effectively as some machine learning models.

- Domain Dependence: The sentiment scores can be influenced by the domain-specific language that might not be well-represented in the lexicon.

- Thresholds and Classification: Determining sentiment based on fixed thresholds (like the compound score threshold in the example) might not be accurate for all types of text.

7.DATA VISUALIZATION AND REPRESENTATION

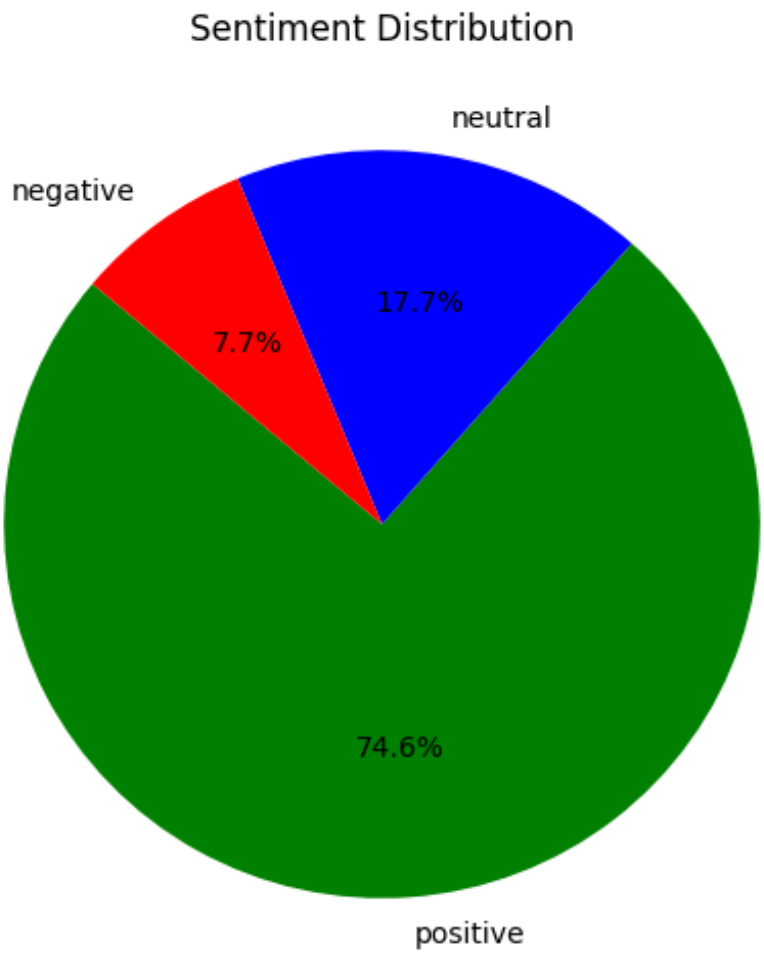

Fig.: Pie Chart for Text Sentiment:

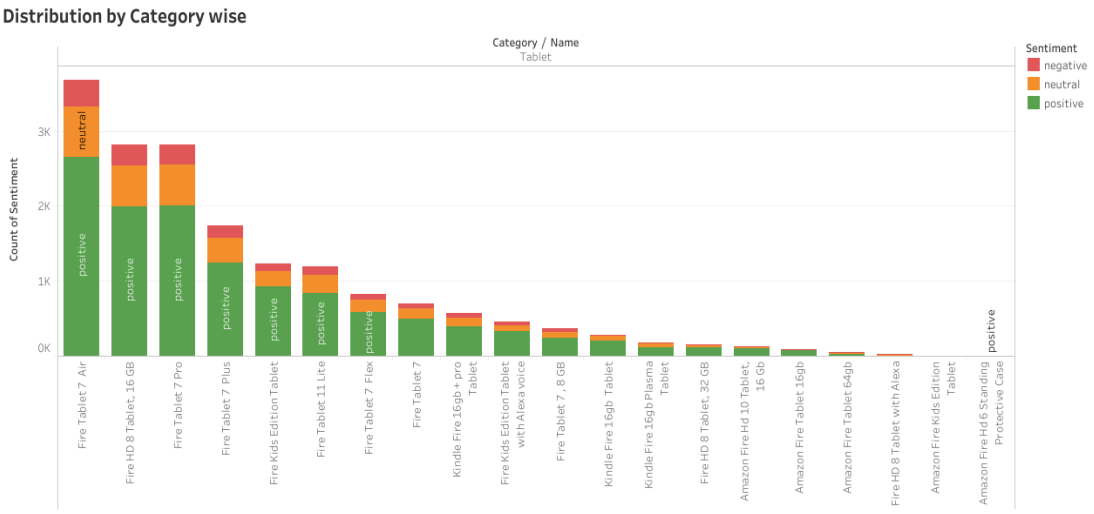

Fig.: Category wise Distribution

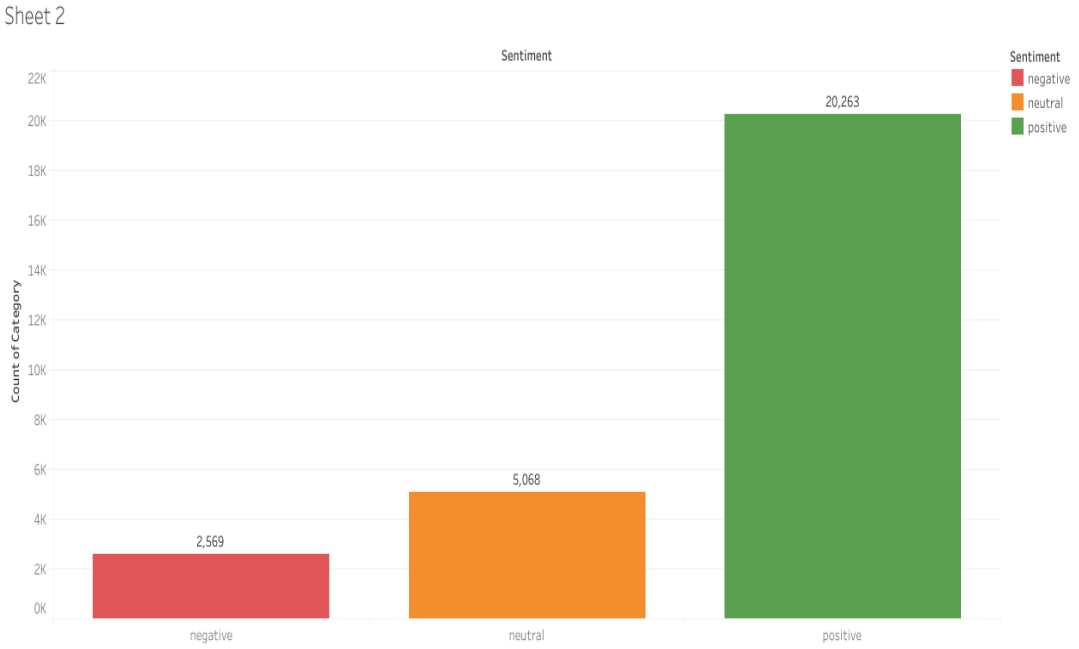

Fig.: Bar Plot Analysis

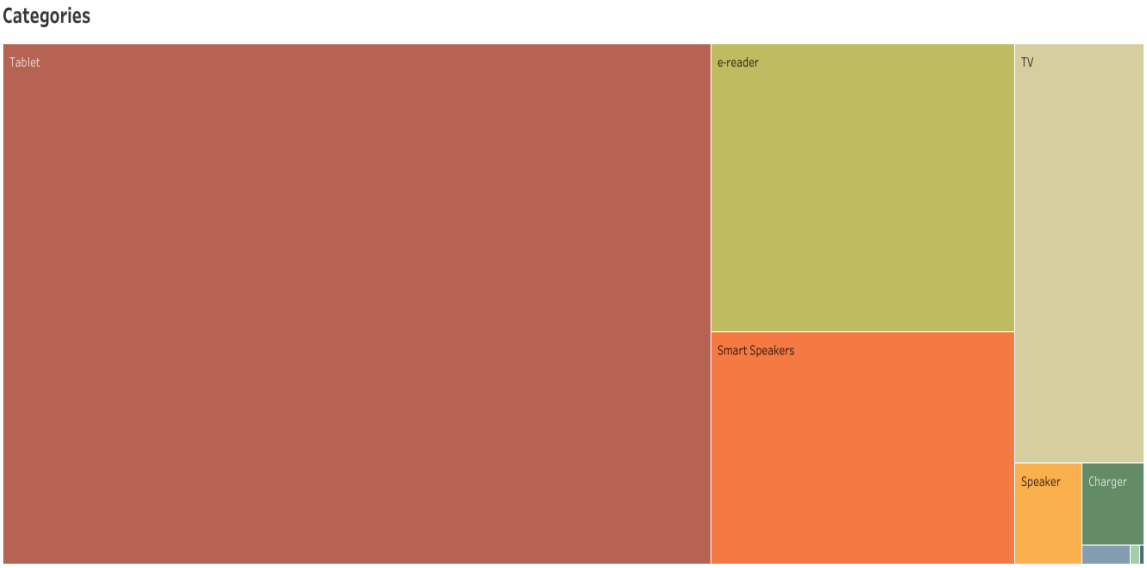

Fig.: Categories with Tree map

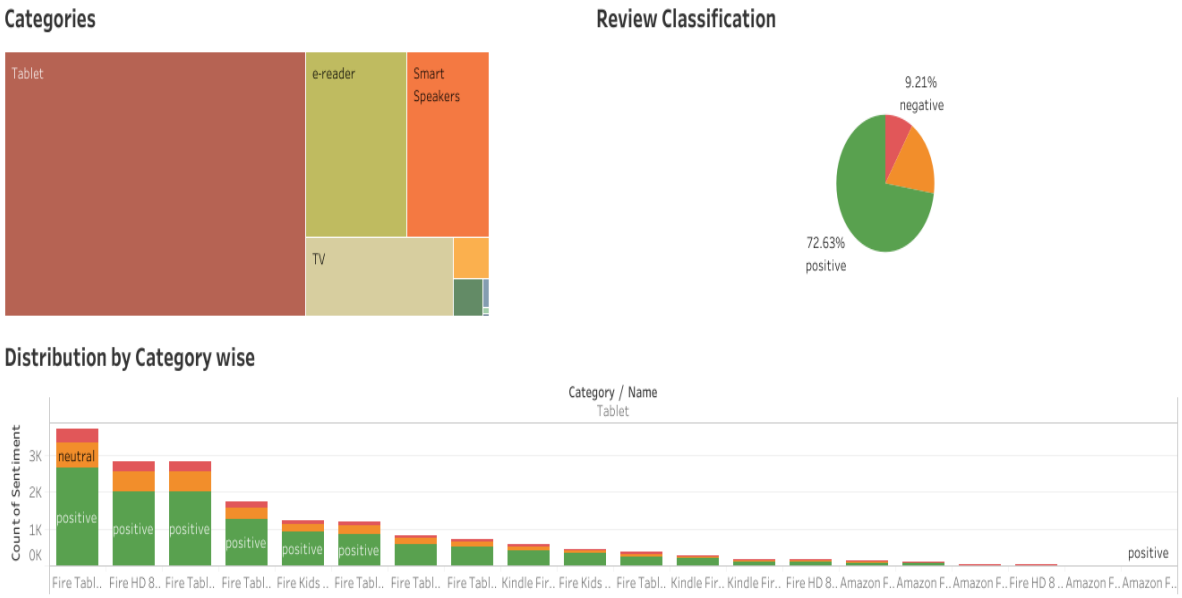

Fig.: Tableau Dashboard

8. STREAMLIT

Streamlit is an open-source Python library that allows data scientists and developers to create web applications for their data projects with minimal effort. It was designed to bridge the gap between data analysis and web application development, eliminating the need for extensive knowledge of frontend technologies like HTML, CSS, and JavaScript.

Key Advantages:

- **Simplicity**: Streamlit aims to simplify the process of creating interactive web applications. Its syntax is similar to writing Python scripts, making it accessible to data scientists and developers who might not be well-versed in web development.

- **Rapid Prototyping**: With Streamlit, you can quickly prototype and visualize your data, machine learning models, and insights without spending time on designing complex frontends.

- **Pythonic**: Streamlit is designed to work seamlessly with Python libraries commonly used in data science, such as NumPy, pandas, Matplotlib, and scikit-learn. This means you can leverage your existing Python skills and integrate your data analysis code directly into the app.

- **Interactive Widgets**: Streamlit provides a variety of built-in widgets like sliders, buttons, text input, and plots. These widgets enable users to interact with the data and models you present in your application.

- **Live Updates**: Streamlit apps automatically update as you modify your code and data. This live updating feature allows you to see changes in real-time as you experiment or fine-tune your models.

- **Customization**: While Streamlit simplifies frontend development, it still offers options for customizing the appearance and layout of your application to some extent.

- **Deployment**: Deploying a Streamlit app is relatively straightforward. You can host your app on various platforms, including cloud services like Heroku, AWS, or even your own server.

Deploying Machine Learning Models with Streamlit:

When it comes to deploying machine learning models using Streamlit, the process is quite streamlined. You can import your trained model, write the necessary code to load input data and obtain predictions, and then use Streamlit widgets to gather user input and display results.

In our Project, we create a simple sentiment analysis web app with Streamlit. The app has a text input widget where users can enter a sentence, and upon clicking a button, the machine learning model could predict whether the sentiment is positive, negative, or neutral. With just a few lines of code and the appropriate widgets, we created an interactive application that showcases your model's capabilities.

9.CONCLUSION AND FUTURE SCOPE

Conclusion:

After analyzing dataset, we can conclude that Machine Learning can be effectively used for understanding the customer's sentiment.

According to the Deep Learning (NLP) model, approximately 74.6% of customers were satisfied with the product, while 7.7% of customers were not satisfied. Additionally, about 17.7% of customers expressed a neutral sentiment towards the product.

In conclusion, leveraging machine learning algorithms enables businesses to make informed decisions by analyzing customer sentiment. This approach aids in prioritizing essential organizational actions and areas of focus.

Future Scope:

Looking ahead, refining Amazon's product review sentiment analysis can involve sensitivity or complex linguistic nuances, time periods of customer dissatisfaction. This would yield more nuanced insights into review sentiment, enabling tailored strategies for customer satisfaction. Furthermore, addressing confusing language and refining service can further enhance the customer experience.

References

1. https://www.kaggle.com/code/amakaebolue/amazon-customer-sentiments/input
2. https://colab.research.google.com/drive/1Bdc_YOd2I8ne5BUy2PQ6nQJIpI0GeCA2#scrollTo=nryFozeKxPd_
3. https://huggingface.co/datasets/amazon_us_reviews
4. https://snap.stanford.edu/data/web-Amazon.html
5. https://github.com/topics/amazon-review-dataset
6. https://data.world/promptcloud/amazon-product-reviews-dataset
7. https://data.world/datasets/reviews
8. https://itk.ilstu.edu//faculty/xfang13/amazon_data.htm
9. https://datarade.ai/data-products/amazon-data-reviews-by-keyword-by-category-by-seller-by-p-dataant
10. https://www.tensorflow.org/datasets/catalog/amazon_us_reviews
11. (PDF) Sentiment Analysis: Machine Learning Approach - ResearchGate https://www.researchgate.net/publication/318368881_Sentiment_Analysis_Machine_Learning_Approach