Sheffield Hallam University

Amazon Reviews for Sentiment Analysis

Name: Brijbhan Govardhan Prajapati (C3084787)

University: Sheffield Hallam University

Subject: Applied Artificial Intelligence

Sheffield Hallam University

INDEX:

Sheffield Hallam University

ABSTRACT:

The undertaking focuses on the sentiment analysis of Amazon reviews through the use of different natural language processing (NLP) strategies. Its essential goal is to gather important experiences from literary information by translating the sentiments passed on inside the reviews. With this point, the undertaking incorporates a few vital parts. At first, the dataset containing Amazon reviews, close by their related sentiment names, is stacked into memory. In this way, the information goes through preprocessing to refine and normalize it for analysis.

This preprocessing stage includes fundamental text standardization methods like tokenization, lowercasing, and the evacuation of stop words.

Following this, sentiment analysis is directed using the VADER (Valence Mindful Word reference and sentiment Reasoner) device, a particular sentiment analysis instrument custom-made for web-based entertainment messages. Besides, the task investigates highlight extraction utilizing the TF-IDF (Term Recurrence Converse Record Recurrence) strategy, which produces vectors addressing the significance of words in each survey compared with the whole dataset.

Eventually, the undertaking comes full circle in an analysis of the sentiment conveyance inside the dataset, joined by representations to delineate the commonness of positive and negative sentiments.

Keywords: Sentiment analysis, Amazon reviews, Natural language processing, Customer feedback, Machine learning, Sentiment classification, Product reviews

Sheffield Hallam University

INTRODUCTION:

In the present computerized period, online reviews assume a critical part in forming purchaser conduct and impacting buying choices. Among the plenty of stages, Amazon stands apart as a force to be reckoned with, facilitating a large number of item reviews traversing different classes. Tackling the abundance of data implanted inside these reviews can open significant experiences into client sentiment, item fulfillment, and market patterns. The "Amazon Reviews for Sentiment Analysis" project leaves on an excursion to tackle the force of natural language processing (NLP) and AI to disentangle the sentiment hidden inside Amazon reviews. We want to furnish organizations with noteworthy knowledge from the aggregate voice of their clients, empowering them to settle on information-driven choices and drive business development. At its center, this venture rotates around the significant comprehension that client sentiments embodied in reviews reach out a long way past simple words; they address an impression of encounters, feelings, and discernments. By diving profound into this repository of criticism, we expect to remove nuanced experiences that rise above the paired grouping of positive, negative, or nonpartisan sentiments. From the perspective of sentiment analysis, we try to disentangle the hidden stories woven into Amazon reviews, knowing examples, patterns, and sentiments that shape buyer mentalities and inclinations. Our central goal isn't simply to analyze individual reviews to improve item contributions, refine showcasing procedures, and develop perseverance through client connections. In this undertaking, we leave on an extraordinary excursion, where the combination of information science, man-made brainpower, and human instinct joins to enlighten the way ahead. By bridging the force of Amazon reviews, we try to enable organizations with the bits of knowledge they need to flourish in a consistently advancing commercial center, where client sentiment rules.

Sheffield Hallam University

Libraries And Dependencies:

The venture depends on a few Python libraries to work with different undertakings, including information stacking, message preprocessing, sentiment analysis, and information representation.

Every library fills a particular need and upgrades the usefulness of the undertaking. The following are the libraries imported in the task and their jobs:

```
[44] # Libreries
     import pandas as pd
     import nltk
     from nltk.tokenize import word_tokenize
     from nltk.corpus import stopwords
     from nltk.sentiment.vader import SentimentIntensityAnalyzer
     from sklearn.feature_extraction.text import TfidfVectorizer
     from sklearn.metrics import accuracy_score, classification_report
     from sklearn.model_selection import train_test_split
     from sklearn.svm import LinearSVC
```

Pandas:

Job: Pandas is a strong information control library in Python, normally utilized for stacking, cleaning, changing, and dissecting organized information.

Import: pandas as pd

NLTK (Natural Language Tool Kit):

Job: NLTK is a far-reaching library for natural language processing (NLP) undertakings in Python, giving devices and assets to errands like tokenization, stemming, lemmatization, grammatical feature labeling, and sentiment analysis.

```
[47] nltk.download('punkt')
     nltk.download('stopwords')
     nltk.download('vader_lexicon')

     [nltk_data] Downloading package punkt to /root/nltk_data...
     [nltk_data]   Package punkt is already up-to-date!
     [nltk_data] Downloading package stopwords to /root/nltk_data...
     [nltk_data]   Package stopwords is already up-to-date!
     [nltk_data] Downloading package vader_lexicon to /root/nltk_data...
     [nltk_data]   Package vader_lexicon is already up-to-date!
     True
```

SCIKIT-LEARN:

JOB: Scikit-learn is a flexible AI library in Python, offering devices and calculations for information mining, information analysis, and AI undertakings like grouping, relapse, bunching, and dimensionality decrease.

MATPLOTLIB:

JOB: matplotlib is a famous plotting library in Python, broadly utilized for making static, vivified, and intelligent representations. It empowers the age of graphs, plots, histograms, and different kinds of visual portrayals. Import: matplotlib.pyplot as plt.

LOADING THE DATASET:

The pd.read_csv() function is used to read the Amazon reviews dataset from a TSV file into a Pandas DataFrame. The file path, separator (sep), and column names are specified as arguments to the function.

```
[10]
     # Loading the data here
     train_data = pd.read_csv("/content/drive/MyDrive/Projects/AAI Proj/archive/test.ft.txt", sep='\t', header=None, names=["review", "sentiment"])
```

DRIVE MOUNT:

If the dataset is stored in a cloud storage service like Google Drive, the project may require mounting the drive to access the dataset. This can be achieved using appropriate methods provided by the cloud service's SDK.

```
[7]  #drive mount
     from google.colab import drive
     drive.mount('/content/drive')

     Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

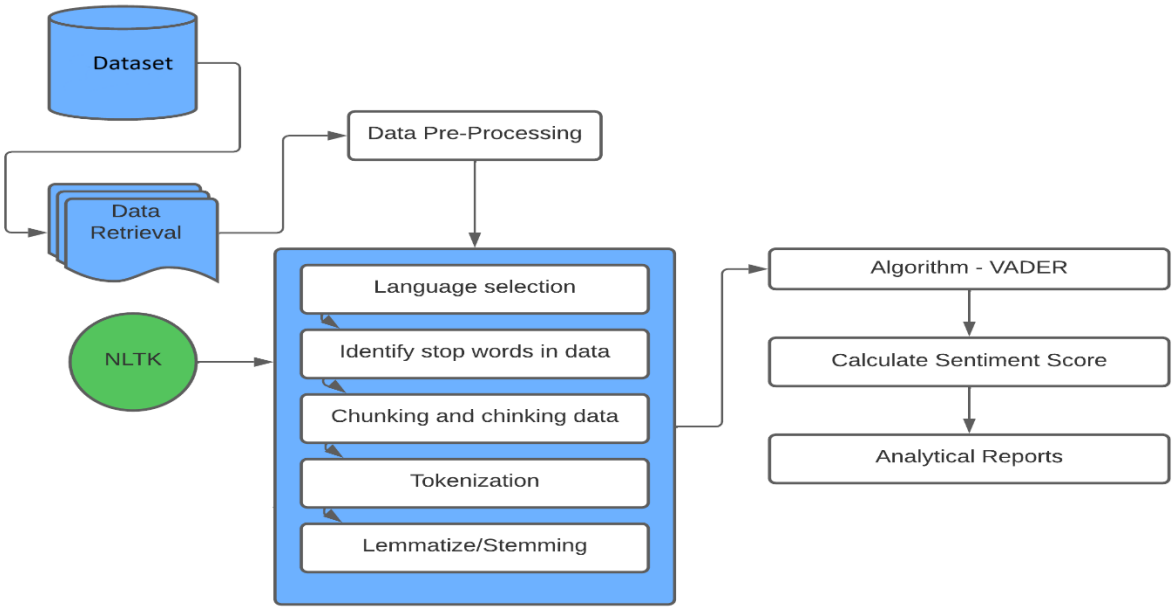
SLICING THE DATASET (Optional):

To expedite data loading and processing during development, a subset of the dataset may be selected for analysis. This can be accomplished by slicing the DataFrame to include a specified number of rows.

```
[8]  # slice the data to load it fast.(beacause In the Dataset there is too much data)
     train_data = train_data[:40000]
```

DATA PREPROCESSING:

When the information is stacked, it goes through preprocessing to set it up for analysis. This includes text standardization procedures like tokenization, lowercase transformation, and expulsion of stop words using NLTK.



```
# Preprocess text (remove stopwords and lowercase)
def preprocess_text(text):
    tokens = word_tokenize(text.lower())

    filtered_tokens = [token for token in tokens if token not in stop_words]
    return ' '.join(filtered_tokens)
stop_words = set(stopwords.words('english'))

train_data['review'] = train_data['review'].apply(preprocess_text)
```

PREPROCESSING STEPS:

Frame the primary preprocessing steps applied to the Amazon review dataset. Notice explicit strategies like tokenization, stop words expulsion, and lowercasing.

TOKENIZATION:

- Characterize tokenization as the method involved with parting text into individual words or tokens.
- Make sense of how tokenization helps separate text information into more modest units for analysis.

STOP WORDS EVACUATION:

- Characterize stop words as familiar words that don't convey critical importance (e.g., "the," "is," "and").
- Examine the significance of eliminating stop words to work on the precision of sentiment analysis.

LOWERCASING:

- Make sense of the idea of lowercasing, which includes switching all text over completely to lowercase.
- Feature that lowercasing guarantees consistency in word portrayal and improves on analysis.
- Other Preprocessing Strategies:
- Momentarily notice other preprocessing strategies that might have been applied, for example, stemming or lemmatization.
- Note that the particular strategies utilized may differ relying upon the qualities of the dataset and the prerequisites of the analysis.

Execution in Code:

- Give a concise outline of how preprocessing strategies were carried out in the code.
- Feature a particular capabilities or libraries utilized for preprocessing, like NLTK or scikit-learn.

Influence on Analysis:

- Examine how preprocessing procedures add to the precision and viability of sentiment analysis.
- Underscore that spotless and well pre-processed message information establishes the groundwork for precise sentiment grouping.

Sentiment Analysis with Vader(Methodology):

Sentiment analysis is performed on the preprocessed message information utilizing the VADER (Valence Mindful Word reference and sentiment Reasoner) instrument.

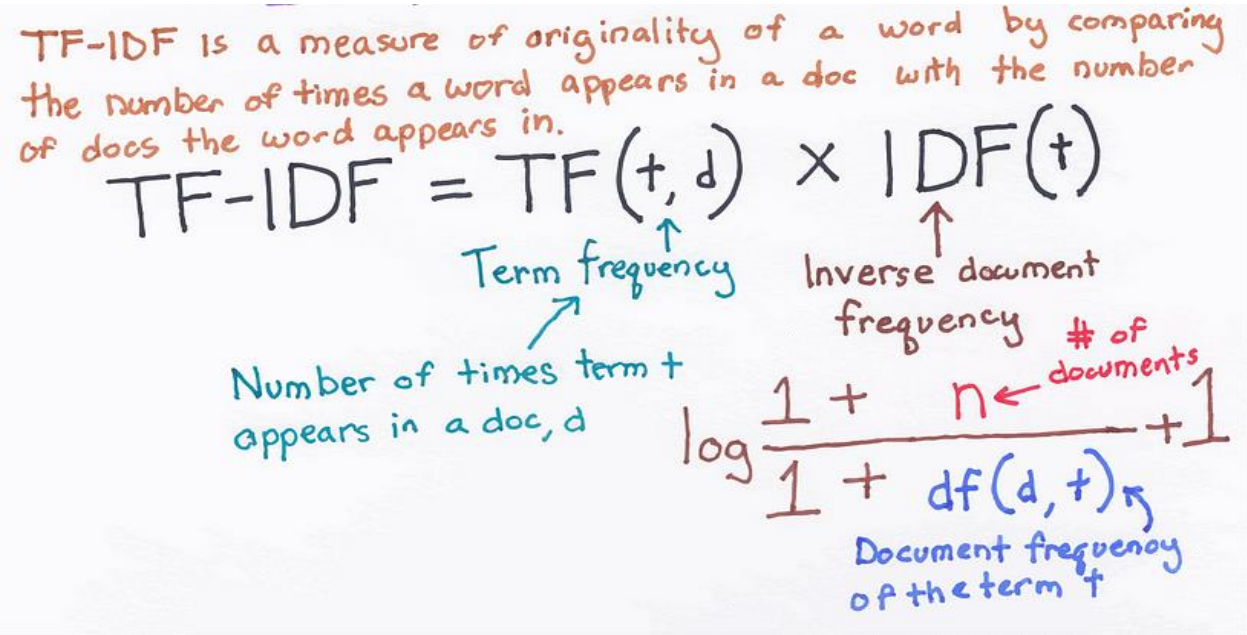
VADER is a standard-based sentiment analysis instrument explicitly intended for breaking down virtual entertainment messages. It gives sentiment scores to each survey, permitting arrangement as certain or negative in view of predefined edges.

```
[7]  # Sentiment analysis using VADER
     vader = SentimentIntensityAnalyzer()

     def get_vader_sentiment(text):
         scores = vader.polarity_scores(text)
         sentiment = 'positive' if scores['pos'] > scores['neg'] else 'negative'
         return sentiment

     train_data['vader_sentiment'] = train_data['review'].apply(get_vader_sentiment)
```

HIGHLIGHT EXTRACTION WITH TF-IDF:

Notwithstanding sentiment analysis with VADER, the task additionally investigates highlight extraction utilizing the TF-IDF (Term Recurrence Converse Report Recurrence) procedure.

TF-IDF is applied to the preprocessed message information to create highlight vectors, which address the significance of words in each survey compared with the whole dataset.




```
[8]  # extraction of the Feature using TF-IDF
     vectorizer = TfidfVectorizer(max_features=2000)  # we can Adjust max_features if needed
     train_features = vectorizer.fit_transform(train_data['review'])
```

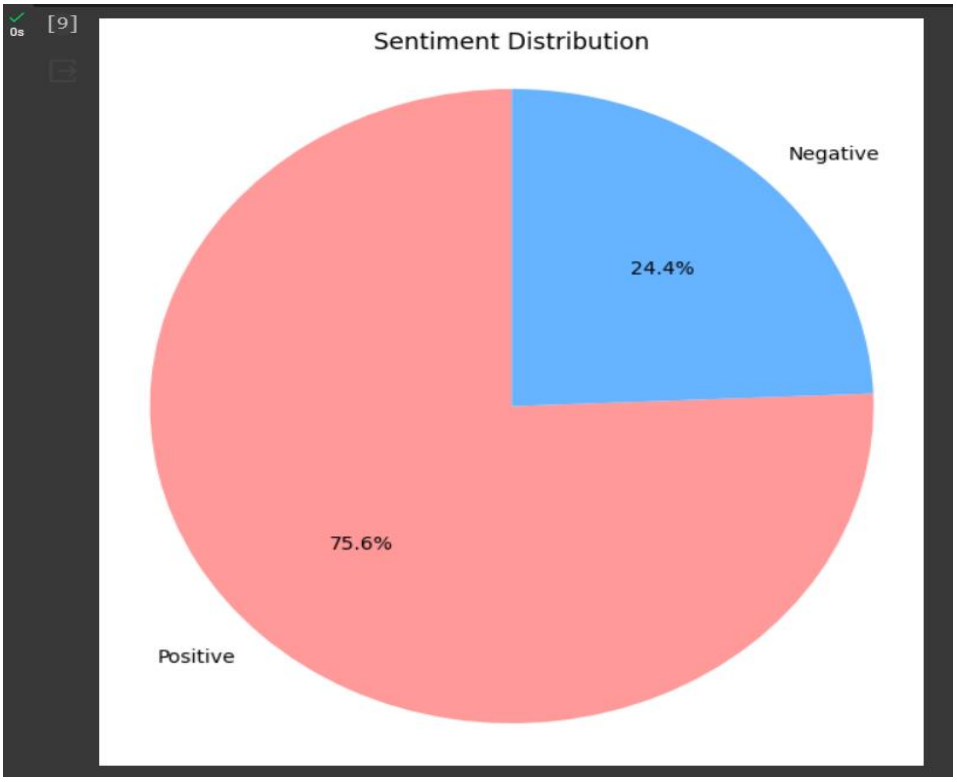
```
[52] # Splitting the data into train and test sets
     X_train, X_test, y_train, y_test = train_test_split(train_features, train_data['vader_sentiment'], test_size=0.2, random_state=42)

     # Training a Linear Support Vector Classifier
     classifier = LinearSVC()
     classifier.fit(X_train, y_train)

     # Predictions on the test set
     predictions = classifier.predict(X_test)

     # Evaluating the accuracy
     accuracy = accuracy_score(y_test, predictions)
     print("Accuracy:", accuracy)

     # Print classification report
     print("\nClassification Report:")
     print(classification_report(y_test, predictions))
```

```
Accuracy: 0.873

Classification Report:
              precision    recall  f1-score   support

    negative       0.77      0.68      0.72      1952
    positive       0.90      0.93      0.92      6048

    accuracy                           0.87      8000
   macro avg       0.84      0.81      0.82      8000
weighted avg       0.87      0.87      0.87      8000
```

SENTIMENT APPROPRIATION ANALYSIS:

The undertaking closes with an analysis of the sentiment conveyance in the dataset. This includes picturing the circulation of positive and negative sentiments utilizing Matplotlib. The level of positive and negative sentiments is determined and deciphered to acquire bits of knowledge into client criticism patterns.

Result in Graph:



Result in Numbers:

```
VADER Sentiment Distribution (Train Data):
positive    30255
negative     9745
Name: vader_sentiment, dtype: int64
```

CONCLUSION:

The sentiment analysis project on Amazon reviews has yielded significant bits of knowledge about client sentiments communicated inside the dataset. Through the utilization of different natural language processing (NLP) strategies, incorporating sentiment analysis with VADER and

highlight extraction utilizing TF-IDF, we have acquired a far-reaching comprehension of the sentiments common in the reviews. Key discoveries and ends from the undertaking include:

SENTIMENT APPROPRIATION ANALYSIS:

The sentiment appropriation analysis uncovered the pervasiveness of positive and negative sentiments inside the dataset. This knowledge gives organizations a comprehension of by and large consumer loyalty levels and regions for development.

HIGHLIGHT SIGNIFICANCE WITH TF-IDF:

Highlight extraction utilizing TF-IDF permitted us to recognize the main words and expressions adding to the sentiment of the reviews. This information can direct organizations in fitting their items and administrations to meet client inclinations.

NOTEWORTHY BITS OF KNOWLEDGE:

By utilizing the bits of knowledge gained from sentiment analysis, organizations can go with information-driven choices to upgrade consumer loyalty, further develop item contributions, and refine advertising procedures.

All in all, the sentiment analysis of Amazon reviews fills in as a significant device for organizations to comprehend and answer client criticism. Pushing ahead, further examination and analysis can extend how we might interpret client sentiments and empower nonstop refinement of items and administrations to meet advancing customer needs and inclinations.

REFERENCES:

Title: Sentiment Analysis and Opinion Mining: A Survey

Authors: **Vinodhini G**, **Dr. RM Chandrasekaran**

URL: (PDF) Sentiment Analysis and Opinion Mining: A Survey (researchgate.net)

Title: SENTIMENT ANALYSIS AND OPINION MINING

Authors: Raisa Varghese, Jayasree M

URL: IJRET_110211048 (psu.edu)

Title: Sentiment Analysis of Amazon Product Reviews using VADER and RoBERTa Models
Authors: Shadmaan Hussain, Namrata Dhanda, Rajat Verma

URL**: Sentiment Analysis of Amazon Product Reviews using VADER and RoBERTa Models | IEEE Conference Publication | IEEE Xplore**

NLTK Documentation:

Website: https://www.nltk.org

Documentation: https://www.nltk.org/#natural-language-toolkit

Sheffield Hallam University

Pandas Documentation:

Website: https://pandas.pydata.org

Documentation: https://pandas.pydata.org/docs/user_guide/index.html#user-guide

ss

Scikit-learn Documentation:

Website: https://scikit-learn.org

Documentation: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html#sklearn.feature_extraction.text.TfidfVectorizer

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html#sklearn.metrics.accuracy_score

Matplotlib Documentation:

Website: https://matplotlib.org

Documentation: https://matplotlib.org/stable/api/pyplot_summary.html#module-matplotlib.pyplot

DATABASE:

Amazon Reviews for Sentiment Analysis (kaggle.com)