E-COMMERCE PRODUCT SENTIMENT ASSESMENT AND ASPECT ANALYSIS

AN PROJECT REPORT
Submitted by

VIJAY RR [RA2011003040038]
AADITYA SHREERAM [RA2011003040051]
PRAVEEN M [RA2011003040086]

Under the Guidance of
Dr. S. Manohar
(Assistant Professor, Department of Computer Science and Engineering)

in partial fulfilment of the requirements for the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

FACULTY OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
VADAPALANI – 600 026

MAY 2024

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

BONAFIDE CERTIFICATE

Certified that this project report titled “**E-COMMERCE PRODUCT SENTIMENT ASSESMENT AND ASPECT ANALYSIS**” is the bonafide work of “**VIJAY R R [RegNo:RA2011003040038], AADITYA SHREERAM RS [RegNo:RA2011003040051],PRAVEEN M [RegNo:RA2011003040086]** ” who carried out the project work under my supervision .Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate. or award was conferred on an earlier occasion on this or any other candidate.

GUIDE
Dr. S. Manohar
Assistant Professor (O.G),
Dept of Computer Science & Engineering,
SRM IST
Vadapalani Campus

HEAD OF THE DEPARTMENT
Dr. S Prasanna Devi
B.E, M.E, PhD, PGDHRM, PDF(IISc)
Professor and Head,
Dept of Computer Science & Engineering,
SRM IST,
Vadapalani Campus

Signature of Internal Examiner

Signature of External Examiner

ACKNOWLEDGEMENTS

We express our humble gratitude to our Honorable Chancellor **Dr. T. R. Paarivendhar**, Pro Chancellor (Administration), **Dr. Ravi Pachamoothu**, Pro Chancellor (Academics) **Dr. P. Sathyanarayanan,** Pro Chancellor (Admin) for the facilities extended for the completion of the project work.

We would record our sincere gratitude to our Vice Chancellor, **Dr. C. Muthamizhchelvan and** Registrar, **Dr. S. Ponnusamy** for their support to complete our project work by giving us the best of academic excellence support system in place. We extend our sincere thanks to our Dean, **Dr. C V Jayakumar** and Vice Principal – Academics, **Dr. C. Gomathy** and Vice Principal - Examination **- Dr. S. Karthikeyan** for their invaluable support.

We wish to thank **Dr. S Prasanna Devi**, Professor & Head, Department of CSE, SRM Institute of Science and Technology, Vadapalani Campus for her valuable suggestions and encouragement throughout the period of the project work and the course.

We are extremely grateful to our Project Coordinator, **Dr. B. Prabha**, Assistant Professor, Department of CSE, SRM Institute of Science and Technology, Vadapalani Campus, for leading and helping us to complete our course.

We extend our gratitude to our supervisor, **Dr. S. Manohar**, Assistant Professor, Department of CSE, SRM Institute of Science and Technology, Vadapalani Campus for providing us an opportunity to pursue our project under his mentorship. He provided us the freedom and support to explore the research topics of our interest.

We sincerely thank all faculty of DCSE, staff and students of the department who have directly or indirectly helped our project.

Finally, we would like to thank our parents, our family members and our friends for their unconditional love, constant support, and encouragement.

Vijay R R

Aaditya Shreeram R S

Praveen M

Department of Computer Science and Engineering

SRM Institute of Science & Technology

Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Student Name : Vijay R R
Reg. Number : RA2011003040038
Title of Work : E-COMMERCE PRODUCT SENTIMENT ASSESMENT AND ASPECT ANALYSIS
Degree/ Course : B.TECH/CSE

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc.)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g., fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website.

  I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except were indicated by referring, and that I have followed the good academic practices noted above. Vijay R R RA2011003040038
If you are working in a group, please write your registration numbers and sign with the date forevery student in your group.

Department of Computer Science and Engineering

SRM Institute of Science & Technology

Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Student Name : Aaditya Shreeram R S
Reg. Number : RA2011003040051
Title of Work :E-COMMERCE PRODUCT SENTIMENT ASSESEMENT AND ASPECT ANALYSIS
Degree/ Course : B.TECH/CSE

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc.)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g., fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website.

  I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except were indicated by referring, and that I have followed the good academic practices noted above.

Aaditya Shreeram R S RA2011003040051

If you are working in a group, please write your registration numbers and sign with the date forevery student in your group.

Department of Computer Science and Engineering

SRM Institute of Science & Technology

Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

<u>To be completed by the student for all assessments</u>

Student Name : Praveen M

Reg. Number : RA2011003040086

Title of Work : E-COMMERCE PRODUCT SENTIMENT ASSESEMENT AND ASPECT ANALYSIS

Degree/ Course : B.TECH/CSE

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc.)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g., fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website.

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except were indicated by referring, and that I have followed the good academic practices noted above. Praveen M RA2011003040086
If you are working in a group, please write your registration numbers and sign with the date forevery student in your group.

ABSTRACT

Our project is dedicated to advancing Sentiment Analysis (SA) and Aspect-Based Sentiment Analysis (ABSA) for e-commerce product reviews, with a primary objective of providing users with valuable insights to support informed decision-making and assisting product creators in comprehending customer perceptions of their products. Embracing an innovative approach, our project surpasses conventional sentiment analysis by meticulously exploring specific product aspects, such as features and qualities, to deliver a detailed and comprehensive analysis.

The project workflow unfolds through several pivotal stages meticulously designed to extract profound insights from the extensive pool of reviews on e-commerce platforms. Commencing with meticulous data preprocessing tasks encompassing stemming, stop words removal, lemmatization, lower casing, contraction expansion, and tokenization, we lay the foundation for pristine and organized text data, ensuring precision in analysis outcomes.

Central to our project is the Aspect Term Extraction (ATE) phase, where an array of rules, including POS Tagging, Noun Combination, Dependency Parsing, and Stop Word Removal, are deployed to discern and extract aspects from the textual data. These rules are tailored to identify specific linguistic patterns and grammatical structures associated with aspects, thereby furnishing a profound understanding of different product facets and enabling a nuanced sentiment analysis.

Our project takes a deep dive into fine-tuning a BERT model for Aspect-Based Sentiment Analysis (ABSA), which involves processing the dataset to tailor the model for aspect sentiment classification. This phase involves tokenizing the text, identifying aspects and their corresponding polarities, then the model is trained to classify aspects as positive, neutral, or negative, and assigning sentiment scores to each aspect. By leveraging advanced machine learning techniques, we aim to enhance the accuracy and depth of sentiment analysis, providing users with detailed insights into customer sentiments regarding e-commerce products.

Overall, this methodology presents a structured and comprehensive approach to sentiment analysis in e-commerce, empowering businesses to extract actionable insights from vast amounts of textual data. By combining advanced techniques in natural language processing and deep learning, this project sets a new standard for sentiment analysis in the digital retail landscape.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	EXPANSION
ABSA	Aspect Based Sentiment Analysis
ATE	Aspect Term Extraction
BERT	Bidirectional Encoder Representations from Transformers
SA	Sentiment Analysis
NB	Multinomial Naive Bayes
AI	Artificial Intelligence
NER	Named Entity Recognition
NLP	Natural Language Processing
POS	Parts of Speech
LDA	Latent Dirichlet Allocation
PAM	Product Aspect Mining
CBOW	Continuous Bag of Words
SVM	Support Vector Machines
CNN	Convolutional Neural Networks
TFIDF	Term Frequency Inverse Document Frequency
ARM	Association Rule Mining

CHAPTER 1

INTRODUCTION

1.1 Sentiment Analysis (SA)

Sentiment analysis, a vital component of natural language processing, is dedicated to discerning and interpreting the sentiments conveyed within textual data. As our digital world continues to burgeon with vast amounts of textual content across various online platforms, sentiment analysis serves as a cornerstone for understanding public opinion, consumer preferences, and market trends. By analyzing the emotions, attitudes, and opinions expressed in text, sentiment analysis offers invaluable insights into customer sentiment, enabling businesses to gauge satisfaction levels, identify areas for improvement, and tailor their strategies to better meet customer needs.

Traditionally, sentiment analysis methods focused on categorizing text into broad sentiment categories, such as positive, negative, or neutral. However, with advancements in natural language processing and machine learning, more sophisticated techniques have emerged, surpassing simple polarity classification. Modern sentiment analysis techniques leverage a blend of text preprocessing, feature extraction, and machine learning algorithms to achieve a deeper understanding of sentiment nuances.

In this context, this project explores advanced sentiment analysis techniques aimed at extracting nuanced sentiment information from textual data. By employing state-of-the-art algorithms and methodologies, this project seeks to enhance the accuracy and granularity of sentiment analysis, enabling businesses to gain deeper insights into customer opinions, preferences, and behaviors. Through a combination of text preprocessing, feature extraction, and machine learning models, this project aims to empower businesses to make informed decisions, optimize their products and services, and ultimately, enhance customer satisfaction and loyalty in an increasingly digital world.

1.2 Aspect-Based Sentiment Analysis (ABSA)

Aspect-Based Sentiment Analysis (ABSA) offers a comprehensive approach to sentiment analysis by focusing on specific aspects or features mentioned in textual data, providing detailed insights into customer sentiments towards different attributes of products or services. In contrast to traditional sentiment analysis, which offers a broad overview of overall sentiment, ABSA delves into the text to analyze sentiments associated with individual aspects, such as product features or service qualities. This granular analysis allows businesses to gain a deeper understanding of customer opinions, identify strengths and weaknesses in products or services with precision, and tailor strategies to meet customer preferences effectively.

ABSA encompasses a range of methodologies for aspect extraction and sentiment analysis, including rule-based approaches, dependency parsing. By leveraging these techniques, ABSA aims to extract relevant aspects from text data and determine the sentiment polarity linked to each aspect.

In the context of this project, ABSA plays a pivotal role in enhancing sentiment analysis by providing detailed insights into customer feedback from e-commerce product reviews. By extracting aspects from the reviews and analyzing sentiment associated with each aspect, businesses can pinpoint specific areas for enhancement, prioritize product improvements, and customize marketing strategies to align with customer expectations.

Through the incorporation of advanced ABSA techniques, this project aims to elevate traditional sentiment analysis methods, enabling businesses to extract deeper insights from textual data and make data-driven decisions to thrive in the competitive e-commerce landscape.

1.3 Our System

Our system represents a paradigm shift in sentiment analysis for e-commerce product reviews, combining Sentiment Analysis (SA) and Aspect-Based Sentiment Analysis (ABSA) to empower users with actionable insights. In today's digital age, the abundance of online product reviews can overwhelm consumers, hindering their ability to make well-informed decisions. Our solution addresses this challenge by distilling vast amounts of review data into digestible insights, aiding users in their purchasing deliberations. Moreover, our system serves as a valuable tool for product creators, offering nuanced feedback on their offerings and illuminating customer sentiments across various aspects.

Built upon a foundation of robust preprocessing techniques, our system employs a meticulous pipeline to extract meaningful insights from raw text data. Through processes such as stemming, stop words removal, and lemmatization, we ensure that the data is primed for analysis. Leveraging advanced rules based on Part-of-Speech (POS) tagging, noun combination, dependency parsing, and stop word removal, our system identifies and extracts aspects crucial to understanding customer sentiment. By dissecting reviews into granular components such as features, performance, and user experience, we enable a comprehensive understanding that goes beyond traditional sentiment analysis.

Central to our system's functionality is the integration of a fine-tuned BERT model for Aspect Sentiment Classification. By leveraging state-of-the-art machine learning techniques, we provide users with sentiment scores for each aspect identified within a review. This sophisticated approach not only categorizes sentiments as positive, neutral, or negative but also offers a nuanced understanding of customer opinions. Through the fusion of cutting-edge natural language processing methods and advanced machine learning models, our system stands at the forefront of e-commerce sentiment analysis, empowering both consumers and product creators in navigating the complexities of the online marketplace.

CHAPTER 2

PROBLEM STATEMENT

In the era of digital commerce, the proliferation of online reviews on platforms like Amazon and eBay presents a formidable challenge for consumers and businesses alike. The sheer volume of feedback inundates consumers, making it arduous to distill pertinent information crucial for making informed purchasing decisions. This flood of opinions not only complicates the consumer journey but also poses a significant hurdle for businesses striving to maintain a positive brand image.

For consumers, navigating through countless reviews to extract meaningful insights becomes a daunting task, often leading to information overload and decision paralysis. Amidst this deluge of feedback, distinguishing between genuine sentiments and noise becomes increasingly challenging, hindering the ability to make confident purchasing choices.

Concurrently, businesses face the formidable task of monitoring and managing online reviews to gauge consumer sentiment accurately. Failing to effectively interpret and respond to feedback can have dire consequences for brand reputation and customer loyalty, impacting market competitiveness and revenue streams.

Our study aims to address these pressing challenges by proposing innovative solutions to alleviate review overload for consumers and equip businesses with actionable insights to navigate the complex landscape of online feedback. Through the integration of advanced technologies and strategic approaches, we aspire to empower both consumers and businesses in optimizing their e-commerce experiences and fostering mutually beneficial relationships in the digital marketplace.

CHAPTER 3

LITERATURE REVIEW

[1] Prakash and Sharma (2023) utilized the Product Aspect Mining (PAM) technique for Aspect-Based Sentiment Analysis (ABSA) on Amazon product reviews. Focusing specifically on headphone and earphone feedback, they effectively extracted aspect terms and their associated polarity. By employing PAM, they were able to identify and analyze various aspects of the products, providing insights into customer sentiments towards different features. [1]

[2] Wahyudi and Kusumaningrum (2019) investigated sentiment analysis in e-commerce user reviews, particularly focusing on Aspect-Based Sentiment Analysis (ABSA). They employed Latent Dirichlet Allocation (LDA), a probabilistic topic modeling technique, to classify sentiments expressed in the reviews. Their study demonstrated the effectiveness of LDA in capturing the underlying topics and sentiments within both general and per-category training data, offering a comprehensive understanding of customer opinions in e-commerce contexts. [2]

[3] He, Zhou, and Zhao (2022) proposed a fusion sentiment analysis method for e-commerce reviews, aiming to enhance sentiment analysis accuracy. By combining traditional techniques with machine learning approaches, their method achieved an impressive accuracy rate of 80.36%. Their study explored various aspects of the e-commerce product experience, providing valuable insights for businesses aiming to improve customer satisfaction and product quality based on customer feedback. [3]

[4] Sudiro, Prasetiyowati, and Sibaroni (2021) conducted Aspect-Based Sentiment Analysis (ABSA) using a combination of feature extraction techniques, including Latent Dirichlet Allocation (LDA) and Word2Vec. By leveraging LDA for topic modeling and Word2Vec for word embeddings, they achieved notable accuracy rates of 80.36% with Skip-gram and 74.37% with Continuous Bag of Words (CBOW) models. Their study focused on analyzing sentiments associated with different aspects of products, with a particular emphasis on the packaging aspect, which achieved an accuracy rate of 89.71%.[4]

[5] Nasim et al. (2017) proposed a hybrid model integrating TF/IDF, lexicon-based approaches, and machine learning algorithms like Random Forest and SVM for analyzing student feedback. Achieving an impressive accuracy of 0.93 and F-measure of 0.92, their approach demonstrated a notable improvement of 0.02 over existing methods. [5]

[6] In a similar vein, Xiangyu et al. (2017) introduced a context-based regularization method for short-text sentiment analysis. Their model combined word similarity and word sentiment, showcasing substantial improvements, with an accuracy boost of over 4.5% compared to the baseline. The inclusion of new word-sentiment calculations

contributed to the enhanced performance observed across various datasets, including movie comments and social media discussions. [6]

[7] Krishna et al. (2017) focused on leveraging coreference resolution alongside SVM for sentiment analysis. By extracting relations between sentences, their approach significantly improved accuracy, particularly in analyzing sentiment from vast datasets sourced from ecommerce sites. The fusion of coreference resolution with SVM yielded promising results, highlighting the efficacy of this feature-based approach. [7]

[8] Yadav and Pandya (2017) presented SentiReview, a sentiment analysis framework based on text and emoticons. Their lexicon-based approach, coupled with machine learning for polarity analysis, offered insights into sentiment polarities across different platforms like Twitter and Weibo. By comparing various methods, SentiReview contributed to the understanding of sentiment analysis techniques and their applicability in diverse contexts. [8]

[9] Lastly, Gao et al. (2016) introduced a Convolutional Neural Network (CNN) based sentiment analysis model utilizing Adaboost combination. By leveraging boosted CNN models and Adaboost regularization, their approach achieved a notable accuracy of 89.4% in movie review sentiment analysis on the IMDB dataset, showcasing a modest improvement of 0.2%.[9]

[10] Ashok et al. (2018) conducted evaluations using Naive Bayes (NB), SVM, and Maximum Entropy (MaxEnt) classifiers. While SVM yielded the best results, indicating a strong connection between product aspects and individual sentiments, the overall accuracy remained moderate. This suggests room for further refinement in model performance to enhance accuracy. [10]

[11] Aitor et al. (2018) introduced W2VLDA, a multidomain and multilingual Aspect-Based Sentiment Analysis (ABSA) system operating primarily in an unsupervised manner. Leveraging vast amounts of unlabeled textual data and a small set of seed words, W2VLDA demonstrated the potential for enhancing ABSA analysis by efficiently identifying polarity. Further improvements could be achieved through better handling of multi-word expressions and negative statements, enhancing the system's versatility and accuracy. [11]

[12] Nurulhuda et al. (2018) integrated Principal Component Analysis (PCA) for sentiment word feature selection in an aspect-based hybrid sentiment classification approach. This involved utilizing Support Vector Machines (SVM) for sentiment classification and combining Part-of-Speech (POS) patterns with Association Rule Mining (ARM) to identify explicit single and multi-word features. Despite the hybrid approach's conceptual soundness in aspect identification, the study highlighted the need for improving accuracy to ensure model stability and flexibility. [12]

These papers collectively contribute to the field of sentiment analysis and Aspect-Based Sentiment Analysis (ABSA), offering insights into various methodologies and techniques for extracting sentiments from textual data in e-commerce contexts.

CHAPTER 4

METHODOLOGY

4.1 Dataset

To facilitate sentiment analysis and domain knowledge acquisition, two distinct datasets have been selected, each tailored to address specific research objectives:

4.1.1 Phone Review Dataset

The phone review dataset, sourced from Kaggle, comprises approximately 17,000 reviews, with each phone model associated with around 2,000 reviews. This dataset offers a comprehensive collection of user opinions and preferences towards various phone models, enabling researchers to conduct sentiment analysis tasks and gain valuable insights into consumer sentiments in the mobile device domain. The extensive coverage of reviews across multiple phones enhances the dataset's utility for training sentiment classification models and deepening domain understanding.

4.1.2 SemEval 2014 Laptop Dataset

The SemEval 2014 Laptop dataset serves as a specialized resource for aspect-based sentiment analysis (ABSA), focusing specifically on laptop products. This annotated dataset features reviews annotated with aspects mentioned in the reviews along with their corresponding sentiment polarities (positive, negative, or neutral). With meticulous annotations and diverse sources, this dataset provides a valuable benchmark for researchers seeking to develop and evaluate algorithms for understanding nuanced sentiments towards specific attributes of laptops. The detailed annotations and varied sentiments captured in this dataset offer researchers a robust foundation for advancing sentiment analysis techniques within the context of product reviews, particularly in a focused domain such as laptops.

4.2 Sentiment Analysis (SA)

Sentiment analysis involves the process of analyzing textual data to determine the underlying sentiment expressed within the text. The following steps outline the methodology employed for sentiment analysis in this research:

4.2.1 Text Preprocessing

Text preprocessing serves as a vital process in refining and standardizing text data for further analysis. It involves various techniques like converting text to lowercase, expanding contractions, managing negations, and removing non-alphanumeric characters and excessive spaces. Through the systematic application of these methods, raw text is converted into a clean and consistent format, enhancing the accuracy and significance of insights derived from sentiment analysis tasks.

1. **Lowercasing:**
   Before analyzing text data, it is essential to standardize the case of all words by converting them to lowercase. This ensures uniformity and prevents the model from treating words with different cases as distinct entities.

2. **Contraction Expansion:**
   Contractions like "don't" or "can't" are expanded to their full forms (e.g., "do not" or "cannot") for consistency and clarity in the text.

3. **Handling Negations:**
   Negations such as "not" or "no" can significantly alter the sentiment of a sentence. Therefore, identifying and appropriately handling these negations is crucial to accurately capture the intended sentiment.

4. **Removing Noise:**
   Non-alphanumeric characters and extra spaces are removed to reduce noise and improve the efficiency of subsequent analysis.

5. **Tokenization:**
   Text is tokenized by breaking it down into smaller units, typically words or phrases referred to as tokens. This process facilitates further analysis by treating each token as a discrete unit of meaning.

6. **Stopword Removal:**
   Stopwords—common words like "and" or "the"—are removed to focus on the essential content of the text and eliminate noise that could distort sentiment analysis results.

7. **Lemmatization:**
   Lemmatization involves reducing words to their base or dictionary form, known as the lemma. This step standardizes words and reduces lexical variations, ensuring consistency in the representation of words with similar meanings.

8. **Eliminating Redundancies:**
   Single-character tokens and numerical values are eliminated as they often do not contribute to the semantic meaning of the text and may introduce noise into the analysis.

4.2.2 Sentiment Classification

Sentiment classification methods are utilized to categorize textual data into positive, negative, or neutral sentiments based on the sentiment conveyed in the text. This classification process provides insights into the overall sentiment direction of the text, facilitating the extraction of sentiment-related information from large volumes of textual data.

4.2.3 Model Selection and Evaluation

Various machine learning and deep learning models, including Multinomial Naive Bayes, Decision Trees, and Logistic Regression, are considered for sentiment classification tasks. These models are trained and evaluated using the curated datasets to assess their performance and accuracy in classifying sentiments within textual data.

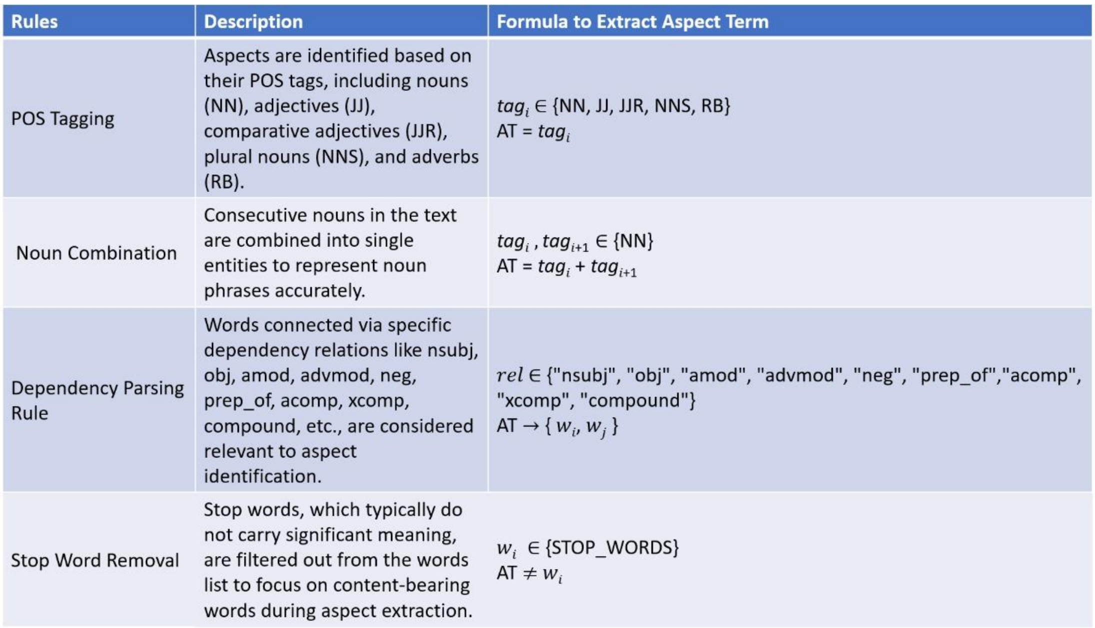
4.2.4 Model Integration and Ensemble Techniques

To enhance the accuracy and robustness of sentiment analysis, ensemble techniques are employed to combine the predictions of multiple models. By leveraging the strengths of different classifiers, ensemble methods offer improved performance and reliability in sentiment classification tasks, ultimately enhancing the quality of extracted sentiment insights.

4.3 Aspect-Based Sentiment Analysis (ABSA)

Aspect-based sentiment analysis (ABSA) focuses on extracting specific aspects or features mentioned in the text and analyzing the sentiment associated with each aspect. The ABSA methodology comprises the following steps:

4.3.1 Aspect Extraction Techniques



Rules	Description	Formula to Extract Aspect Term
POS Tagging	Aspects are identified based on their POS tags, including nouns (NN), adjectives (JJ), comparative adjectives (JJR), plural nouns (NNS), and adverbs (RB).	$tag_i \in$ {NN, JJ, JJR, NNS, RB} AT = tag_i
Noun Combination	Consecutive nouns in the text are combined into single entities to represent noun phrases accurately.	tag_i , $tag_{i+1} \in$ {NN} AT = $tag_i + tag_{i+1}$
Dependency Parsing Rule	Words connected via specific dependency relations like nsubj, obj, amod, advmod, neg, prep_of, acomp, xcomp, compound, etc., are considered relevant to aspect identification.	$rel \in$ {"nsubj", "obj", "amod", "advmod", "neg", "prep_of","acomp", "xcomp", "compound"} AT → { w_i, w_j }
Stop Word Removal	Stop words, which typically do not carry significant meaning, are filtered out from the words list to focus on content-bearing words during aspect extraction.	w_i $\in$ {STOP_WORDS} AT $\neq$ w_i

Fig 4.1 Rule-Based Aspect Extraction Methods

Aspect extraction techniques are employed to identify and extract key aspects or features from the text data. These techniques may include rule-based approaches, dependency parsing, and linguistic patterns to capture relevant aspects mentioned in the reviews.

4.3.2 Aspect Sentiment Classification

For Aspect Sentiment Classification, our methodology employs a fine-tuning approach with the BERT (Bidirectional Encoder Representations from Transformers) model. Initially, we initialize the BERT tokenizer and model, setting the stage for subsequent processing steps. Following initialization, we proceed by defining the target aspect and corresponding sentiment for each review sentence. This crucial step lays the groundwork for accurately categorizing sentiment towards specific aspects of the product under review.

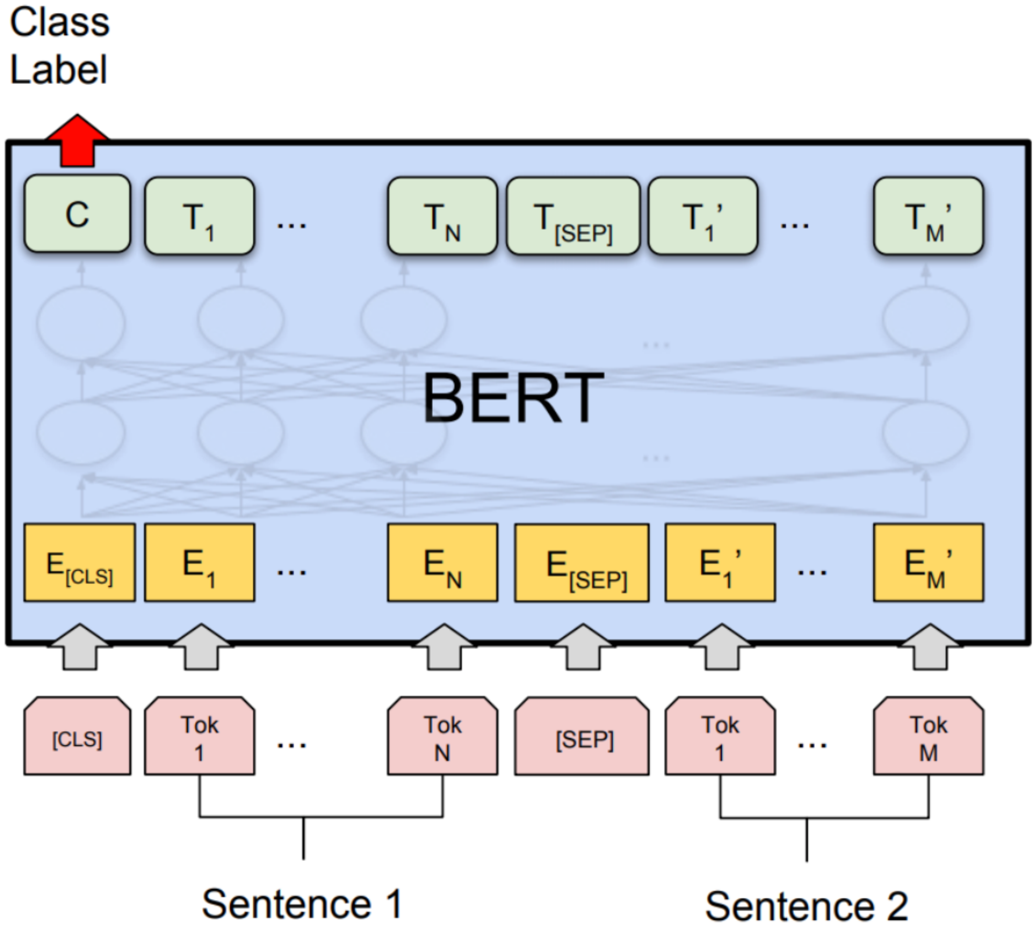


Fig 4.2 BERT Fine-Tuning Model

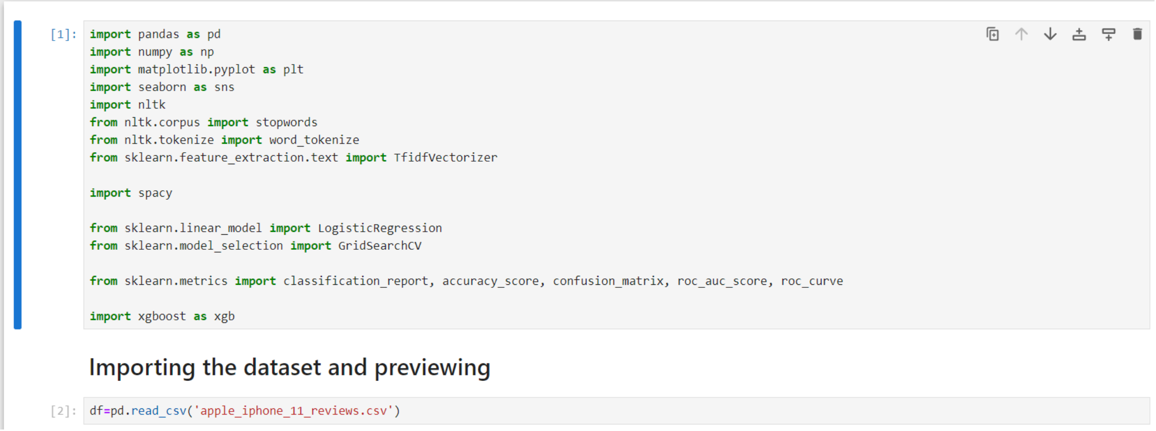
The subsequent steps of our methodology involve the transformation of text data into input tensors suitable for BERT processing. This involves tokenization of the text, construction of input tensors with special tokens marking the beginning and end of the text, and segment IDs distinguishing between the sentence and aspect tokens. Once these preparations are complete, the input tensors are passed through the BERT model, extracting hidden states and generating sentiment predictions via a linear layer. Fine-tuning of the model occurs through the utilization of labeled data, enabling adjustment of the model's parameters through backpropagation to minimize the loss function. With the model trained and optimized, it becomes proficient in Aspect Sentiment Classification, accurately predicting sentiment scores (positive, neutral, or negative) for each aspect mentioned in the review sentences. This comprehensive methodology ensures a robust and precise analysis of sentiment across various aspects, thereby enhancing the depth and granularity of our system's insights into customer opinions.

4.4 Website Development using Streamlit and ABSA Integration

In our methodology, we have utilized Streamlit to develop a user-friendly website interface for seamless interaction with our Aspect-Based Sentiment Analysis (ABSA) model. Leveraging Streamlit's capabilities, we have crafted a dynamic platform that allows users to effortlessly access and utilize the ABSA model trained as part of our project. The website interface provides an intuitive environment where users can input product reviews, triggering the ABSA model to generate insightful sentiment analyses for each aspect of the reviewed products. By integrating our ABSA model into this web-based interface, we ensure that the sophisticated sentiment analysis capabilities developed in our project are easily accessible and effectively utilized by users. This integration not only enhances the usability of our system but also democratizes access to advanced sentiment analysis tools, empowering users to make informed decisions based on comprehensive insights extracted from e-commerce product reviews.

CHAPTER 5

CODING & TESTING



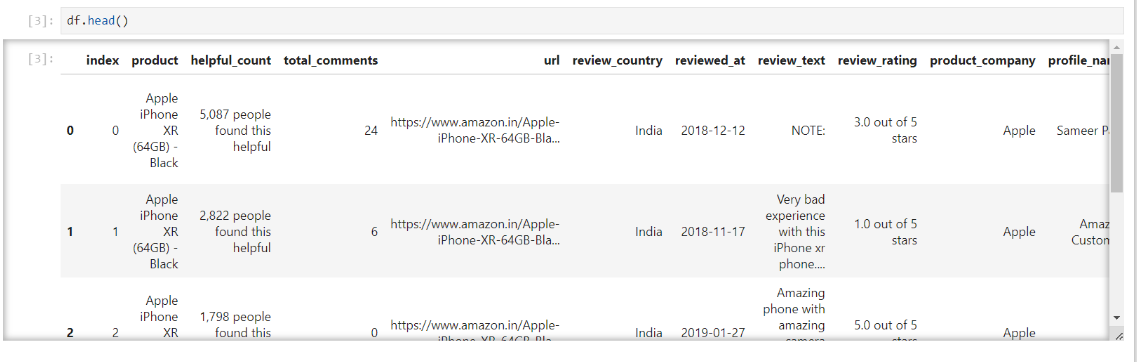
[1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer

import spacy

from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV

from sklearn.metrics import classification_report, accuracy_score, confusion_matrix, roc_auc_score, roc_curve

import xgboost as xgb
```

Importing the dataset and previewing

[2]:

```
df=pd.read_csv('apple_iphone_11_reviews.csv')
```

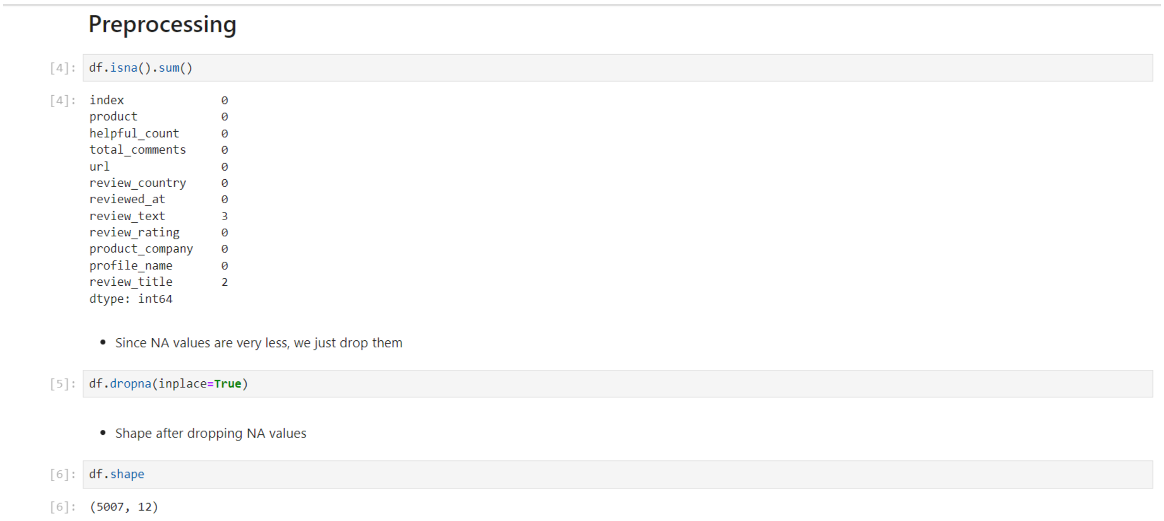


[3]:

```
df.head()
```

[3]:

	index	product	helpful_count	total_comments	url	review_country	reviewed_at	review_text	review_rating	product_company	profile_nai
0	0	Apple iPhone XR (64GB) - Black	5,087 people found this helpful	24	https://www.amazon.in/Apple-iPhone-XR-64GB-Bla...	India	2018-12-12	NOTE:	3.0 out of 5 stars	Apple	Sameer P.
1	1	Apple iPhone XR (64GB) - Black	2,822 people found this helpful	6	https://www.amazon.in/Apple-iPhone-XR-64GB-Bla...	India	2018-11-17	Very bad experience with this iPhone xr phone....	1.0 out of 5 stars	Apple	Amaz Custon
2	2	Apple iPhone XR	1,798 people found this	0	https://www.amazon.in/Apple-	India	2019-01-27	Amazing phone with amazing	5.0 out of 5	Apple	



Preprocessing

[4]:

```
df.isna().sum()
```

[4]:

```
index              0
product            0
helpful_count      0
total_comments     0
url                0
review_country     0
reviewed_at        0
review_text        3
review_rating      0
product_company    0
profile_name       0
review_title       2
dtype: int64
```

- Since NA values are very less, we just drop them

[5]:

```
df.dropna(inplace=True)
```

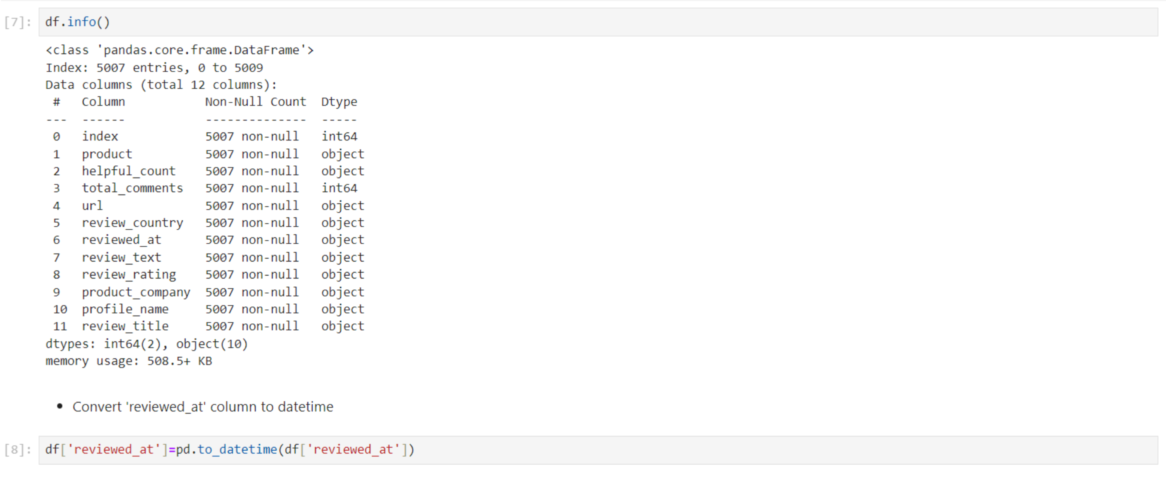
- Shape after dropping NA values

[6]:

```
df.shape
```

[6]:

```
(5007, 12)
```

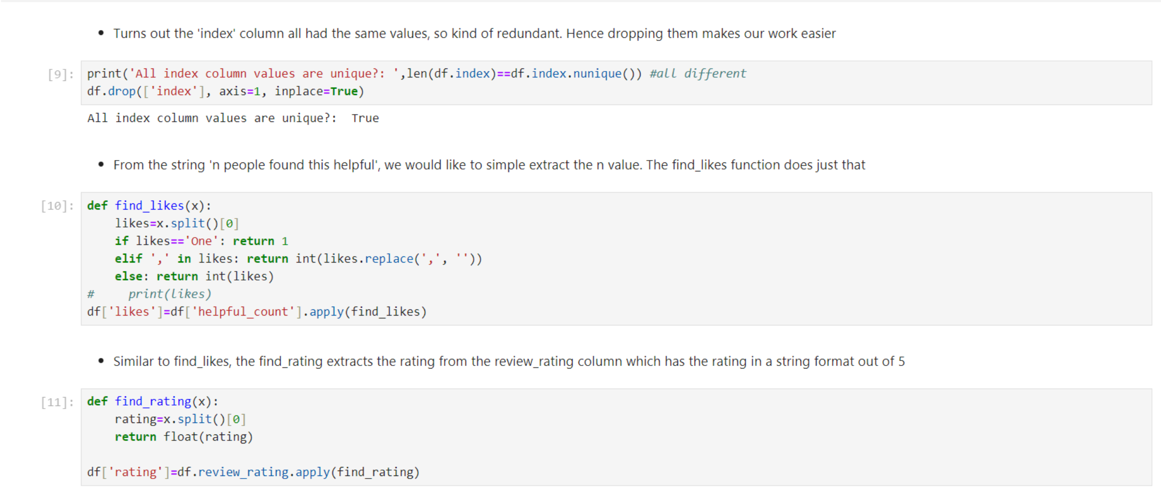


```
[7]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 5007 entries, 0 to 5009
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   index            5007 non-null   int64 
 1   product          5007 non-null   object
 2   helpful_count    5007 non-null   object
 3   total_comments   5007 non-null   int64 
 4   url              5007 non-null   object
 5   review_country   5007 non-null   object
 6   reviewed_at      5007 non-null   object
 7   review_text      5007 non-null   object
 8   review_rating    5007 non-null   object
 9   product_company  5007 non-null   object
 10  profile_name     5007 non-null   object
 11  review_title     5007 non-null   object
dtypes: int64(2), object(10)
memory usage: 508.5+ KB
```

- Convert 'reviewed_at' column to datetime

```
[8]: df['reviewed_at']=pd.to_datetime(df['reviewed_at'])
```



- Turns out the 'index' column all had the same values, so kind of redundant. Hence dropping them makes our work easier

```
[9]: print('All index column values are unique?: ',len(df.index)==df.index.nunique()) #all different
     df.drop(['index'], axis=1, inplace=True)
```

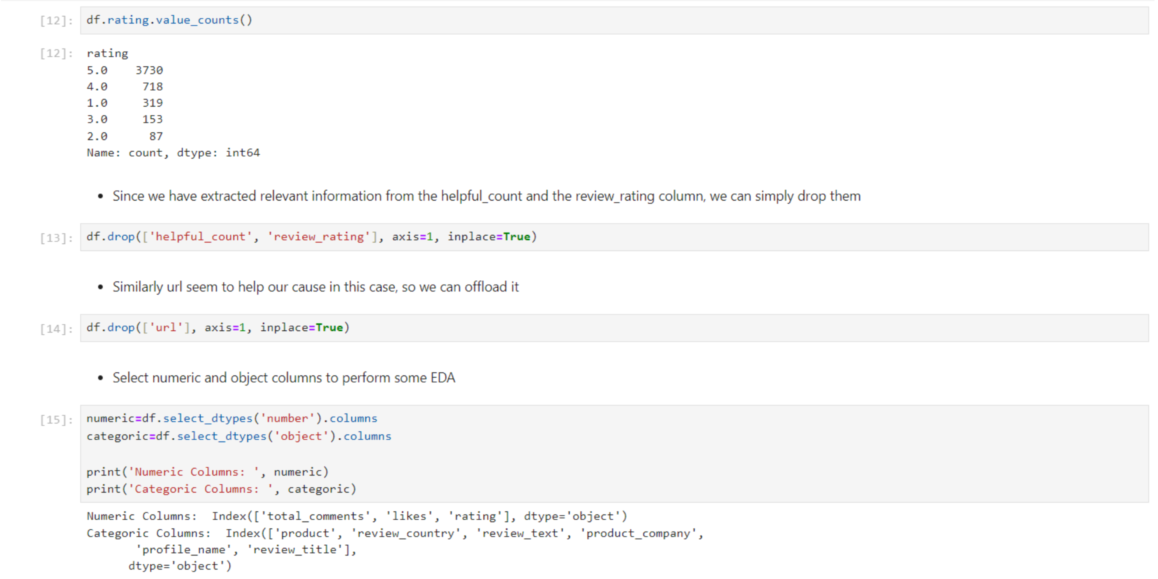
```
All index column values are unique?:  True
```

- From the string 'n people found this helpful', we would like to simple extract the n value. The find_likes function does just that

```
[10]: def find_likes(x):
          likes=x.split()[0]
          if likes=='One': return 1
          elif ',' in likes: return int(likes.replace(',', ''))
          else: return int(likes)
      #     print(likes)
      df['likes']=df['helpful_count'].apply(find_likes)
```

- Similar to find_likes, the find_rating extracts the rating from the review_rating column which has the rating in a string format out of 5

```
[11]: def find_rating(x):
          rating=x.split()[0]
          return float(rating)

      df['rating']=df.review_rating.apply(find_rating)
```



```
[12]: df.rating.value_counts()
```

```
[12]: rating
      5.0    3730
      4.0     718
      1.0     319
      3.0     153
      2.0      87
      Name: count, dtype: int64
```

- Since we have extracted relevant information from the helpful_count and the review_rating column, we can simply drop them

```
[13]: df.drop(['helpful_count', 'review_rating'], axis=1, inplace=True)
```

- Similarly url seem to help our cause in this case, so we can offload it

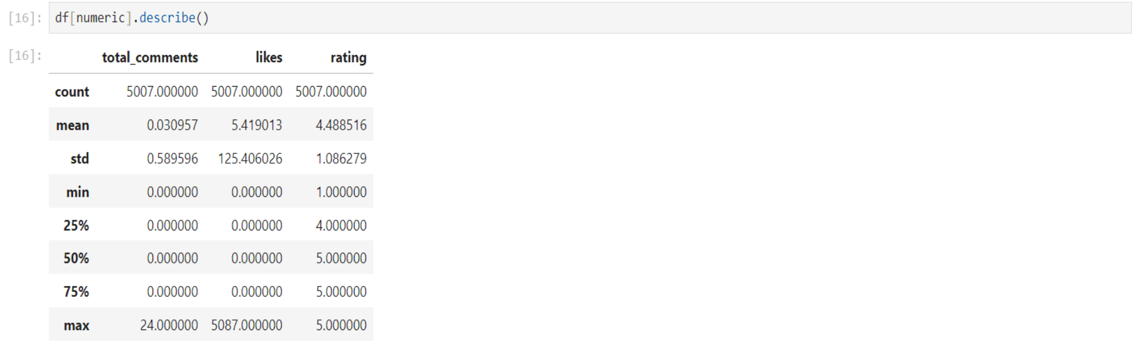
```
[14]: df.drop(['url'], axis=1, inplace=True)
```

- Select numeric and object columns to perform some EDA

```
[15]: numeric=df.select_dtypes('number').columns
      categoric=df.select_dtypes('object').columns

      print('Numeric Columns: ', numeric)
      print('Categoric Columns: ', categoric)
```

```
Numeric Columns:  Index(['total_comments', 'likes', 'rating'], dtype='object')
Categoric Columns:  Index(['product', 'review_country', 'review_text', 'product_company',
       'profile_name', 'review_title'],
      dtype='object')
```

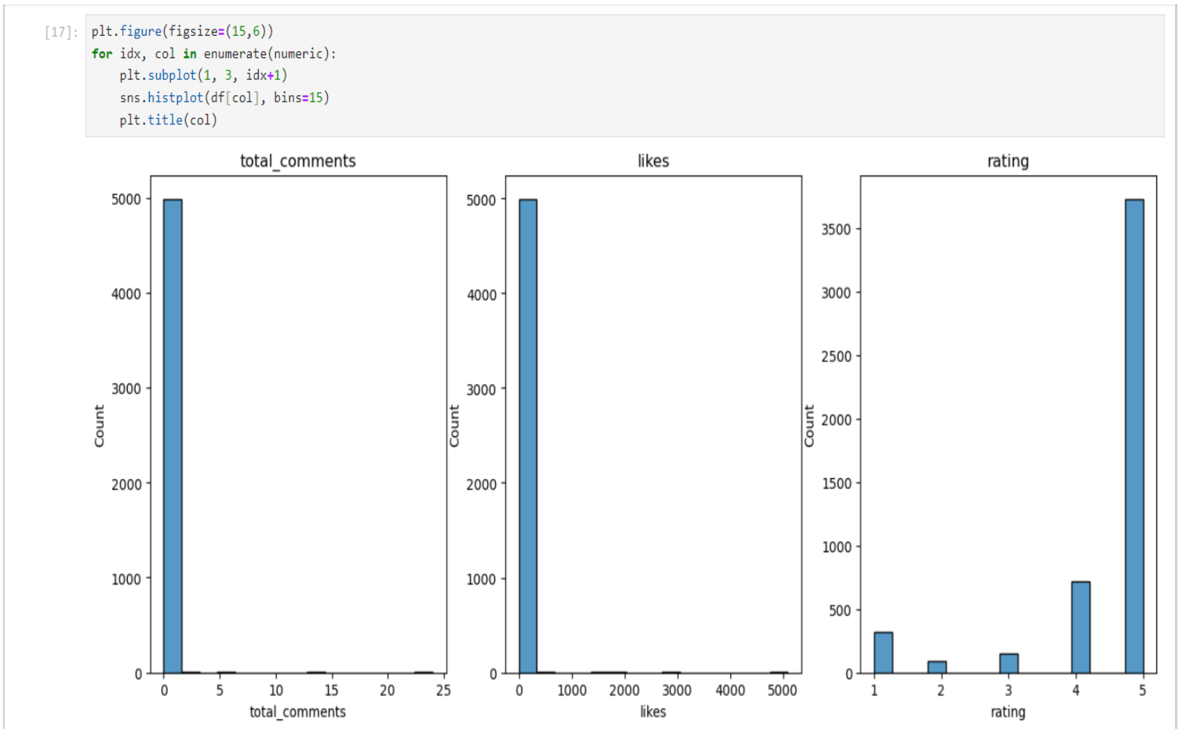
EDA & Visualizations



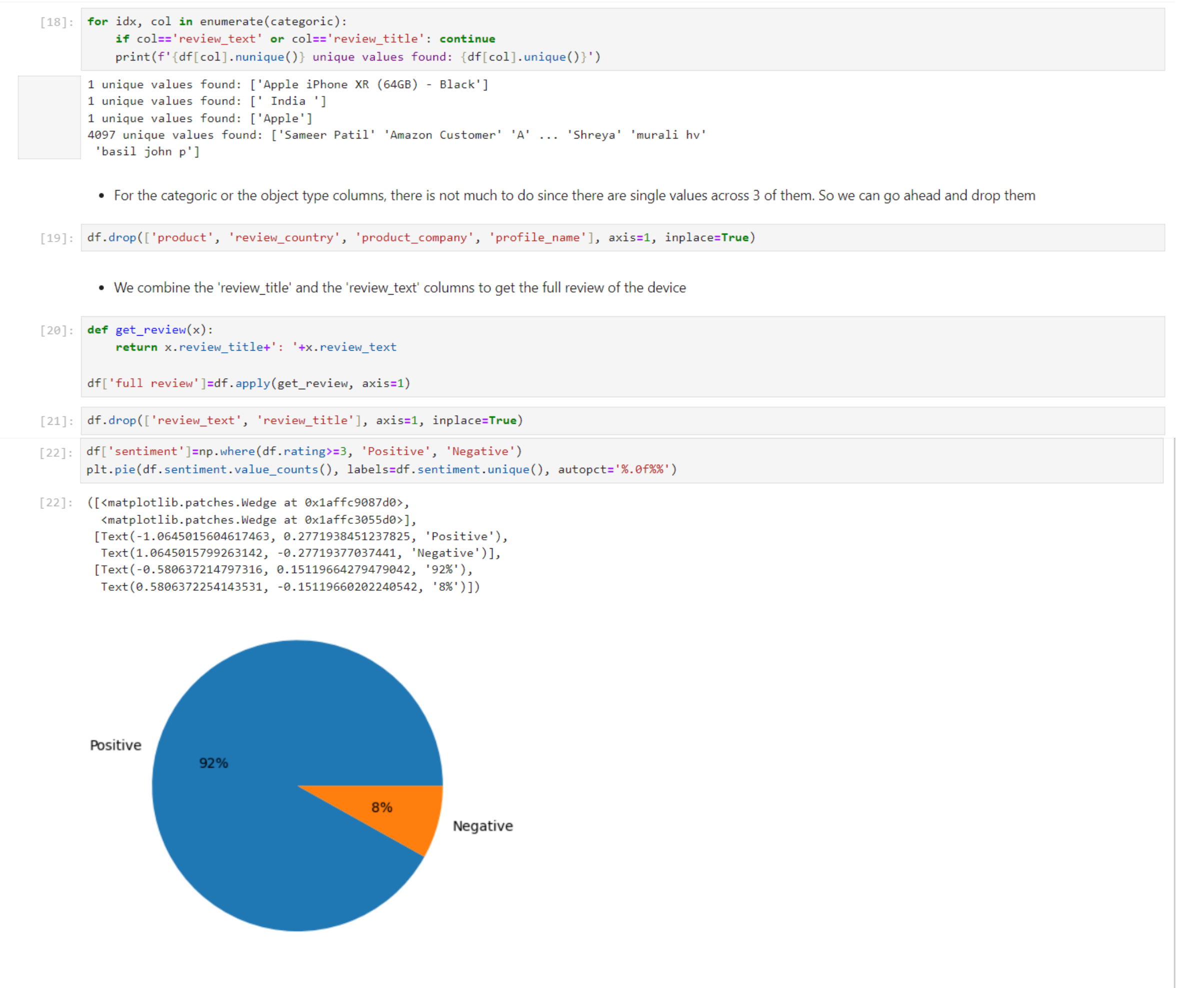
```
[16]: df[numeric].describe()
```

[16]:

	total_comments	likes	rating
count	5007.000000	5007.000000	5007.000000
mean	0.030957	5.419013	4.488516
std	0.589596	125.406026	1.086279
min	0.000000	0.000000	1.000000
25%	0.000000	0.000000	4.000000
50%	0.000000	0.000000	5.000000
75%	0.000000	0.000000	5.000000
max	24.000000	5087.000000	5.000000



```
[17]: plt.figure(figsize=(15,6))
      for idx, col in enumerate(numeric):
          plt.subplot(1, 3, idx+1)
          sns.histplot(df[col], bins=15)
          plt.title(col)
```



```
[18]: for idx, col in enumerate(categoric):
          if col=='review_text' or col=='review_title': continue
          print(f'{df[col].nunique()} unique values found: {df[col].unique()}')
```

```
1 unique values found: ['Apple iPhone XR (64GB) - Black']
1 unique values found: [' India ']
1 unique values found: ['Apple']
4097 unique values found: ['Sameer Patil' 'Amazon Customer' 'A' ... 'Shreya' 'murali hv'
 'basil john p']
```

- For the categoric or the object type columns, there is not much to do since there are single values across 3 of them. So we can go ahead and drop them

```
[19]: df.drop(['product', 'review_country', 'product_company', 'profile_name'], axis=1, inplace=True)
```

- We combine the 'review_title' and the 'review_text' columns to get the full review of the device

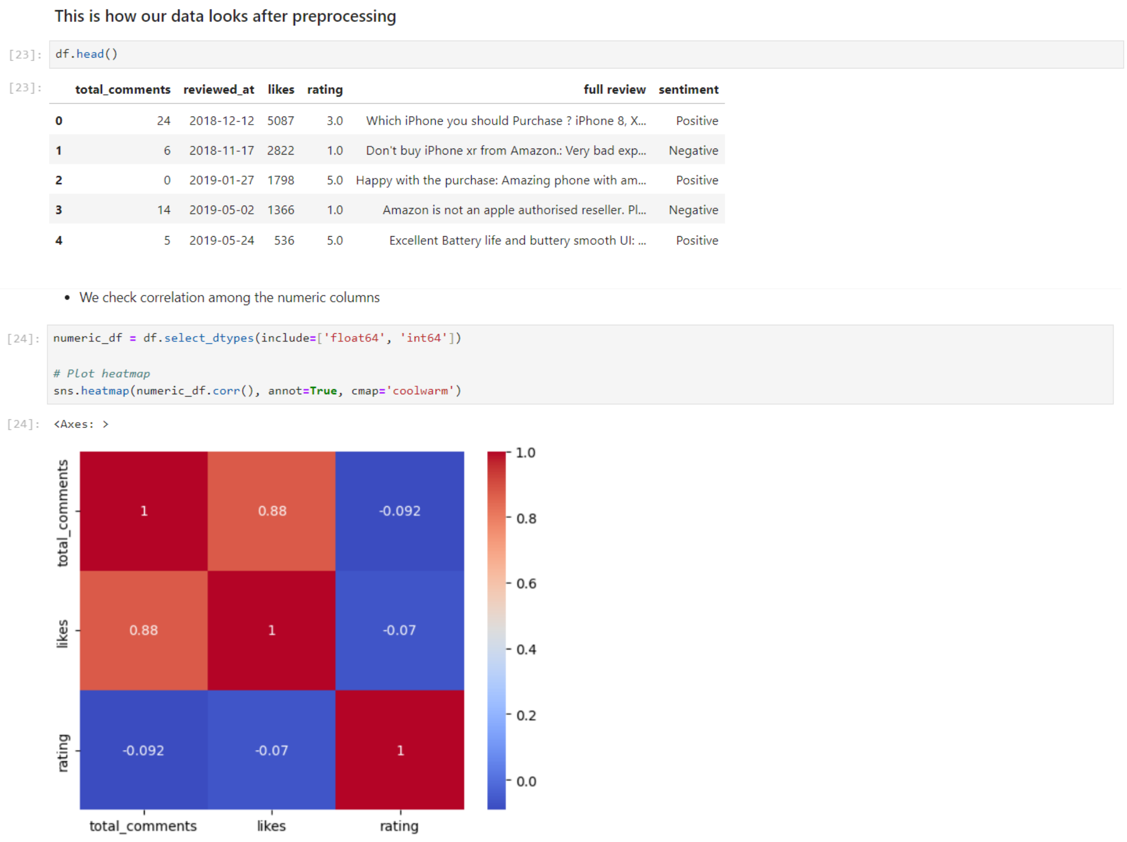
```
[20]: def get_review(x):
          return x.review_title+': '+x.review_text

      df['full review']=df.apply(get_review, axis=1)
```

```
[21]: df.drop(['review_text', 'review_title'], axis=1, inplace=True)
```

```
[22]: df['sentiment']=np.where(df.rating>=3, 'Positive', 'Negative')
      plt.pie(df.sentiment.value_counts(), labels=df.sentiment.unique(), autopct='%.0f%%')
```

```
[22]: ([<matplotlib.patches.Wedge at 0x1affc9087d0>,
        <matplotlib.patches.Wedge at 0x1affc3055d0>],
       [Text(-1.0645015604617463, 0.2771938451237825, 'Positive'),
        Text(1.0645015799263142, -0.27719377037441, 'Negative')],
       [Text(-0.580637214797316, 0.15119664279479042, '92%'),
        Text(0.5806372254143531, -0.15119660202240542, '8%')])
```

This is how our data looks after preprocessing



This is how our data looks after preprocessing

```
[23]: df.head()
```

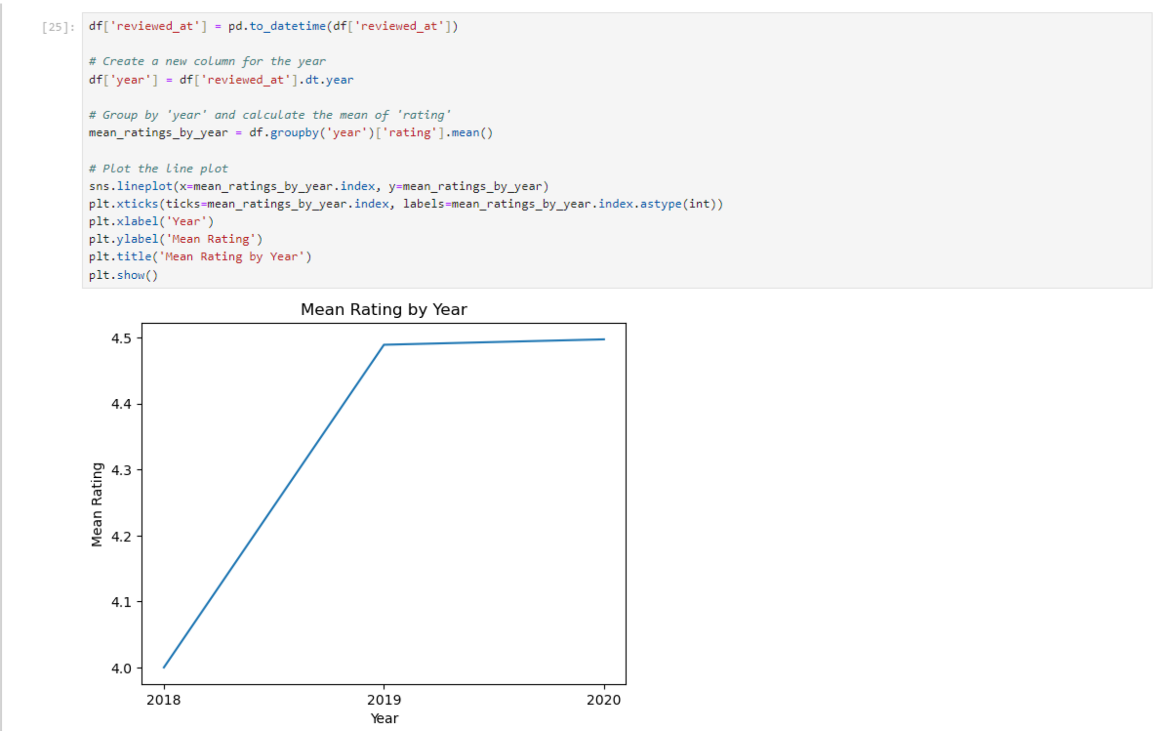
[23]:

	total_comments	reviewed_at	likes	rating	full review	sentiment
0	24	2018-12-12	5087	3.0	Which iPhone you should Purchase ? iPhone 8, X...	Positive
1	6	2018-11-17	2822	1.0	Don't buy iPhone xr from Amazon.: Very bad exp...	Negative
2	0	2019-01-27	1798	5.0	Happy with the purchase: Amazing phone with am...	Positive
3	14	2019-05-02	1366	1.0	Amazon is not an apple authorised reseller. Pl...	Negative
4	5	2019-05-24	536	5.0	Excellent Battery life and buttery smooth UI: ...	Positive

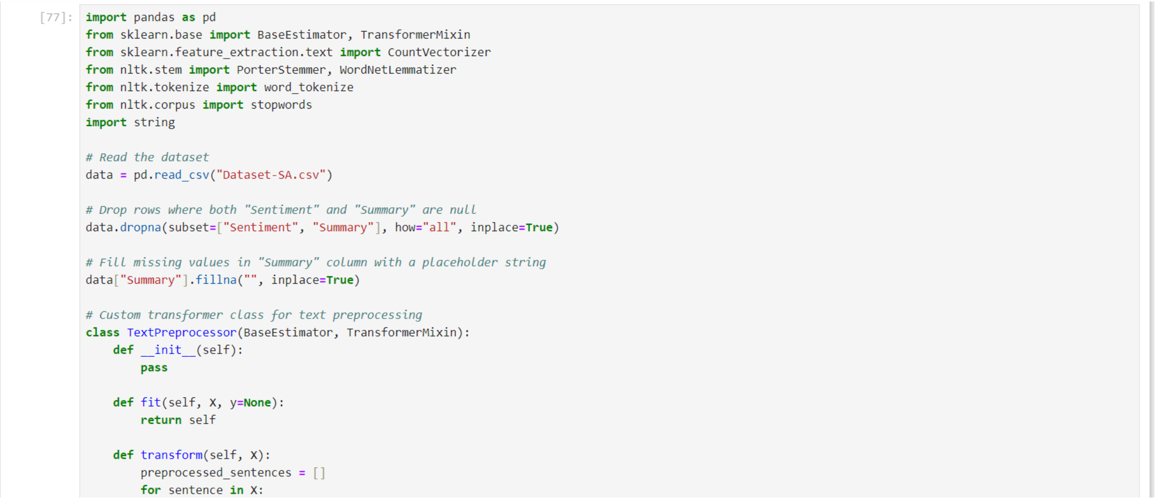
- We check correlation among the numeric columns

```
[24]: numeric_df = df.select_dtypes(include=['float64', 'int64'])

      # Plot heatmap
      sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
```

```
[24]: <Axes: >
```

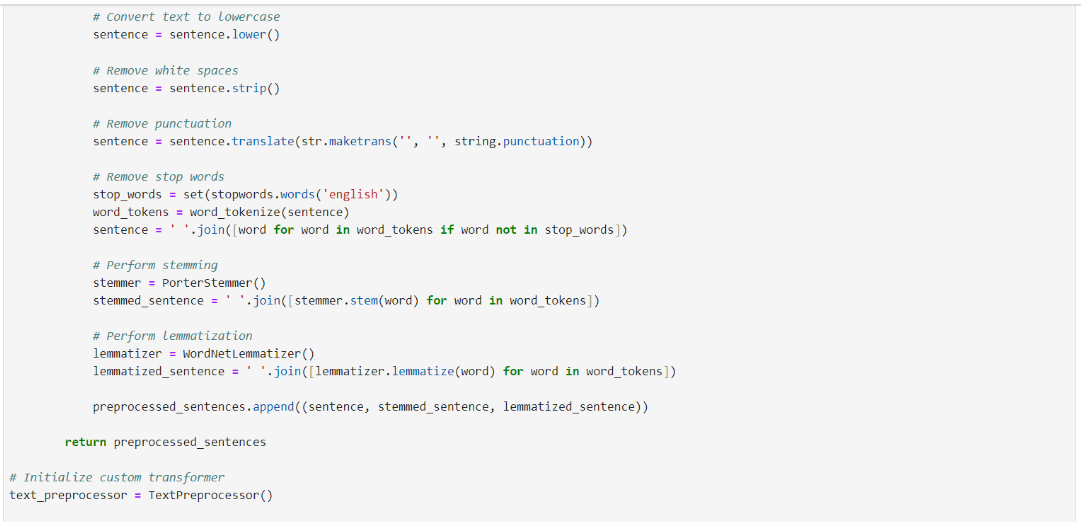


```
[25]: df['reviewed_at'] = pd.to_datetime(df['reviewed_at'])

      # Create a new column for the year
      df['year'] = df['reviewed_at'].dt.year

      # Group by 'year' and calculate the mean of 'rating'
      mean_ratings_by_year = df.groupby('year')['rating'].mean()

      # Plot the line plot
      sns.lineplot(x=mean_ratings_by_year.index, y=mean_ratings_by_year)
      plt.xticks(ticks=mean_ratings_by_year.index, labels=mean_ratings_by_year.index.astype(int))
      plt.xlabel('Year')
      plt.ylabel('Mean Rating')
      plt.title('Mean Rating by Year')
      plt.show()
```

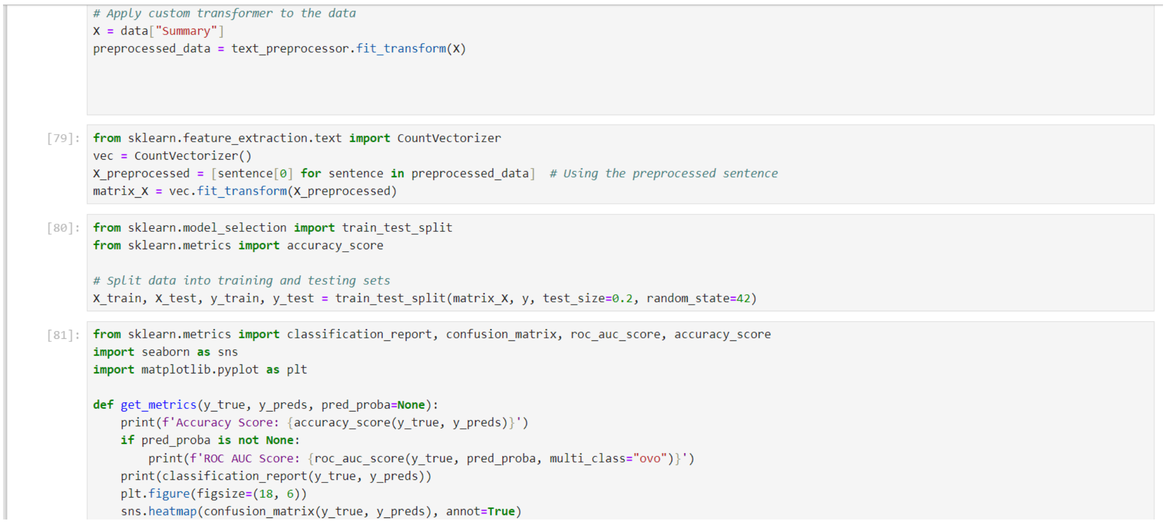


[77]:

```
import pandas as pd
from sklearn.base import BaseEstimator, TransformerMixin
from sklearn.feature_extraction.text import CountVectorizer
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import string

# Read the dataset
data = pd.read_csv("Dataset-SA.csv")

# Drop rows where both "Sentiment" and "Summary" are null
data.dropna(subset=["Sentiment", "Summary"], how="all", inplace=True)

# Fill missing values in "Summary" column with a placeholder string
data["Summary"].fillna("", inplace=True)

# Custom transformer class for text preprocessing
class TextPreprocessor(BaseEstimator, TransformerMixin):
    def __init__(self):
        pass

    def fit(self, X, y=None):
        return self

    def transform(self, X):
        preprocessed_sentences = []
        for sentence in X:
```



```
            # Convert text to lowercase
            sentence = sentence.lower()

            # Remove white spaces
            sentence = sentence.strip()

            # Remove punctuation
            sentence = sentence.translate(str.maketrans('', '', string.punctuation))

            # Remove stop words
            stop_words = set(stopwords.words('english'))
            word_tokens = word_tokenize(sentence)
            sentence = ' '.join([word for word in word_tokens if word not in stop_words])

            # Perform stemming
            stemmer = PorterStemmer()
            stemmed_sentence = ' '.join([stemmer.stem(word) for word in word_tokens])

            # Perform lemmatization
            lemmatizer = WordNetLemmatizer()
            lemmatized_sentence = ' '.join([lemmatizer.lemmatize(word) for word in word_tokens])

            preprocessed_sentences.append((sentence, stemmed_sentence, lemmatized_sentence))

        return preprocessed_sentences

# Initialize custom transformer
text_preprocessor = TextPreprocessor()
```



```
# Apply custom transformer to the data
X = data["Summary"]
preprocessed_data = text_preprocessor.fit_transform(X)
```

[79]:

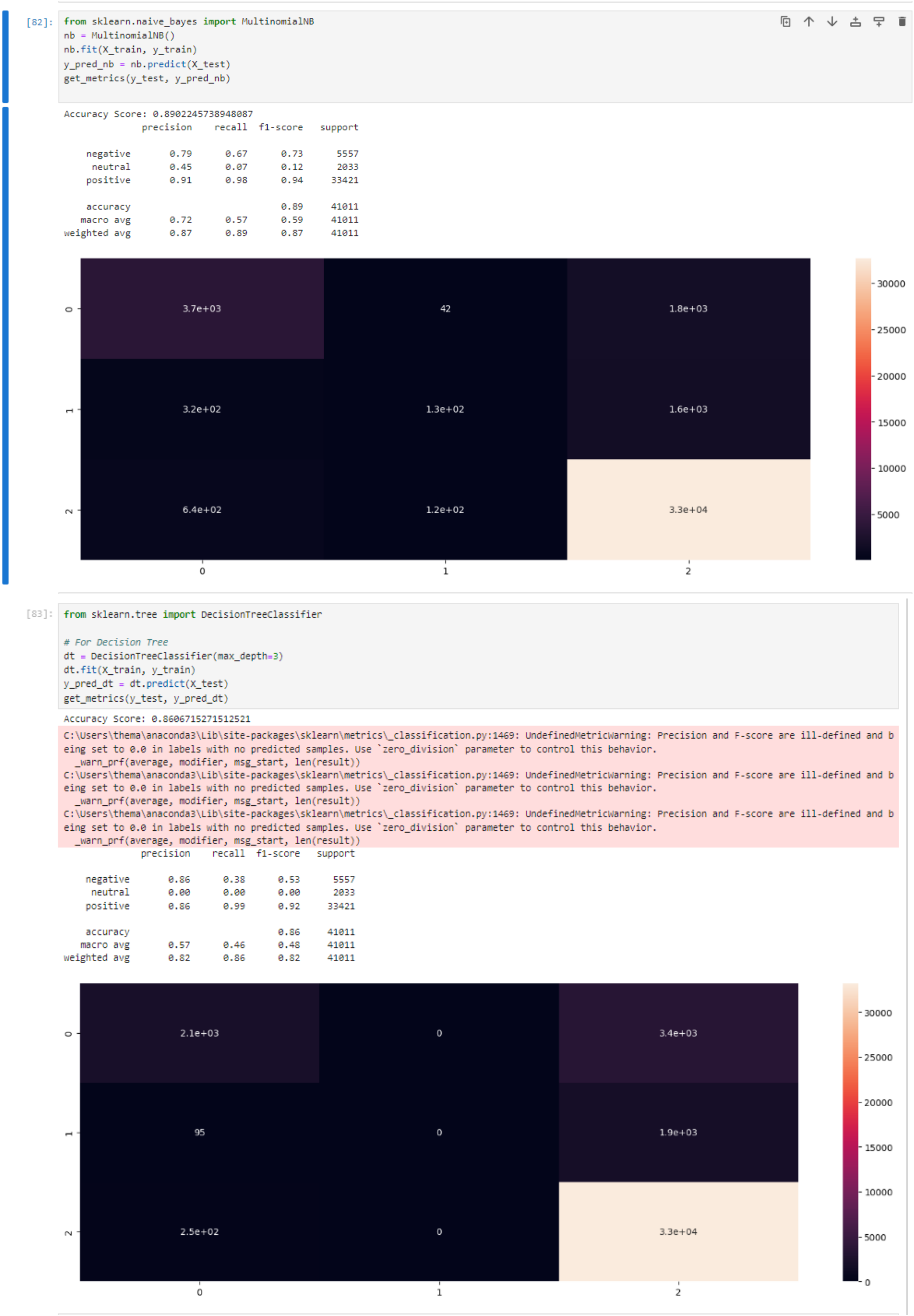
```
from sklearn.feature_extraction.text import CountVectorizer
vec = CountVectorizer()
X_preprocessed = [sentence[0] for sentence in preprocessed_data]  # Using the preprocessed sentence
matrix_X = vec.fit_transform(X_preprocessed)
```

[80]:

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(matrix_X, y, test_size=0.2, random_state=42)
```

[81]:

```
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score, accuracy_score
import seaborn as sns
import matplotlib.pyplot as plt

def get_metrics(y_true, y_preds, pred_proba=None):
    print(f'Accuracy Score: {accuracy_score(y_true, y_preds)}')
    if pred_proba is not None:
        print(f'ROC AUC Score: {roc_auc_score(y_true, pred_proba, multi_class="ovo")}')
    print(classification_report(y_true, y_preds))
    plt.figure(figsize=(18, 6))
    sns.heatmap(confusion_matrix(y_true, y_preds), annot=True)
```

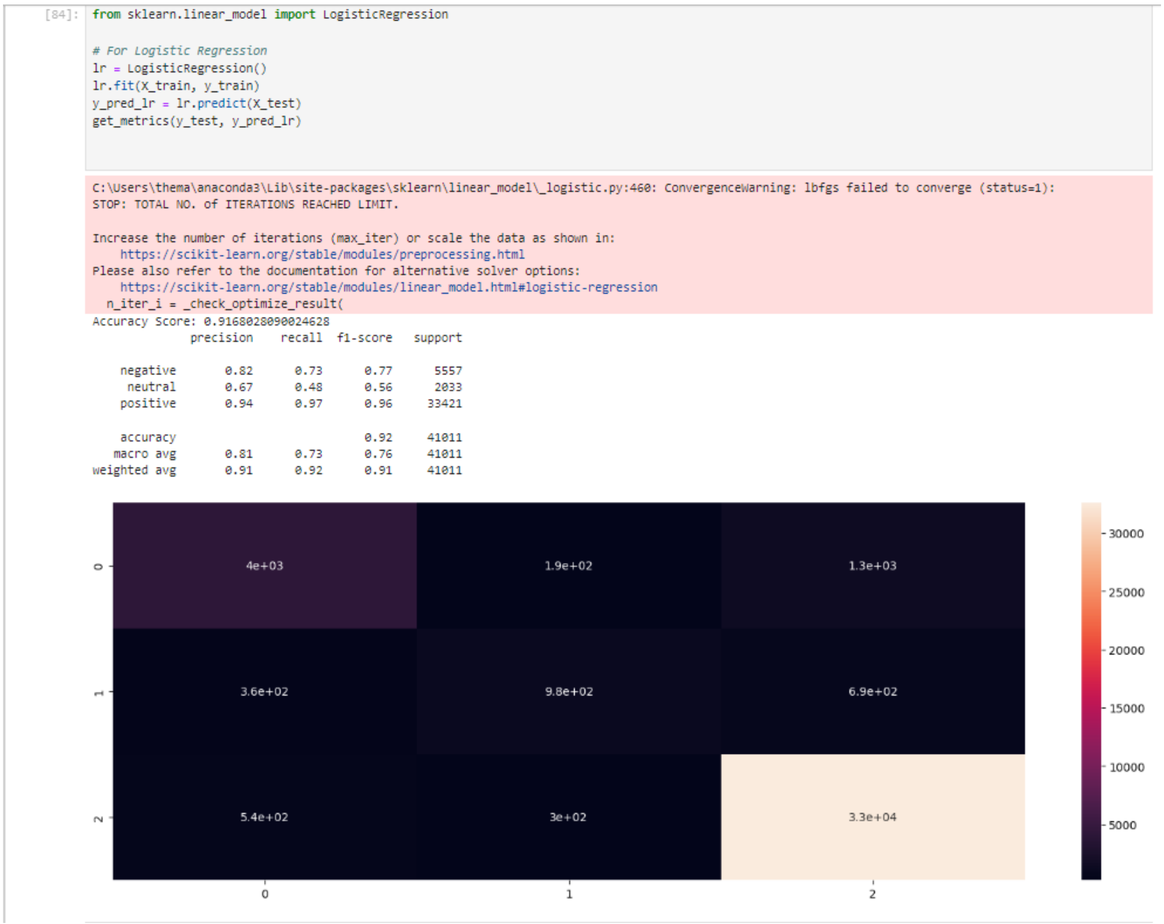


[82]:
```
from sklearn.naive_bayes import MultinomialNB
nb = MultinomialNB()
nb.fit(X_train, y_train)
y_pred_nb = nb.predict(X_test)
get_metrics(y_test, y_pred_nb)
```

```
Accuracy Score: 0.8902245738948087
              precision    recall  f1-score   support

    negative       0.79      0.67      0.73      5557
     neutral       0.45      0.07      0.12      2033
    positive       0.91      0.98      0.94     33421

    accuracy                           0.89     41011
   macro avg       0.72      0.57      0.59     41011
weighted avg       0.87      0.89      0.87     41011
```

[83]:
```
from sklearn.tree import DecisionTreeClassifier

# For Decision Tree
dt = DecisionTreeClassifier(max_depth=3)
dt.fit(X_train, y_train)
y_pred_dt = dt.predict(X_test)
get_metrics(y_test, y_pred_dt)
```

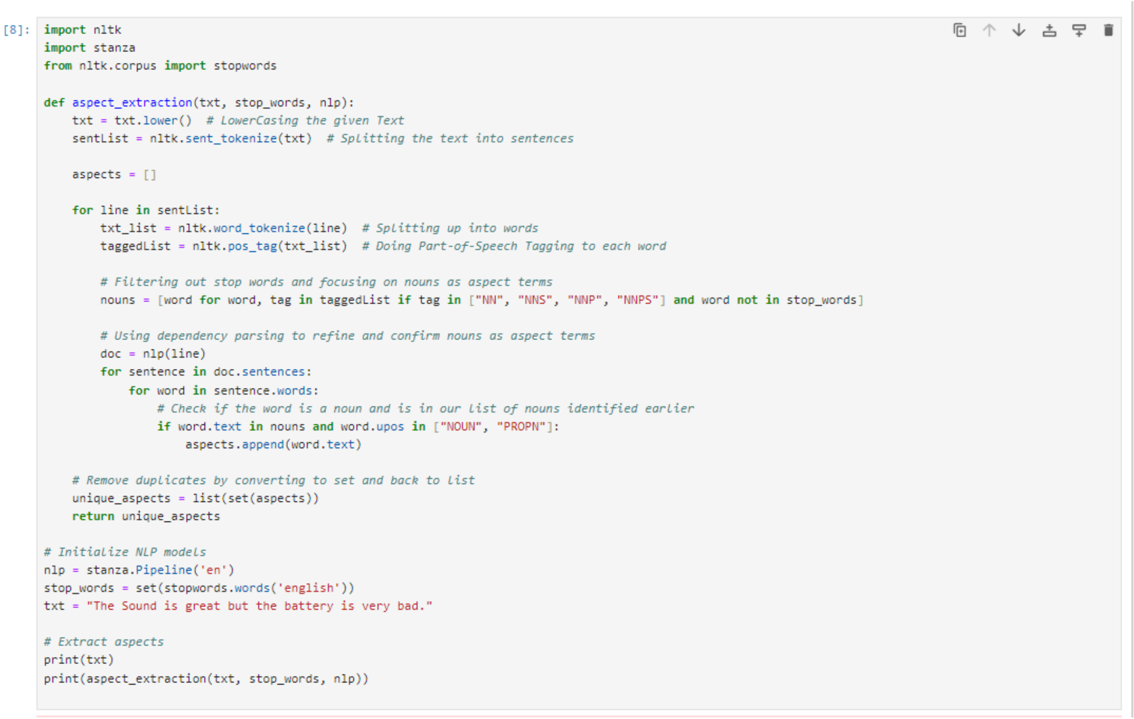
```
Accuracy Score: 0.8606715271512521
C:\Users\thema\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and b
eing set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\thema\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and b
eing set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\thema\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and b
eing set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

    negative       0.86      0.38      0.53      5557
     neutral       0.00      0.00      0.00      2033
    positive       0.86      0.99      0.92     33421

    accuracy                           0.86     41011
   macro avg       0.57      0.46      0.48     41011
weighted avg       0.82      0.86      0.82     41011
```



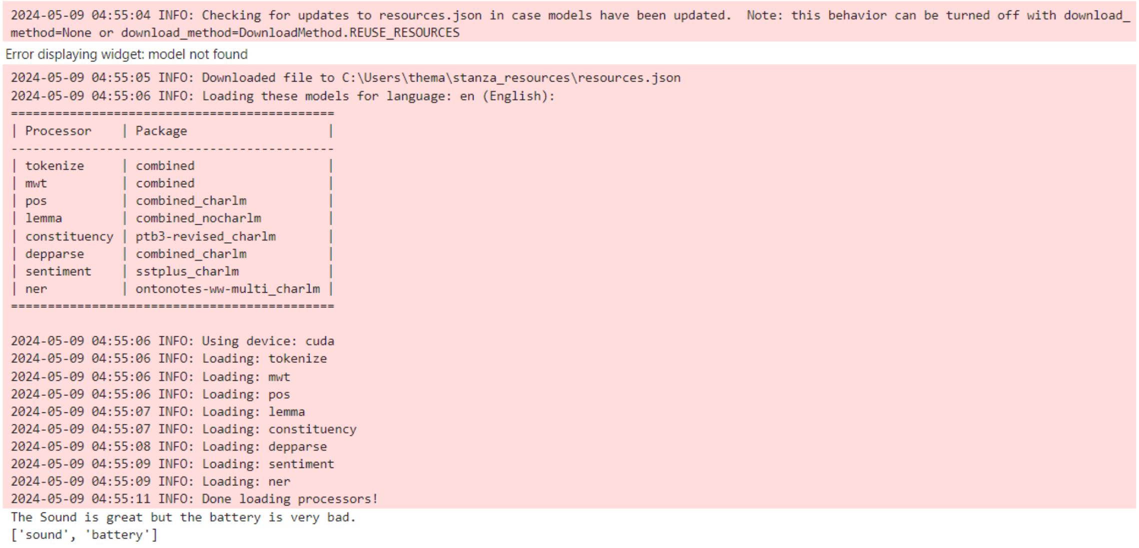
```
[84]: from sklearn.linear_model import LogisticRegression

      # For Logistic Regression
      lr = LogisticRegression()
      lr.fit(X_train, y_train)
      y_pred_lr = lr.predict(X_test)
      get_metrics(y_test, y_pred_lr)
```

```
C:\Users\thema\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:460: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
Accuracy Score: 0.9168028090024628
              precision    recall  f1-score   support

    negative       0.82      0.73      0.77      5557
     neutral       0.67      0.48      0.56      2033
    positive       0.94      0.97      0.96     33421

    accuracy                           0.92     41011
   macro avg       0.81      0.73      0.76     41011
weighted avg       0.91      0.92      0.91     41011
```

ASPECT TERM EXTRACTION



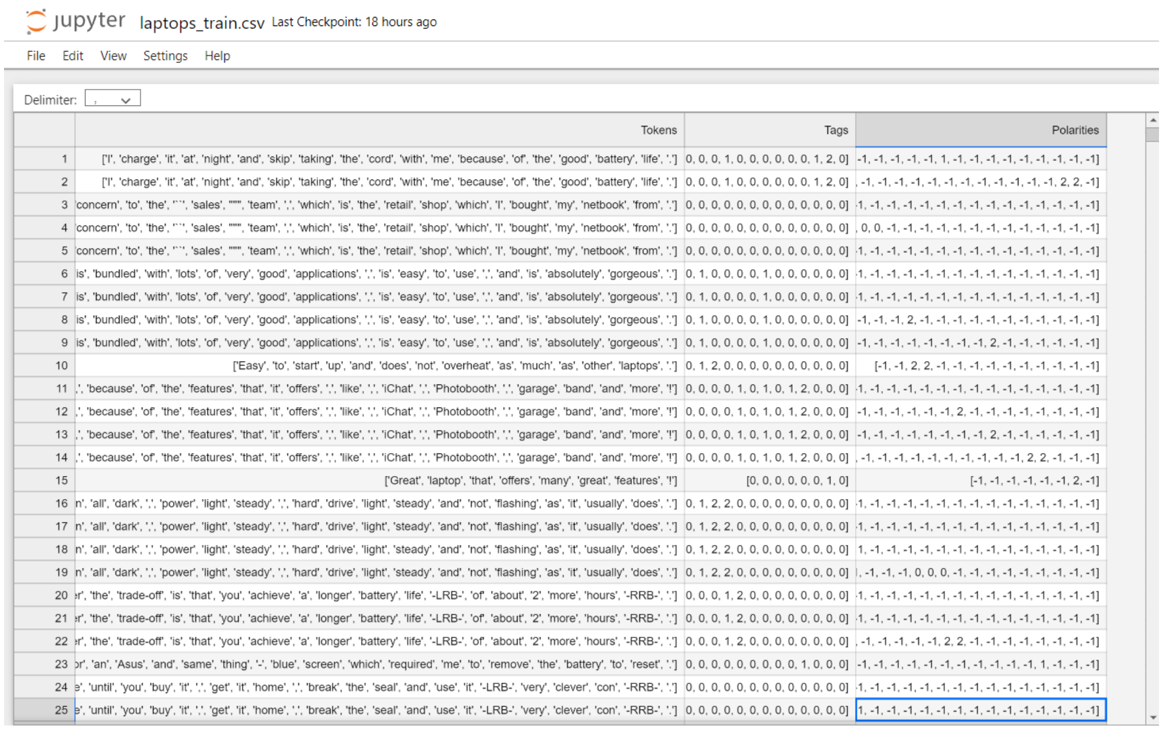
```
[8]: import nltk
     import stanza
     from nltk.corpus import stopwords

     def aspect_extraction(txt, stop_words, nlp):
         txt = txt.lower()  # LowerCasing the given Text
         sentList = nltk.sent_tokenize(txt)  # Splitting the text into sentences

         aspects = []

         for line in sentList:
             txt_list = nltk.word_tokenize(line)  # Splitting up into words
             taggedList = nltk.pos_tag(txt_list)  # Doing Part-of-Speech Tagging to each word

             # Filtering out stop words and focusing on nouns as aspect terms
             nouns = [word for word, tag in taggedList if tag in ["NN", "NNS", "NNP", "NNPS"] and word not in stop_words]

             # Using dependency parsing to refine and confirm nouns as aspect terms
             doc = nlp(line)
             for sentence in doc.sentences:
                 for word in sentence.words:
                     # Check if the word is a noun and is in our list of nouns identified earlier
                     if word.text in nouns and word.upos in ["NOUN", "PROPN"]:
                         aspects.append(word.text)

         # Remove duplicates by converting to set and back to list
         unique_aspects = list(set(aspects))
         return unique_aspects

     # Initialize NLP models
     nlp = stanza.Pipeline('en')
     stop_words = set(stopwords.words('english'))
     txt = "The Sound is great but the battery is very bad."

     # Extract aspects
     print(txt)
     print(aspect_extraction(txt, stop_words, nlp))
```



```
2024-05-09 04:55:04 INFO: Checking for updates to resources.json in case models have been updated.  Note: this behavior can be turned off with download_
method=None or download_method=DownloadMethod.REUSE_RESOURCES
Error displaying widget: model not found
2024-05-09 04:55:05 INFO: Downloaded file to C:\Users\thema\stanza_resources\resources.json
2024-05-09 04:55:06 INFO: Loading these models for language: en (English):
=============================================
| Processor     | Package                   |
---------------------------------------------
| tokenize      | combined                  |
| mwt           | combined                  |
| pos           | combined_charlm           |
| lemma         | combined_nocharlm         |
| constituency  | ptb3-revised_charlm       |
| depparse      | combined_charlm           |
| sentiment     | sstplus_charlm            |
| ner           | ontonotes-ww-multi_charlm |
=============================================

2024-05-09 04:55:06 INFO: Using device: cuda
2024-05-09 04:55:06 INFO: Loading: tokenize
2024-05-09 04:55:06 INFO: Loading: mwt
2024-05-09 04:55:06 INFO: Loading: pos
2024-05-09 04:55:07 INFO: Loading: lemma
2024-05-09 04:55:07 INFO: Loading: constituency
2024-05-09 04:55:08 INFO: Loading: depparse
2024-05-09 04:55:09 INFO: Loading: sentiment
2024-05-09 04:55:09 INFO: Loading: ner
2024-05-09 04:55:11 INFO: Done loading processors!
The Sound is great but the battery is very bad.
['sound', 'battery']
```

ASPECT BASED SENTIMENT ANALYSIS

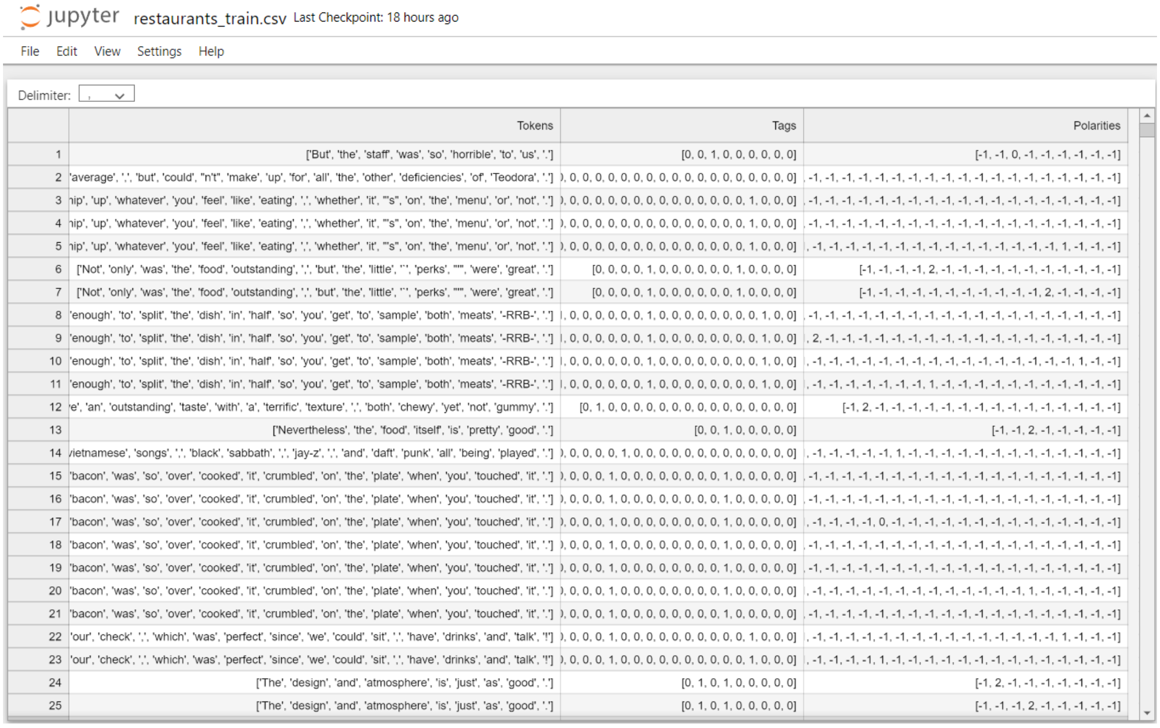
Datasets



Delimiter: ,

	Tokens	Tags	Polarities
1	['I', 'charge', 'it', 'at', 'night', 'and', 'skip', 'taking', 'the', 'cord', 'with', 'me', 'because', 'of', 'the', 'good', 'battery', 'life', '.']	0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 2, 0]	-1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
2	['I', 'charge', 'it', 'at', 'night', 'and', 'skip', 'taking', 'the', 'cord', 'with', 'me', 'because', 'of', 'the', 'good', 'battery', 'life', '.']	0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 2, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 2, 2, -1]
3	concern', 'to', 'the', '``', 'sales', "''", 'team', ',', 'which', 'is', 'the', 'retail', 'shop', 'which', 'I', 'bought', 'my', 'netbook', 'from', '.']	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
4	concern', 'to', 'the', '``', 'sales', "''", 'team', ',', 'which', 'is', 'the', 'retail', 'shop', 'which', 'I', 'bought', 'my', 'netbook', 'from', '.']	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	, 0, 0, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
5	concern', 'to', 'the', '``', 'sales', "''", 'team', ',', 'which', 'is', 'the', 'retail', 'shop', 'which', 'I', 'bought', 'my', 'netbook', 'from', '.']	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
6	is', 'bundled', 'with', 'lots', 'of', 'very', 'good', 'applications', ',', 'is', 'easy', 'to', 'use', ',', 'and', 'is', 'absolutely', 'gorgeous', '.']	0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]	1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
7	is', 'bundled', 'with', 'lots', 'of', 'very', 'good', 'applications', ',', 'is', 'easy', 'to', 'use', ',', 'and', 'is', 'absolutely', 'gorgeous', '.']	0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]	1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
8	is', 'bundled', 'with', 'lots', 'of', 'very', 'good', 'applications', ',', 'is', 'easy', 'to', 'use', ',', 'and', 'is', 'absolutely', 'gorgeous', '.']	0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]	-1, -1, -1, 2, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
9	is', 'bundled', 'with', 'lots', 'of', 'very', 'good', 'applications', ',', 'is', 'easy', 'to', 'use', ',', 'and', 'is', 'absolutely', 'gorgeous', '.']	0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]	-1, -1, -1, -1, -1, -1, -1, -1, 2, -1, -1, -1, -1, -1, -1]
10	['Easy', 'to', 'start', 'up', 'and', 'does', 'not', 'overheat', 'as', 'much', 'as', 'other', 'laptops', '.']	0, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[-1, -1, 2, 2, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
11	,', 'because', 'of', 'the', 'features', 'that', 'it', 'offers', ',', 'like', ',', 'iChat', ',', 'Photobooth', ',', 'garage', 'band', 'and', 'more', '!']	0, 0, 0, 0, 1, 0, 1, 0, 1, 2, 0, 0, 0]	1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
12	,', 'because', 'of', 'the', 'features', 'that', 'it', 'offers', ',', 'like', ',', 'iChat', ',', 'Photobooth', ',', 'garage', 'band', 'and', 'more', '!']	0, 0, 0, 0, 1, 0, 1, 0, 1, 2, 0, 0, 0]	-1, -1, -1, -1, -1, -1, 2, -1, -1, -1, -1, -1, -1, -1, -1]
13	,', 'because', 'of', 'the', 'features', 'that', 'it', 'offers', ',', 'like', ',', 'iChat', ',', 'Photobooth', ',', 'garage', 'band', 'and', 'more', '!']	0, 0, 0, 0, 1, 0, 1, 0, 1, 2, 0, 0, 0]	-1, -1, -1, -1, -1, -1, -1, -1, 2, -1, -1, -1, -1, -1, -1]
14	,', 'because', 'of', 'the', 'features', 'that', 'it', 'offers', ',', 'like', ',', 'iChat', ',', 'Photobooth', ',', 'garage', 'band', 'and', 'more', '!']	0, 0, 0, 0, 1, 0, 1, 0, 1, 2, 0, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 2, 2, -1, -1, -1]
15	['Great', 'laptop', 'that', 'offers', 'many', 'great', 'features', '!']	[0, 0, 0, 0, 0, 0, 1, 0]	[-1, -1, -1, -1, -1, -1, 2, -1]
16	n', 'all', 'dark', ',', 'power', 'light', 'steady', ',', 'hard', 'drive', 'light', 'steady', 'and', 'not', 'flashing', 'as', 'it', 'usually', 'does', '.']	0, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0]	1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
17	n', 'all', 'dark', ',', 'power', 'light', 'steady', ',', 'hard', 'drive', 'light', 'steady', 'and', 'not', 'flashing', 'as', 'it', 'usually', 'does', '.']	0, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0]	1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
18	n', 'all', 'dark', ',', 'power', 'light', 'steady', ',', 'hard', 'drive', 'light', 'steady', 'and', 'not', 'flashing', 'as', 'it', 'usually', 'does', '.']	0, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0]	1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
19	n', 'all', 'dark', ',', 'power', 'light', 'steady', ',', 'hard', 'drive', 'light', 'steady', 'and', 'not', 'flashing', 'as', 'it', 'usually', 'does', '.']	0, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0]	, -1, -1, -1, 0, 0, 0, -1, -1, -1, -1, -1, -1, -1, -1]
20	er', 'the', 'trade-off', 'is', 'that', 'you', 'achieve', 'a', 'longer', 'battery', 'life', '-LRB-', 'of', 'about', '2', 'more', 'hours', '-RRB-', '.']	0, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0]	1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
21	er', 'the', 'trade-off', 'is', 'that', 'you', 'achieve', 'a', 'longer', 'battery', 'life', '-LRB-', 'of', 'about', '2', 'more', 'hours', '-RRB-', '.']	0, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0]	1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
22	er', 'the', 'trade-off', 'is', 'that', 'you', 'achieve', 'a', 'longer', 'battery', 'life', '-LRB-', 'of', 'about', '2', 'more', 'hours', '-RRB-', '.']	0, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0]	, -1, -1, -1, -1, -1, 2, 2, -1, -1, -1, -1, -1, -1, -1]
23	or', 'an', 'Asus', 'and', 'same', 'thing', '-', 'blue', 'screen', 'which', 'required', 'me', 'to', 'remove', 'the', 'battery', 'to', 'reset', '.']	0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]	-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1]
24	e', 'until', 'you', 'buy', 'it', ',', 'get', 'it', 'home', ',', 'break', 'the', 'seal', 'and', 'use', 'it', '-LRB-', 'very', 'clever', 'con', '-RRB-', '.']	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
25	e', 'until', 'you', 'buy', 'it', ',', 'get', 'it', 'home', ',', 'break', 'the', 'seal', 'and', 'use', 'it', '-LRB-', 'very', 'clever', 'con', '-RRB-', '.']	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]

Fig 5.1 Laptops_train.csv Dataset

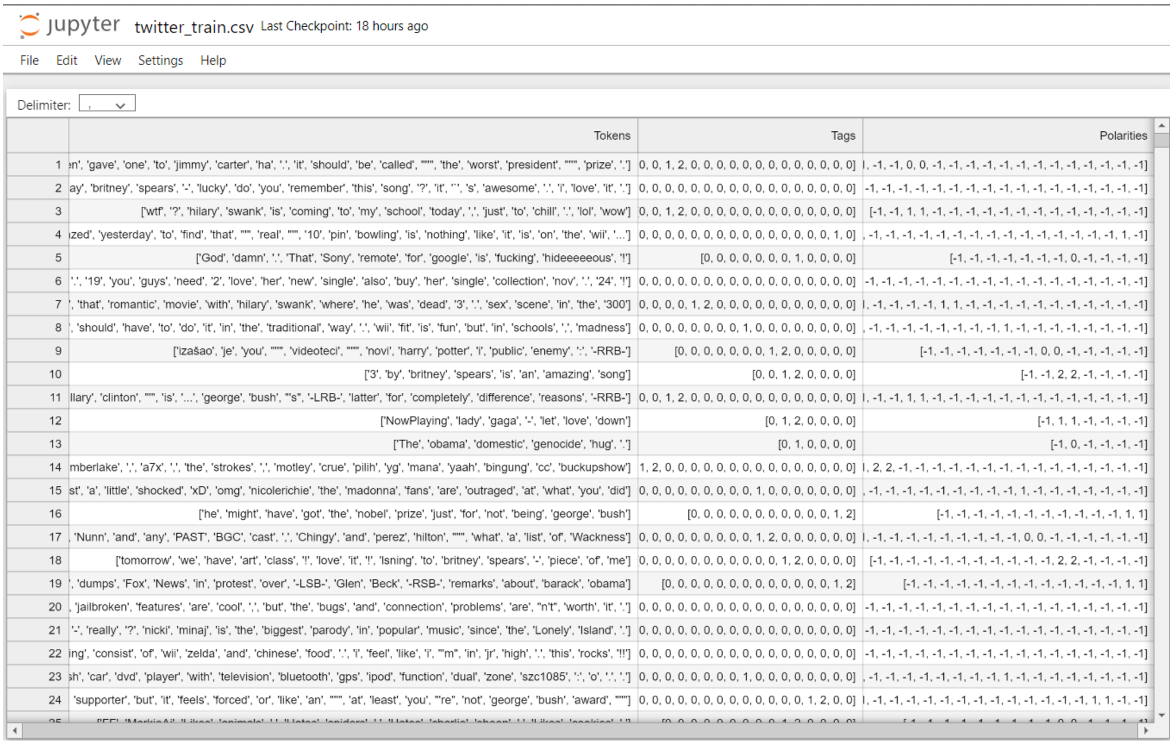


restaurants_train.csv Last Checkpoint: 18 hours ago

Delimiter: ,

	Tokens	Tags	Polarities
1	['But', 'the', 'staff', 'was', 'so', 'horrible', 'to', 'us', '.']	[0, 0, 1, 0, 0, 0, 0, 0, 0]	[-1, -1, 0, -1, -1, -1, -1, -1, -1]
2	'average', ',', 'but', 'could', "n't", 'make', 'up', 'for', 'all', 'the', 'other', 'deficiencies', 'of', 'Teodora', '.']	), 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
3	hip', 'up', 'whatever', 'you', 'feel', 'like', 'eating', ',', 'whether', 'it', "'s", 'on', 'the', 'menu', 'or', 'not', '.']	), 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
4	hip', 'up', 'whatever', 'you', 'feel', 'like', 'eating', ',', 'whether', 'it', "'s", 'on', 'the', 'menu', 'or', 'not', '.']	), 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
5	hip', 'up', 'whatever', 'you', 'feel', 'like', 'eating', ',', 'whether', 'it', "'s", 'on', 'the', 'menu', 'or', 'not', '.']	), 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1]
6	['Not', 'only', 'was', 'the', 'food', 'outstanding', ',', 'but', 'the', 'little', '``', 'perks', "''", 'were', 'great', '.']	[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]	[-1, -1, -1, -1, 2, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
7	['Not', 'only', 'was', 'the', 'food', 'outstanding', ',', 'but', 'the', 'little', '``', 'perks', "''", 'were', 'great', '.']	[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]	[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 2, -1, -1, -1]
8	'enough', 'to', 'split', 'the', 'dish', 'in', 'half', 'so', 'you', 'get', 'to', 'sample', 'both', 'meats', '-RRB-', '.']	, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
9	'enough', 'to', 'split', 'the', 'dish', 'in', 'half', 'so', 'you', 'get', 'to', 'sample', 'both', 'meats', '-RRB-', '.']	, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]	, 2, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
10	'enough', 'to', 'split', 'the', 'dish', 'in', 'half', 'so', 'you', 'get', 'to', 'sample', 'both', 'meats', '-RRB-', '.']	, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1]
11	'enough', 'to', 'split', 'the', 'dish', 'in', 'half', 'so', 'you', 'get', 'to', 'sample', 'both', 'meats', '-RRB-', '.']	, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
12	e', 'an', 'outstanding', 'taste', 'with', 'a', 'terrific', 'texture', ',', 'both', 'chewy', 'yet', 'not', 'gummy', '.']	[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[-1, 2, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
13	['Nevertheless', 'the', 'food', 'itself', 'is', 'pretty', 'good', '.']	[0, 0, 1, 0, 0, 0, 0, 0]	[-1, -1, 2, -1, -1, -1, -1, -1]
14	vietnamese', 'songs', ',', 'black', 'sabbath', ',', 'jay-z', ',', 'and', 'daft', 'punk', 'all', 'being', 'played', '.']	), 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
15	'bacon', 'was', 'so', 'over', 'cooked', 'it', 'crumbled', 'on', 'the', 'plate', 'when', 'you', 'touched', 'it', '.']	), 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
16	'bacon', 'was', 'so', 'over', 'cooked', 'it', 'crumbled', 'on', 'the', 'plate', 'when', 'you', 'touched', 'it', '.']	), 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
17	'bacon', 'was', 'so', 'over', 'cooked', 'it', 'crumbled', 'on', 'the', 'plate', 'when', 'you', 'touched', 'it', '.']	), 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]	, -1, -1, -1, -1, 0, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
18	'bacon', 'was', 'so', 'over', 'cooked', 'it', 'crumbled', 'on', 'the', 'plate', 'when', 'you', 'touched', 'it', '.']	), 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
19	'bacon', 'was', 'so', 'over', 'cooked', 'it', 'crumbled', 'on', 'the', 'plate', 'when', 'you', 'touched', 'it', '.']	), 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1]
20	'bacon', 'was', 'so', 'over', 'cooked', 'it', 'crumbled', 'on', 'the', 'plate', 'when', 'you', 'touched', 'it', '.']	), 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
21	'bacon', 'was', 'so', 'over', 'cooked', 'it', 'crumbled', 'on', 'the', 'plate', 'when', 'you', 'touched', 'it', '.']	), 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
22	'our', 'check', ',', 'which', 'was', 'perfect', 'since', 'we', 'could', 'sit', ',', 'have', 'drinks', 'and', 'talk', '!']	), 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1]
23	'our', 'check', ',', 'which', 'was', 'perfect', 'since', 'we', 'could', 'sit', ',', 'have', 'drinks', 'and', 'talk', '!']	), 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]	, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
24	['The', 'design', 'and', 'atmosphere', 'is', 'just', 'as', 'good', '.']	[0, 1, 0, 1, 0, 0, 0, 0, 0]	[-1, 2, -1, -1, -1, -1, -1, -1, -1]
25	['The', 'design', 'and', 'atmosphere', 'is', 'just', 'as', 'good', '.']	[0, 1, 0, 1, 0, 0, 0, 0, 0]	[-1, -1, -1, 2, -1, -1, -1, -1, -1]

Fig 5.2 Restaurants_train.csv Dataset



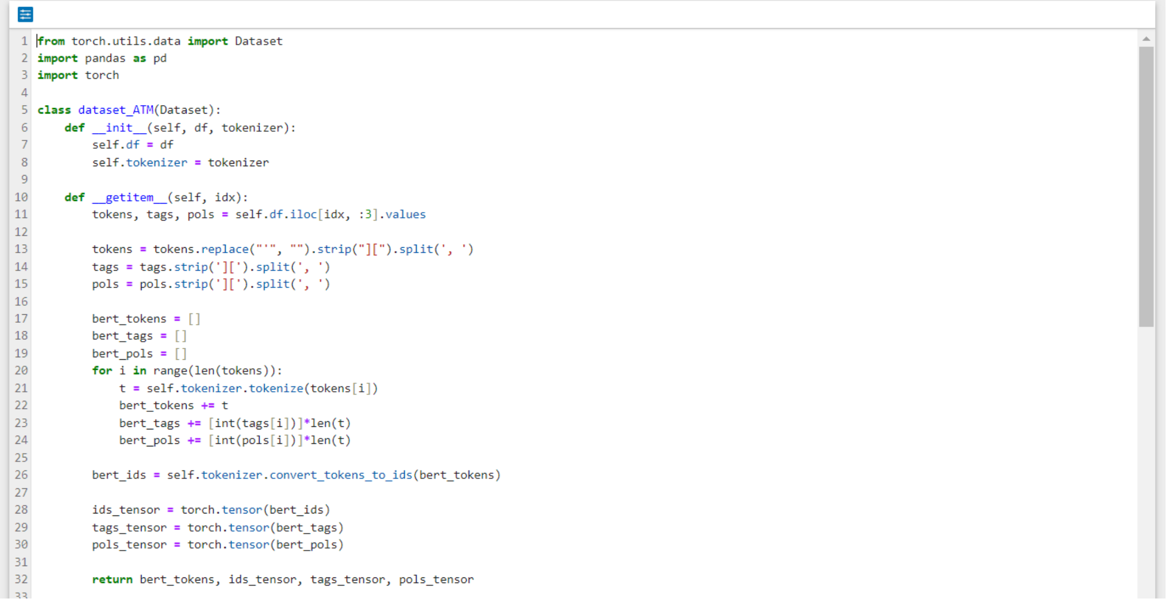
twitter_train.csv Last Checkpoint: 18 hours ago

Delimiter: ,

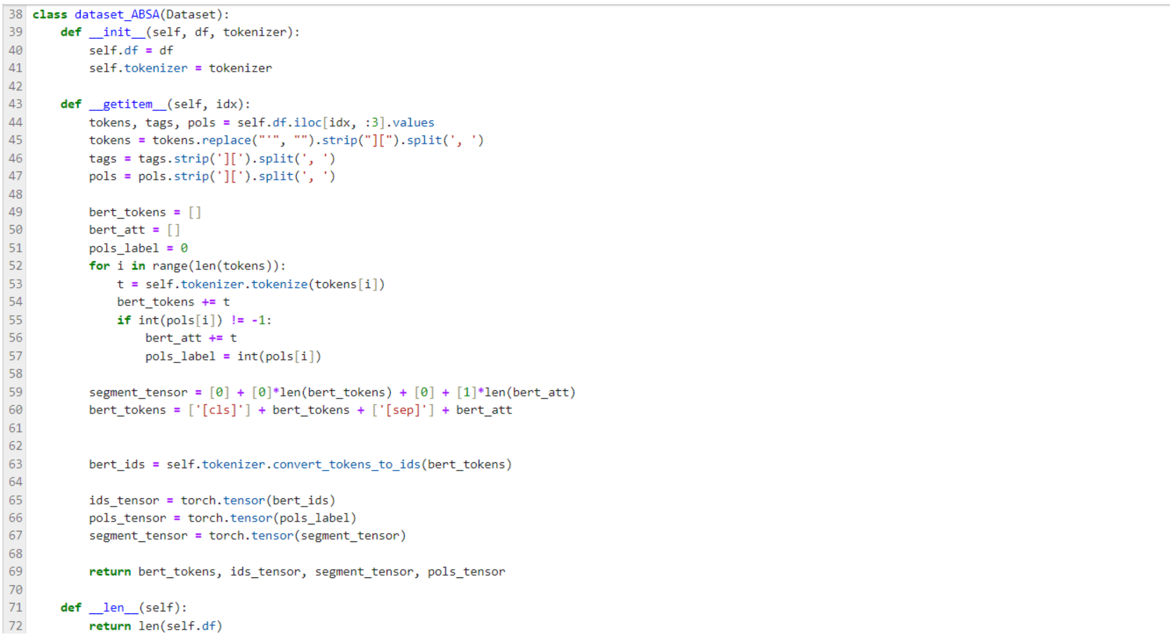
	Tokens	Tags	Polarities
1	en', 'gave', 'one', 'to', 'jimmy', 'carter', 'ha', '.', 'it', 'should', 'be', 'called', "''", 'the', 'worst', 'president', "''", 'prize', '.']	0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	, -1, -1, 0, 0, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
2	ay', 'britney', 'spears', '-', 'lucky', 'do', 'you', 'remember', 'this', 'song', '?', 'it', '`', 's', 'awesome', '.', 'i', 'love', 'it', '.']	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
3	['wtf', '?', 'hilary', 'swank', 'is', 'coming', 'to', 'my', 'school', 'today', ',', 'just', 'to', 'chill', '.', 'lol', 'wow']	[0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[-1, -1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
4	ized', 'yesterday', 'to', 'find', 'that', "''", 'real', "''", '10', 'pin', 'bowling', 'is', 'nothing', 'like', 'it', 'is', 'on', 'the', 'wii', '...']	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1]
5	['God', 'damn', '.', 'That', 'Sony', 'remote', 'for', 'google', 'is', 'fucking', 'hideeeeeous', '!']	[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]	[-1, -1, -1, -1, -1, -1, -1, 0, -1, -1, -1, -1]
6	'.', '19', 'you', 'guys', 'need', '2', 'love', 'her', 'new', 'single', 'also', 'buy', 'her', 'single', 'collection', 'nov', '.', '24', '!']	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
7	', 'that', 'romantic', 'movie', 'with', 'hilary', 'swank', 'where', 'he', 'was', 'dead', '3', '.', 'sex', 'scene', 'in', 'the', '300']	0, 0, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	, -1, -1, -1, -1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
8	', 'should', 'have', 'to', 'do', 'it', 'in', 'the', 'traditional', 'way', '.', 'wii', 'fit', 'is', 'fun', 'but', 'in', 'schools', ',', 'madness']	0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1]
9	['izašao', 'je', 'you', "''", 'videoteci', "''", 'novi', 'harry', 'potter', 'i', 'public', 'enemy', ':', '-RRB-']	[0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 0, 0]	[-1, -1, -1, -1, -1, -1, -1, 0, 0, -1, -1, -1, -1, -1]
10	['3', 'by', 'britney', 'spears', 'is', 'an', 'amazing', 'song']	[0, 0, 1, 2, 0, 0, 0, 0]	[-1, -1, 2, 2, -1, -1, -1, -1]
11	llary', 'clinton', "''", 'is', '...', 'george', 'bush', "'s", '-LRB-', 'latter', 'for', 'completely', 'difference', 'reasons', '-RRB-']	0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	, -1, -1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
12	['NowPlaying', 'lady', 'gaga', '-', 'let', 'love', 'down']	[0, 1, 2, 0, 0, 0, 0]	[-1, 1, 1, -1, -1, -1, -1]
13	['The', 'obama', 'domestic', 'genocide', 'hug', '.']	[0, 1, 0, 0, 0, 0]	[-1, 0, -1, -1, -1, -1]
14	mberlake', ',', 'a7x', ',', 'the', 'strokes', ',', 'motley', 'crue', 'pilih', 'yg', 'mana', 'yaah', 'bingung', 'cc', 'buckupshow']	1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	, 2, 2, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
15	st', 'a', 'little', 'shocked', 'xD', 'omg', 'nicolerichie', 'the', 'madonna', 'fans', 'are', 'outraged', 'at', 'what', 'you', 'did']	0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1]
16	['he', 'might', 'have', 'got', 'the', 'nobel', 'prize', 'just', 'for', 'not', 'being', 'george', 'bush']	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2]	[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1]
17	, 'Nunn', 'and', 'any', 'PAST', 'BGC', 'cast', ',', 'Chingy', 'and', 'perez', 'hilton', "''", 'what', 'a', 'list', 'of', 'Wackness']	0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, -1, 0, 0, -1, -1, -1, -1, -1]
18	['tomorrow', 'we', 'have', 'art', 'class', '!', 'love', 'it', '!', 'lsning', 'to', 'britney', 'spears', '-', 'piece', 'of', 'me']	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 0]	[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 2, 2, -1, -1, -1, -1]
19	', 'dumps', 'Fox', 'News', 'in', 'protest', 'over', '-LSB-', 'Glen', 'Beck', '-RSB-', 'remarks', 'about', 'barack', 'obama']	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2]	[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1]
20	, 'jailbroken', 'features', 'are', 'cool', ',', 'but', 'the', 'bugs', 'and', 'connection', 'problems', 'are', "n't", 'worth', 'it', '.']	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
21	'-', 'really', '?', 'nicki', 'minaj', 'is', 'the', 'biggest', 'parody', 'in', 'popular', 'music', 'since', 'the', 'Lonely', 'Island', '.']	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
22	ing', 'consist', 'of', 'wii', 'zelda', 'and', 'chinese', 'food', '.', 'i', 'feel', 'like', 'i', "'m", 'in', 'jr', 'high', '.', 'this', 'rocks', '!!']	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
23	sh', 'car', 'dvd', 'player', 'with', 'television', 'bluetooth', 'gps', 'ipod', 'function', 'dual', 'zone', 'szc1085', ':', 'o', '.', '.']	0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1]
24	'supporter', 'but', 'it', 'feels', 'forced', 'or', 'like', 'an', "''", 'at', 'least', 'you', "'re", 'not', 'george', 'bush', 'award', "''"]	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 0]	, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, -1, -1]
25	['EE', 'MarkieAi', 'Likes', 'animals', '.', 'Hates', 'spiders', '.', 'Hates', 'charlie', 'sheen', '.', 'Likes', 'cookies', '.']	[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 0, 0, 0]	[-1, -1, -1, -1, -1, -1, -1, -1, -1, 0, 0, -1, -1, -1, -1]

Fig 5.3 Twitter_train.csv Dataset

Dataset.py

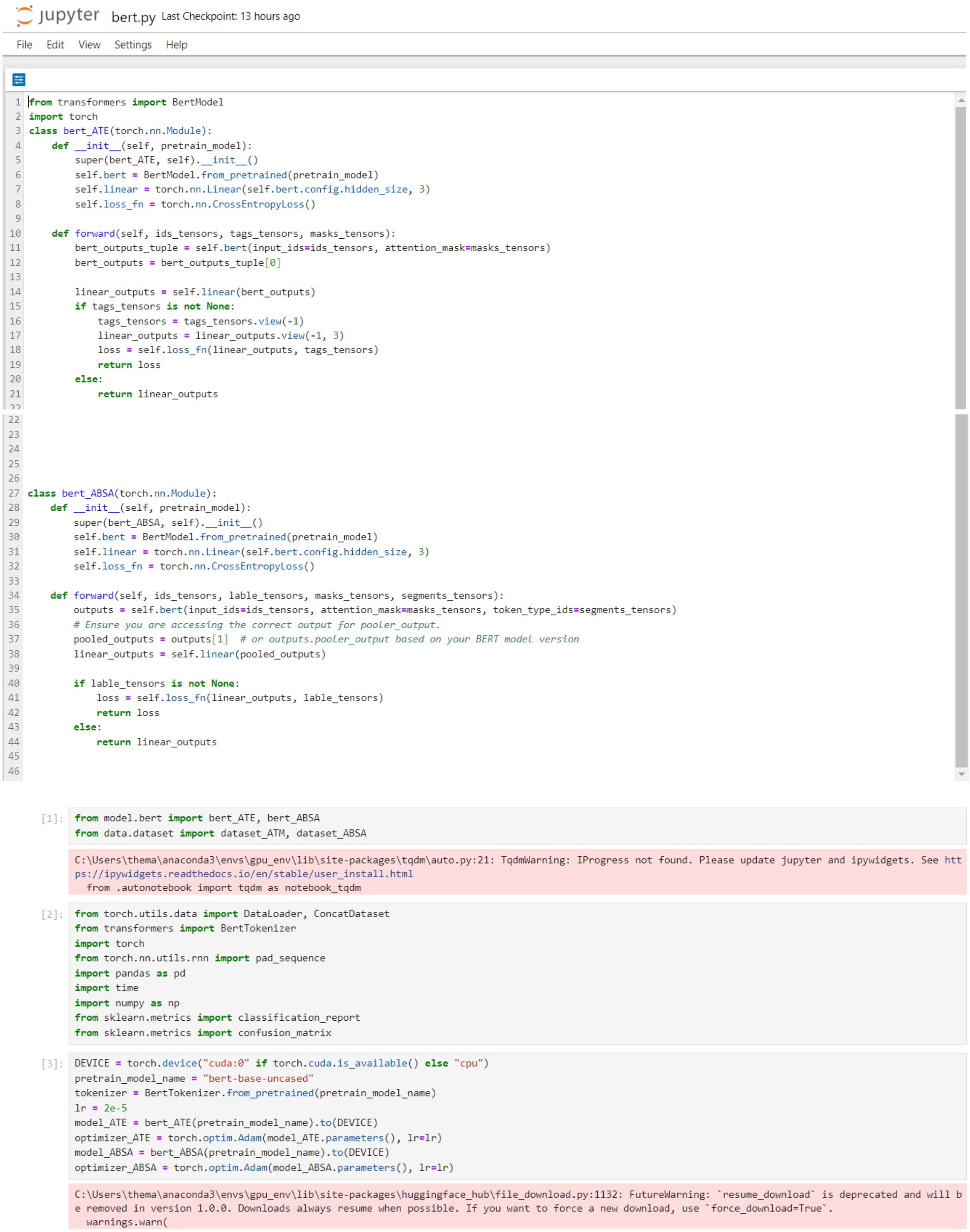


```python
from torch.utils.data import Dataset
import pandas as pd
import torch

class dataset_ATM(Dataset):
    def __init__(self, df, tokenizer):
        self.df = df
        self.tokenizer = tokenizer

    def __getitem__(self, idx):
        tokens, tags, pols = self.df.iloc[idx, :3].values

        tokens = tokens.replace("'", "").strip("][").split(', ')
        tags = tags.strip('][').split(', ')
        pols = pols.strip('][').split(', ')

        bert_tokens = []
        bert_tags = []
        bert_pols = []
        for i in range(len(tokens)):
            t = self.tokenizer.tokenize(tokens[i])
            bert_tokens += t
            bert_tags += [int(tags[i])]*len(t)
            bert_pols += [int(pols[i])]*len(t)

        bert_ids = self.tokenizer.convert_tokens_to_ids(bert_tokens)

        ids_tensor = torch.tensor(bert_ids)
        tags_tensor = torch.tensor(bert_tags)
        pols_tensor = torch.tensor(bert_pols)

        return bert_tokens, ids_tensor, tags_tensor, pols_tensor
```



```python
class dataset_ABSA(Dataset):
    def __init__(self, df, tokenizer):
        self.df = df
        self.tokenizer = tokenizer

    def __getitem__(self, idx):
        tokens, tags, pols = self.df.iloc[idx, :3].values
        tokens = tokens.replace("'", "").strip("][").split(', ')
        tags = tags.strip('][').split(', ')
        pols = pols.strip('][').split(', ')

        bert_tokens = []
        bert_att = []
        pols_label = 0
        for i in range(len(tokens)):
            t = self.tokenizer.tokenize(tokens[i])
            bert_tokens += t
            if int(pols[i]) != -1:
                bert_att += t
                pols_label = int(pols[i])

        segment_tensor = [0] + [0]*len(bert_tokens) + [0] + [1]*len(bert_att)
        bert_tokens = ['[cls]'] + bert_tokens + ['[sep]'] + bert_att


        bert_ids = self.tokenizer.convert_tokens_to_ids(bert_tokens)

        ids_tensor = torch.tensor(bert_ids)
        pols_tensor = torch.tensor(pols_label)
        segment_tensor = torch.tensor(segment_tensor)

        return bert_tokens, ids_tensor, segment_tensor, pols_tensor

    def __len__(self):
        return len(self.df)
```

Bert.py



jupyter
bert.py Last Checkpoint: 13 hours ago
File Edit View Settings Help

```
from transformers import BertModel
import torch
class bert_ATE(torch.nn.Module):
    def __init__(self, pretrain_model):
        super(bert_ATE, self).__init__()
        self.bert = BertModel.from_pretrained(pretrain_model)
        self.linear = torch.nn.Linear(self.bert.config.hidden_size, 3)
        self.loss_fn = torch.nn.CrossEntropyLoss()

    def forward(self, ids_tensors, tags_tensors, masks_tensors):
        bert_outputs_tuple = self.bert(input_ids=ids_tensors, attention_mask=masks_tensors)
        bert_outputs = bert_outputs_tuple[0]

        linear_outputs = self.linear(bert_outputs)
        if tags_tensors is not None:
            tags_tensors = tags_tensors.view(-1)
            linear_outputs = linear_outputs.view(-1, 3)
            loss = self.loss_fn(linear_outputs, tags_tensors)
            return loss
        else:
            return linear_outputs






class bert_ABSA(torch.nn.Module):
    def __init__(self, pretrain_model):
        super(bert_ABSA, self).__init__()
        self.bert = BertModel.from_pretrained(pretrain_model)
        self.linear = torch.nn.Linear(self.bert.config.hidden_size, 3)
        self.loss_fn = torch.nn.CrossEntropyLoss()

    def forward(self, ids_tensors, lable_tensors, masks_tensors, segments_tensors):
        outputs = self.bert(input_ids=ids_tensors, attention_mask=masks_tensors, token_type_ids=segments_tensors)
        # Ensure you are accessing the correct output for pooler_output.
        pooled_outputs = outputs[1]  # or outputs.pooler_output based on your BERT model version
        linear_outputs = self.linear(pooled_outputs)

        if lable_tensors is not None:
            loss = self.loss_fn(linear_outputs, lable_tensors)
            return loss
        else:
            return linear_outputs
```

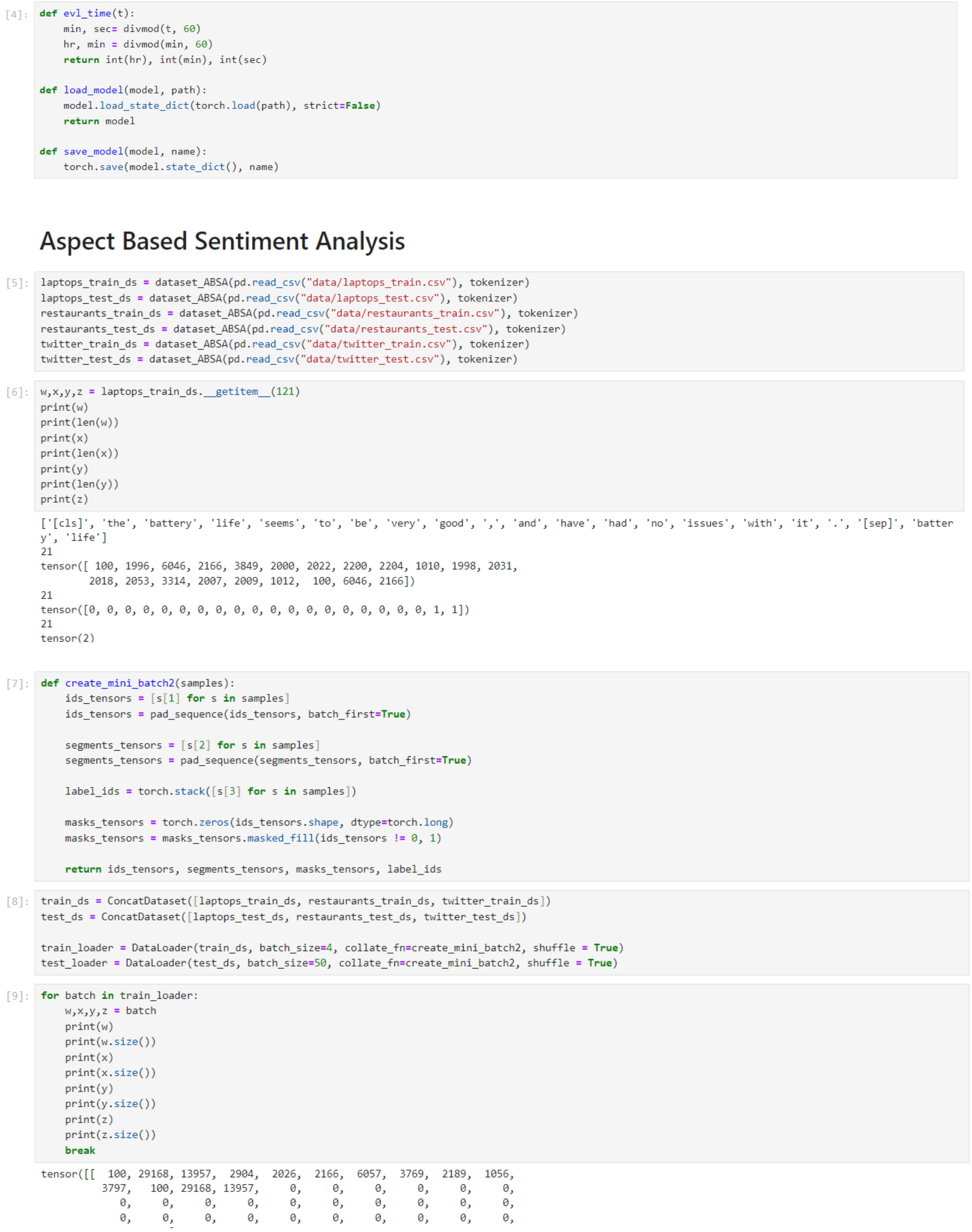
```
[1]: from model.bert import bert_ATE, bert_ABSA
     from data.dataset import dataset_ATM, dataset_ABSA
```

```
C:\Users\thema\anaconda3\envs\gpu_env\lib\site-packages\tqdm\auto.py:21: TqdmWarning: IProgress not found. Please update jupyter and ipywidgets. See htt
ps://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
```

```
[2]: from torch.utils.data import DataLoader, ConcatDataset
     from transformers import BertTokenizer
     import torch
     from torch.nn.utils.rnn import pad_sequence
     import pandas as pd
     import time
     import numpy as np
     from sklearn.metrics import classification_report
     from sklearn.metrics import confusion_matrix
```

```
[3]: DEVICE = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
     pretrain_model_name = "bert-base-uncased"
     tokenizer = BertTokenizer.from_pretrained(pretrain_model_name)
     lr = 2e-5
     model_ATE = bert_ATE(pretrain_model_name).to(DEVICE)
     optimizer_ATE = torch.optim.Adam(model_ATE.parameters(), lr=lr)
     model_ABSA = bert_ABSA(pretrain_model_name).to(DEVICE)
     optimizer_ABSA = torch.optim.Adam(model_ABSA.parameters(), lr=lr)
```

```
C:\Users\thema\anaconda3\envs\gpu_env\lib\site-packages\huggingface_hub\file_download.py:1132: FutureWarning: `resume_download` is deprecated and will b
e removed in version 1.0.0. Downloads always resume when possible. If you want to force a new download, use `force_download=True`.
  warnings.warn(
```



[4]:

```
def evl_time(t):
    min, sec= divmod(t, 60)
    hr, min = divmod(min, 60)
    return int(hr), int(min), int(sec)

def load_model(model, path):
    model.load_state_dict(torch.load(path), strict=False)
    return model

def save_model(model, name):
    torch.save(model.state_dict(), name)
```

Aspect Based Sentiment Analysis

[5]:

```
laptops_train_ds = dataset_ABSA(pd.read_csv("data/laptops_train.csv"), tokenizer)
laptops_test_ds = dataset_ABSA(pd.read_csv("data/laptops_test.csv"), tokenizer)
restaurants_train_ds = dataset_ABSA(pd.read_csv("data/restaurants_train.csv"), tokenizer)
restaurants_test_ds = dataset_ABSA(pd.read_csv("data/restaurants_test.csv"), tokenizer)
twitter_train_ds = dataset_ABSA(pd.read_csv("data/twitter_train.csv"), tokenizer)
twitter_test_ds = dataset_ABSA(pd.read_csv("data/twitter_test.csv"), tokenizer)
```

[6]:

```
w,x,y,z = laptops_train_ds.__getitem__(121)
print(w)
print(len(w))
print(x)
print(len(x))
print(y)
print(len(y))
print(z)
```

```
['[cls]', 'the', 'battery', 'life', 'seems', 'to', 'be', 'very', 'good', ',', 'and', 'have', 'had', 'no', 'issues', 'with', 'it', '.', '[sep]', 'batter
y', 'life']
21
tensor([ 100, 1996, 6046, 2166, 3849, 2000, 2022, 2200, 2204, 1010, 1998, 2031,
        2018, 2053, 3314, 2007, 2009, 1012,  100, 6046, 2166])
21
tensor([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1])
21
tensor(2)
```

[7]:

```
def create_mini_batch2(samples):
    ids_tensors = [s[1] for s in samples]
    ids_tensors = pad_sequence(ids_tensors, batch_first=True)

    segments_tensors = [s[2] for s in samples]
    segments_tensors = pad_sequence(segments_tensors, batch_first=True)

    label_ids = torch.stack([s[3] for s in samples])

    masks_tensors = torch.zeros(ids_tensors.shape, dtype=torch.long)
    masks_tensors = masks_tensors.masked_fill(ids_tensors != 0, 1)

    return ids_tensors, segments_tensors, masks_tensors, label_ids
```

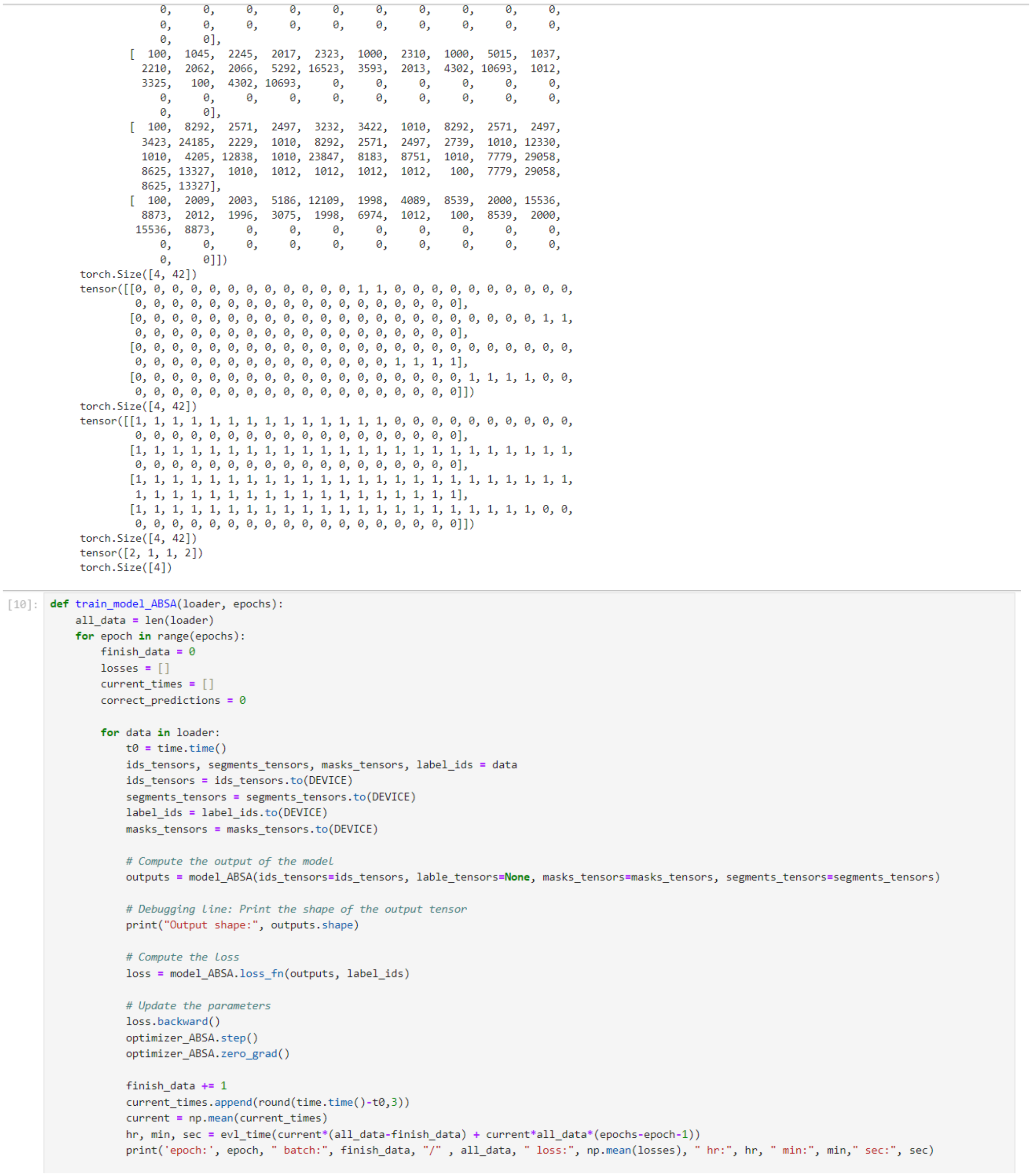
[8]:

```
train_ds = ConcatDataset([laptops_train_ds, restaurants_train_ds, twitter_train_ds])
test_ds = ConcatDataset([laptops_test_ds, restaurants_test_ds, twitter_test_ds])

train_loader = DataLoader(train_ds, batch_size=4, collate_fn=create_mini_batch2, shuffle = True)
test_loader = DataLoader(test_ds, batch_size=50, collate_fn=create_mini_batch2, shuffle = True)
```

[9]:

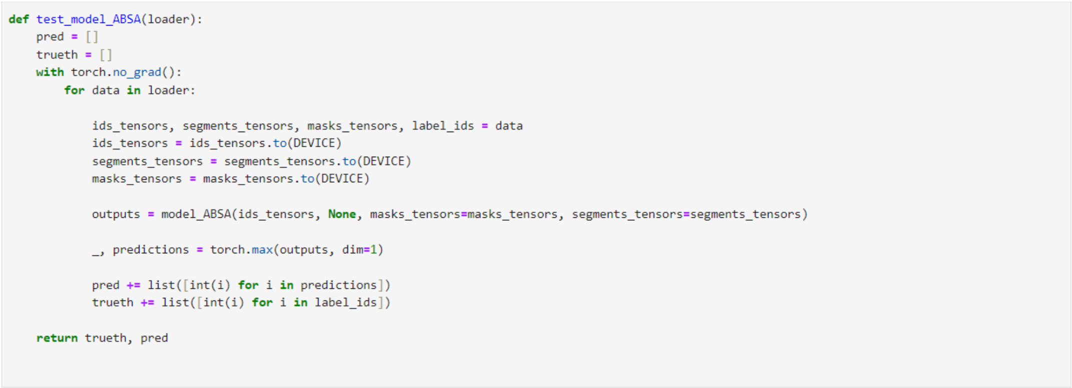
```
for batch in train_loader:
    w,x,y,z = batch
    print(w)
    print(w.size())
    print(x)
    print(x.size())
    print(y)
    print(y.size())
    print(z)
    print(z.size())
    break
```

```
tensor([[  100, 29168, 13957,  2904,  2026,  2166,  6057,  3769,  2189,  1056,
          3797,   100, 29168, 13957,     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,
```

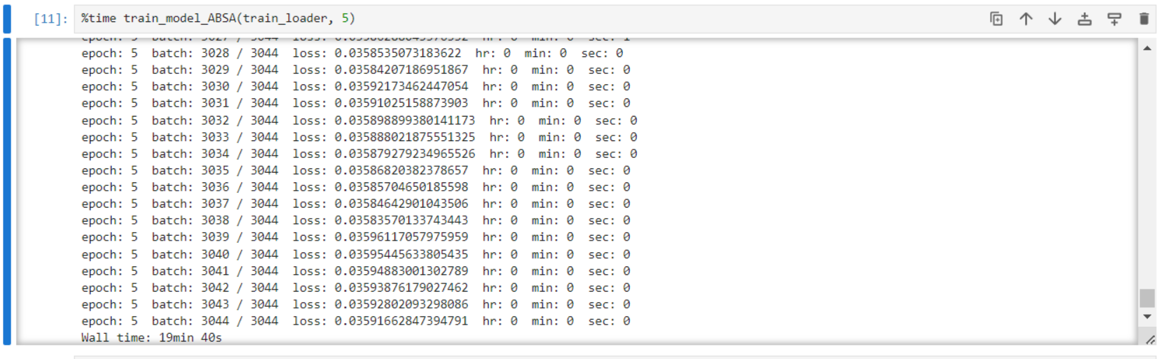


```
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0],
        [  100,  1045,  2245,  2017,  2323,  1000,  2310,  1000,  5015,  1037,
          2210,  2062,  2066,  5292, 16523,  3593,  2013,  4302, 10693,  1012,
          3325,   100,  4302, 10693,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0],
        [  100,  8292,  2571,  2497,  3232,  3422,  1010,  8292,  2571,  2497,
          3423, 24185,  2229,  1010,  8292,  2571,  2497,  2739,  1010, 12330,
          1010,  4205, 12838,  1010, 23847,  8183,  8751,  1010,  7779, 29058,
          8625, 13327,  1010,  1012,  1012,  1012,  1012,   100,  7779, 29058,
          8625, 13327],
        [  100,  2009,  2003,  5186, 12109,  1998,  4089,  8539,  2000, 15536,
          8873,  2012,  1996,  3075,  1998,  6974,  1012,   100,  8539,  2000,
         15536,  8873,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0]])
torch.Size([4, 42])
tensor([[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
         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, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1],
        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]])
torch.Size([4, 42])
tensor([[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 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],
        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
         1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]])
torch.Size([4, 42])
tensor([2, 1, 1, 2])
torch.Size([4])
```

```
[10]: def train_model_ABSA(loader, epochs):
          all_data = len(loader)
          for epoch in range(epochs):
              finish_data = 0
              losses = []
              current_times = []
              correct_predictions = 0

              for data in loader:
                  t0 = time.time()
                  ids_tensors, segments_tensors, masks_tensors, label_ids = data
                  ids_tensors = ids_tensors.to(DEVICE)
                  segments_tensors = segments_tensors.to(DEVICE)
                  label_ids = label_ids.to(DEVICE)
                  masks_tensors = masks_tensors.to(DEVICE)

                  # Compute the output of the model
                  outputs = model_ABSA(ids_tensors=ids_tensors, lable_tensors=None, masks_tensors=masks_tensors, segments_tensors=segments_tensors)

                  # Debugging line: Print the shape of the output tensor
                  print("Output shape:", outputs.shape)

                  # Compute the loss
                  loss = model_ABSA.loss_fn(outputs, label_ids)

                  # Update the parameters
                  loss.backward()
                  optimizer_ABSA.step()
                  optimizer_ABSA.zero_grad()

                  finish_data += 1
                  current_times.append(round(time.time()-t0,3))
                  current = np.mean(current_times)
                  hr, min, sec = evl_time(current*(all_data-finish_data) + current*all_data*(epochs-epoch-1))
                  print('epoch:', epoch, " batch:", finish_data, "/" , all_data, " loss:", np.mean(losses), " hr:", hr, " min:", min," sec:", sec)
```



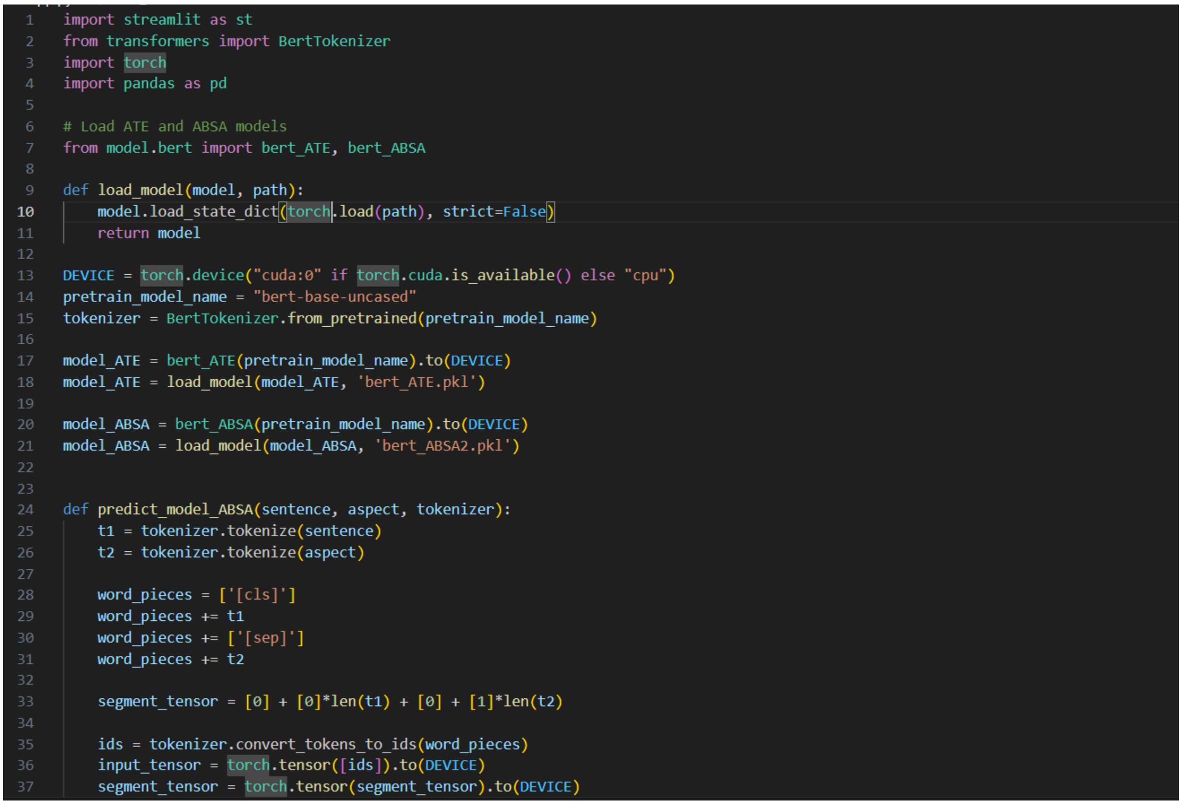
```
def test_model_ABSA(loader):
    pred = []
    trueth = []
    with torch.no_grad():
        for data in loader:

            ids_tensors, segments_tensors, masks_tensors, label_ids = data
            ids_tensors = ids_tensors.to(DEVICE)
            segments_tensors = segments_tensors.to(DEVICE)
            masks_tensors = masks_tensors.to(DEVICE)

            outputs = model_ABSA(ids_tensors, None, masks_tensors=masks_tensors, segments_tensors=segments_tensors)

            _, predictions = torch.max(outputs, dim=1)

            pred += list([int(i) for i in predictions])
            trueth += list([int(i) for i in label_ids])

    return trueth, pred
```



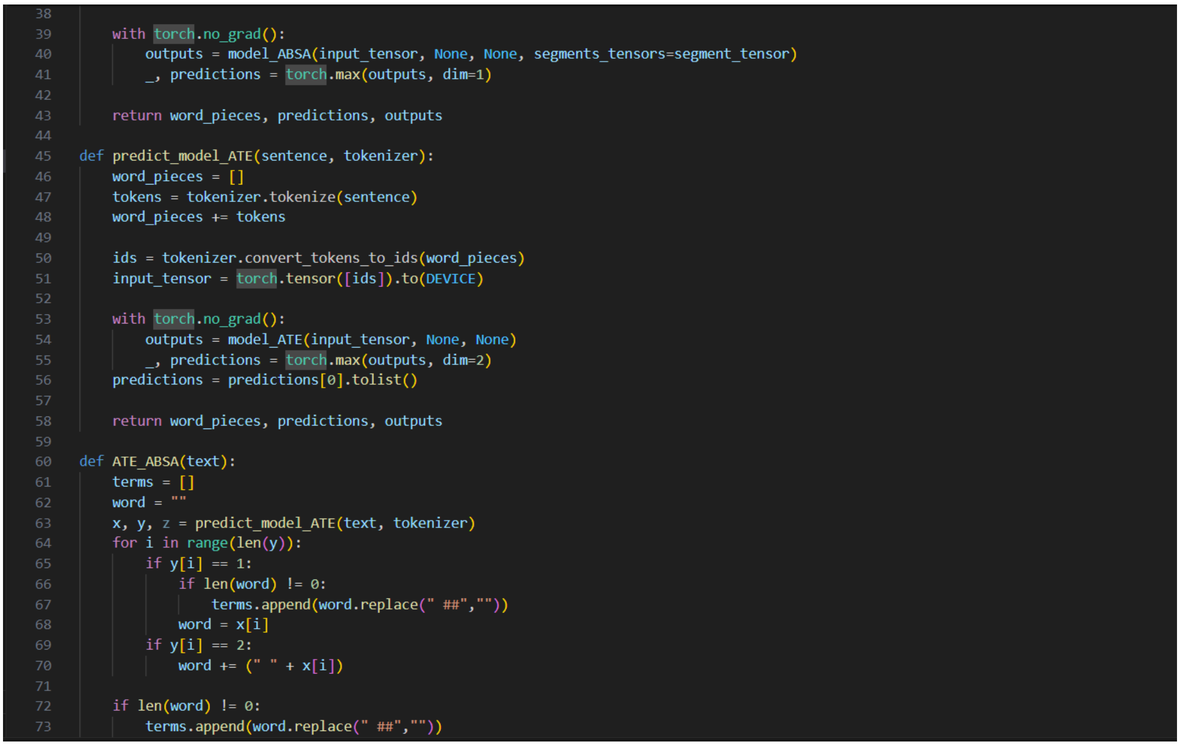
```
[11]: %time train_model_ABSA(train_loader, 5)
```

```
epoch: 5  batch: 3028 / 3044  loss: 0.0358535073183622  hr: 0  min: 0  sec: 0
epoch: 5  batch: 3029 / 3044  loss: 0.0358420718695l867  hr: 0  min: 0  sec: 0
epoch: 5  batch: 3030 / 3044  loss: 0.03592173462447054  hr: 0  min: 0  sec: 0
epoch: 5  batch: 3031 / 3044  loss: 0.03591025158873903  hr: 0  min: 0  sec: 0
epoch: 5  batch: 3032 / 3044  loss: 0.035898899380141173  hr: 0  min: 0  sec: 0
epoch: 5  batch: 3033 / 3044  loss: 0.035888021875551325  hr: 0  min: 0  sec: 0
epoch: 5  batch: 3034 / 3044  loss: 0.035879279234965526  hr: 0  min: 0  sec: 0
epoch: 5  batch: 3035 / 3044  loss: 0.03586820382378657  hr: 0  min: 0  sec: 0
epoch: 5  batch: 3036 / 3044  loss: 0.03585704650185598  hr: 0  min: 0  sec: 0
epoch: 5  batch: 3037 / 3044  loss: 0.03584642901043506  hr: 0  min: 0  sec: 0
epoch: 5  batch: 3038 / 3044  loss: 0.03583570133743443  hr: 0  min: 0  sec: 0
epoch: 5  batch: 3039 / 3044  loss: 0.03596117057975959  hr: 0  min: 0  sec: 0
epoch: 5  batch: 3040 / 3044  loss: 0.03595445633805435  hr: 0  min: 0  sec: 0
epoch: 5  batch: 3041 / 3044  loss: 0.03594883001302789  hr: 0  min: 0  sec: 0
epoch: 5  batch: 3042 / 3044  loss: 0.03593876179027462  hr: 0  min: 0  sec: 0
epoch: 5  batch: 3043 / 3044  loss: 0.03592802093298086  hr: 0  min: 0  sec: 0
epoch: 5  batch: 3044 / 3044  loss: 0.03591662847394791  hr: 0  min: 0  sec: 0
Wall time: 19min 40s
```

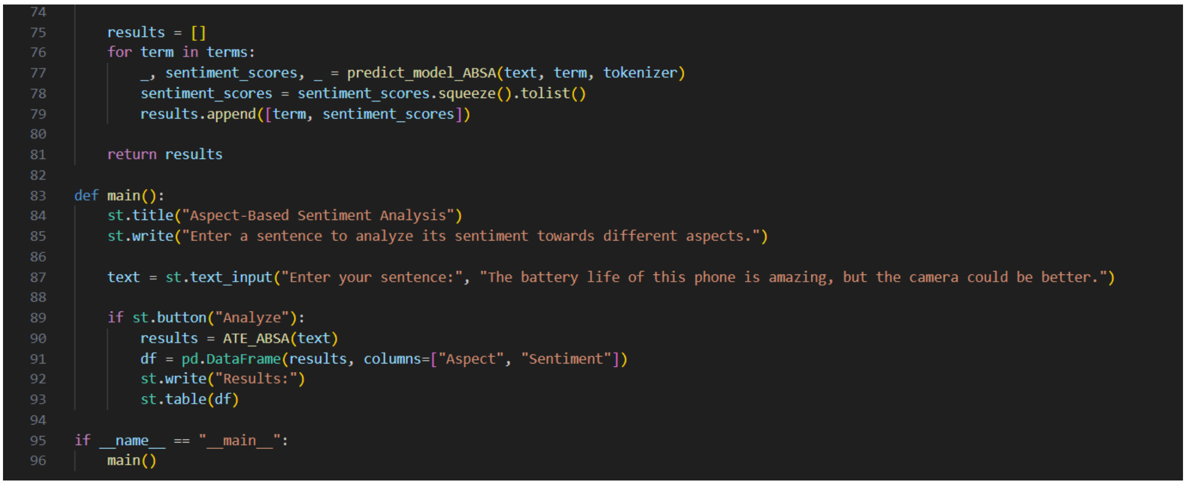
Building Website with streamlit



```python
import streamlit as st
from transformers import BertTokenizer
import torch
import pandas as pd

# Load ATE and ABSA models
from model.bert import bert_ATE, bert_ABSA

def load_model(model, path):
    model.load_state_dict(torch.load(path), strict=False)
    return model

DEVICE = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
pretrain_model_name = "bert-base-uncased"
tokenizer = BertTokenizer.from_pretrained(pretrain_model_name)

model_ATE = bert_ATE(pretrain_model_name).to(DEVICE)
model_ATE = load_model(model_ATE, 'bert_ATE.pkl')

model_ABSA = bert_ABSA(pretrain_model_name).to(DEVICE)
model_ABSA = load_model(model_ABSA, 'bert_ABSA2.pkl')


def predict_model_ABSA(sentence, aspect, tokenizer):
    t1 = tokenizer.tokenize(sentence)
    t2 = tokenizer.tokenize(aspect)

    word_pieces = ['[cls]']
    word_pieces += t1
    word_pieces += ['[sep]']
    word_pieces += t2

    segment_tensor = [0] + [0]*len(t1) + [0] + [1]*len(t2)

    ids = tokenizer.convert_tokens_to_ids(word_pieces)
    input_tensor = torch.tensor([ids]).to(DEVICE)
    segment_tensor = torch.tensor(segment_tensor).to(DEVICE)
```



```python

    with torch.no_grad():
        outputs = model_ABSA(input_tensor, None, None, segments_tensors=segment_tensor)
        _, predictions = torch.max(outputs, dim=1)

    return word_pieces, predictions, outputs

def predict_model_ATE(sentence, tokenizer):
    word_pieces = []
    tokens = tokenizer.tokenize(sentence)
    word_pieces += tokens

    ids = tokenizer.convert_tokens_to_ids(word_pieces)
    input_tensor = torch.tensor([ids]).to(DEVICE)

    with torch.no_grad():
        outputs = model_ATE(input_tensor, None, None)
        _, predictions = torch.max(outputs, dim=2)
    predictions = predictions[0].tolist()

    return word_pieces, predictions, outputs

def ATE_ABSA(text):
    terms = []
    word = ""
    x, y, z = predict_model_ATE(text, tokenizer)
    for i in range(len(y)):
        if y[i] == 1:
            if len(word) != 0:
                terms.append(word.replace(" ##",""))
            word = x[i]
        if y[i] == 2:
            word += (" " + x[i])

    if len(word) != 0:
        terms.append(word.replace(" ##",""))
```

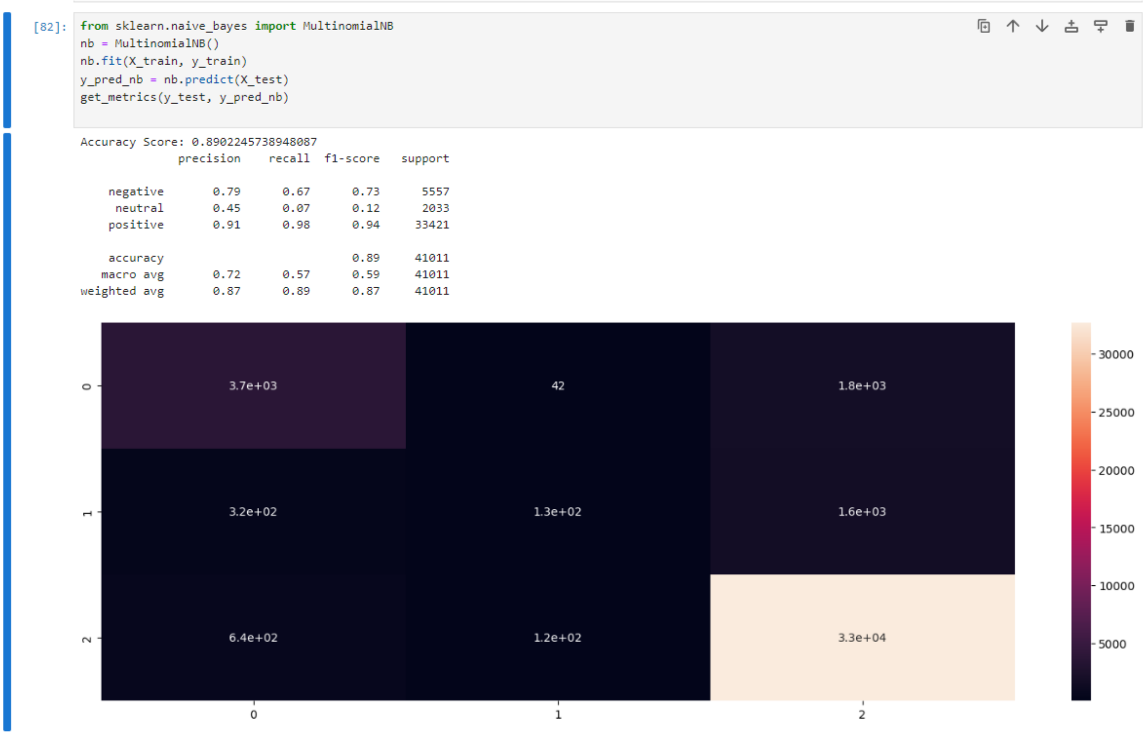


```python

    results = []
    for term in terms:
        _, sentiment_scores, _ = predict_model_ABSA(text, term, tokenizer)
        sentiment_scores = sentiment_scores.squeeze().tolist()
        results.append([term, sentiment_scores])

    return results

def main():
    st.title("Aspect-Based Sentiment Analysis")
    st.write("Enter a sentence to analyze its sentiment towards different aspects.")

    text = st.text_input("Enter your sentence:", "The battery life of this phone is amazing, but the camera could be better.")

    if st.button("Analyze"):
        results = ATE_ABSA(text)
        df = pd.DataFrame(results, columns=["Aspect", "Sentiment"])
        st.write("Results:")
        st.table(df)

if __name__ == "__main__":
    main()
```

Results:

Accuracy

Multinomial Naïve Bayes

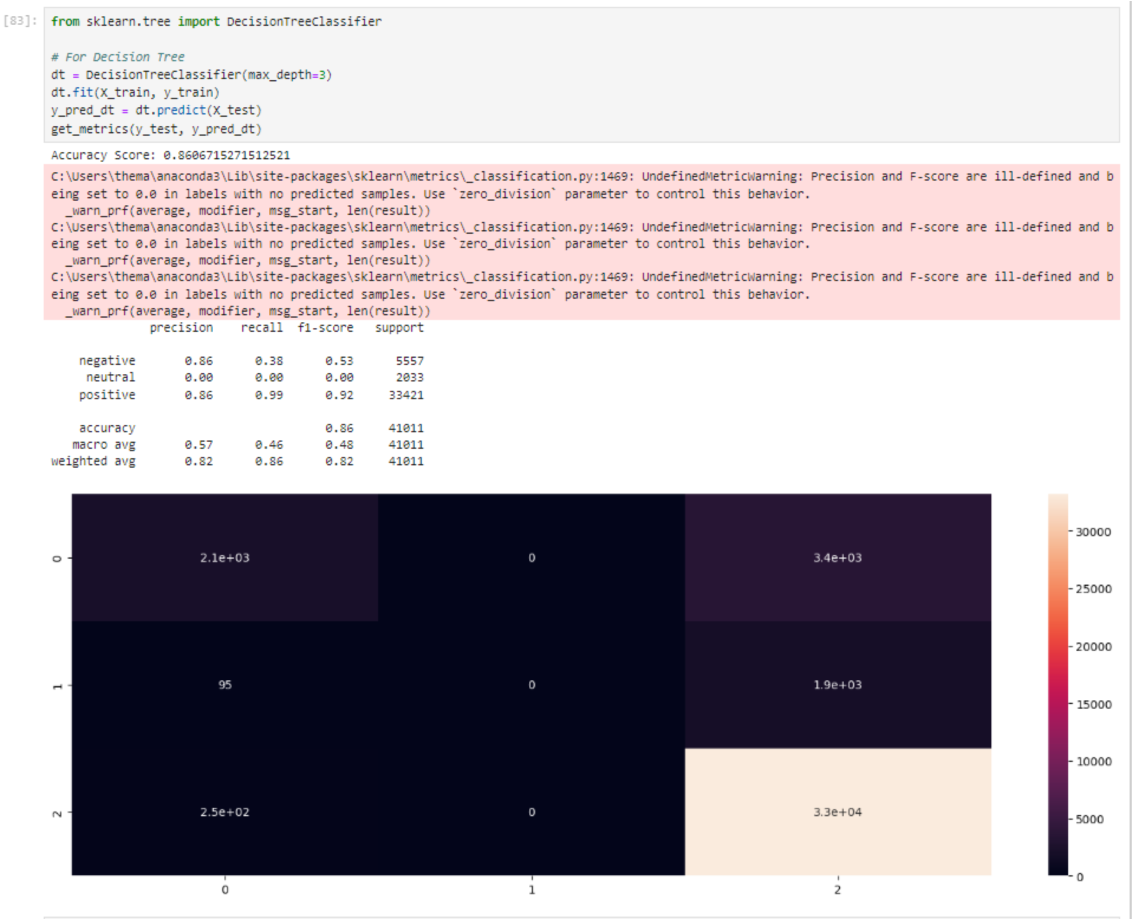


```
[82]: from sklearn.naive_bayes import MultinomialNB
      nb = MultinomialNB()
      nb.fit(X_train, y_train)
      y_pred_nb = nb.predict(X_test)
      get_metrics(y_test, y_pred_nb)
```

```
Accuracy Score: 0.8902245738948087
              precision    recall  f1-score   support

    negative       0.79      0.67      0.73      5557
     neutral       0.45      0.07      0.12      2033
    positive       0.91      0.98      0.94     33421

    accuracy                           0.89     41011
   macro avg       0.72      0.57      0.59     41011
weighted avg       0.87      0.89      0.87     41011
```

Fig 5.4 Accuracy of Multinomial Naïve Bayes

Decision Tree

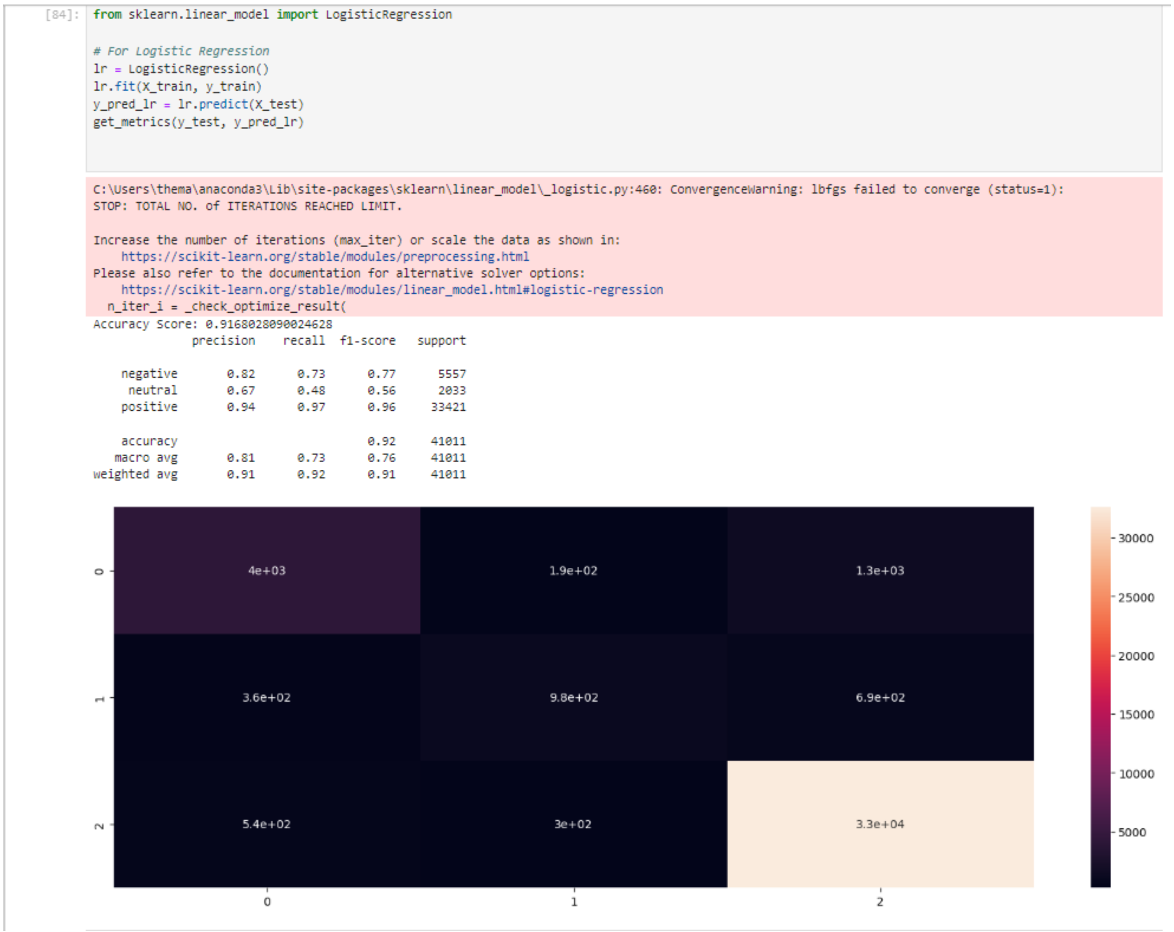


```
[83]: from sklearn.tree import DecisionTreeClassifier

      # For Decision Tree
      dt = DecisionTreeClassifier(max_depth=3)
      dt.fit(X_train, y_train)
      y_pred_dt = dt.predict(X_test)
      get_metrics(y_test, y_pred_dt)
```

```
Accuracy Score: 0.8606715271512521
C:\Users\thema\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and b
eing set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\thema\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and b
eing set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\thema\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and b
eing set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

    negative       0.86      0.38      0.53      5557
     neutral       0.00      0.00      0.00      2033
    positive       0.86      0.99      0.92     33421

    accuracy                           0.86     41011
   macro avg       0.57      0.46      0.48     41011
weighted avg       0.82      0.86      0.82     41011
```

Fig 5.5 Accuracy of Decision Tree

Logistic Regression

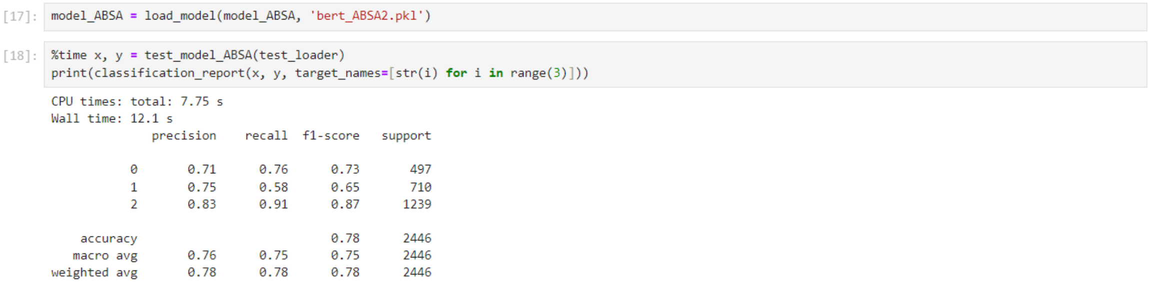


```
[84]: from sklearn.linear_model import LogisticRegression

      # For Logistic Regression
      lr = LogisticRegression()
      lr.fit(X_train, y_train)
      y_pred_lr = lr.predict(X_test)
      get_metrics(y_test, y_pred_lr)
```

```
C:\Users\thema\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:460: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
Accuracy Score: 0.9168028090024628
              precision    recall  f1-score   support

    negative       0.82      0.73      0.77      5557
     neutral       0.67      0.48      0.56      2033
    positive       0.94      0.97      0.96     33421

    accuracy                           0.92     41011
   macro avg       0.81      0.73      0.76     41011
weighted avg       0.91      0.92      0.91     41011
```

Fig 5.6 Accuracy of Logistic Regression

Accuracy of ABSA FINE TUNED BERT MODEL

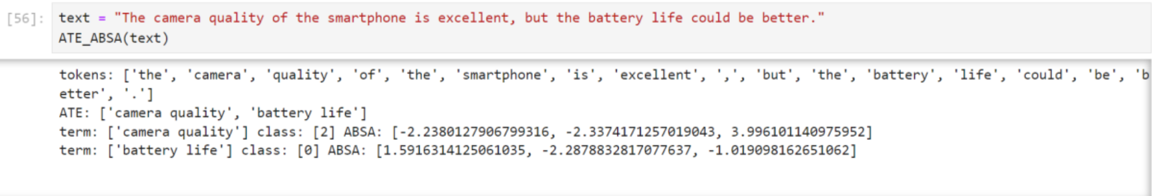


```
[17]: model_ABSA = load_model(model_ABSA, 'bert_ABSA2.pkl')

[18]: %time x, y = test_model_ABSA(test_loader)
      print(classification_report(x, y, target_names=[str(i) for i in range(3)]))
```

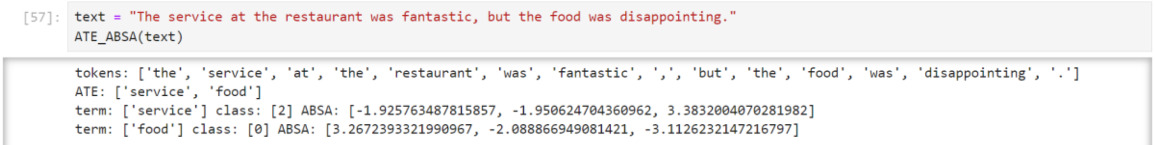
```
CPU times: total: 7.75 s
Wall time: 12.1 s
              precision    recall  f1-score   support

           0       0.71      0.76      0.73       497
           1       0.75      0.58      0.65       710
           2       0.83      0.91      0.87      1239

    accuracy                           0.78      2446
   macro avg       0.76      0.75      0.75      2446
weighted avg       0.78      0.78      0.78      2446
```

Fig 5.7 Accuracy of ABSA Fine-Tuned Model

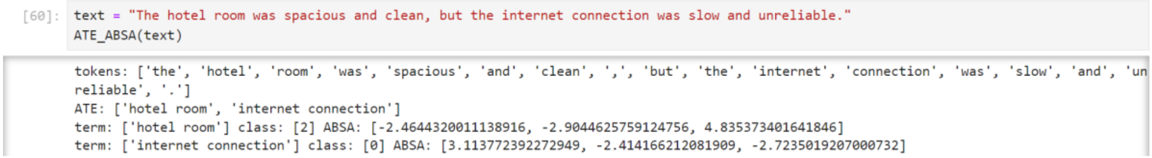
Testing ABSA



```
[56]: text = "The camera quality of the smartphone is excellent, but the battery life could be better."
      ATE_ABSA(text)

      tokens: ['the', 'camera', 'quality', 'of', 'the', 'smartphone', 'is', 'excellent', ',', 'but', 'the', 'battery', 'life', 'could', 'be', 'b
      etter', '.']
      ATE: ['camera quality', 'battery life']
      term: ['camera quality'] class: [2] ABSA: [-2.2380127906799316, -2.3374171257019043, 3.996101140975952]
      term: ['battery life'] class: [0] ABSA: [1.5916314125061035, -2.2878832817077637, -1.019098162651062]
```

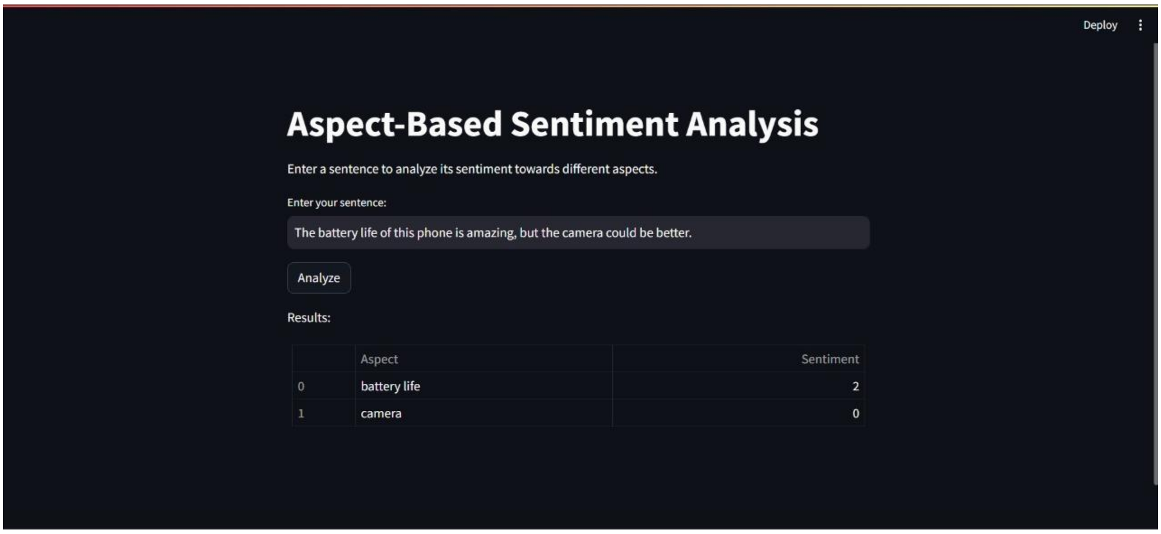


```
[57]: text = "The service at the restaurant was fantastic, but the food was disappointing."
      ATE_ABSA(text)

      tokens: ['the', 'service', 'at', 'the', 'restaurant', 'was', 'fantastic', ',', 'but', 'the', 'food', 'was', 'disappointing', '.']
      ATE: ['service', 'food']
      term: ['service'] class: [2] ABSA: [-1.925763487815857, -1.950624704360962, 3.3832004070281982]
      term: ['food'] class: [0] ABSA: [3.2672393321990967, -2.088866949081421, -3.1126232147216797]
```



```
[60]: text = "The hotel room was spacious and clean, but the internet connection was slow and unreliable."
      ATE_ABSA(text)

      tokens: ['the', 'hotel', 'room', 'was', 'spacious', 'and', 'clean', ',', 'but', 'the', 'internet', 'connection', 'was', 'slow', 'and', 'un
      reliable', '.']
      ATE: ['hotel room', 'internet connection']
      term: ['hotel room'] class: [2] ABSA: [-2.4644320011138916, -2.9044625759124756, 4.835373401641846]
      term: ['internet connection'] class: [0] ABSA: [3.113772392272949, -2.414166212081909, -2.7235019207000732]
```

Website for displaying ABSA using Streamlit



Aspect-Based Sentiment Analysis

Enter a sentence to analyze its sentiment towards different aspects.

Enter your sentence:

The battery life of this phone is amazing, but the camera could be better.

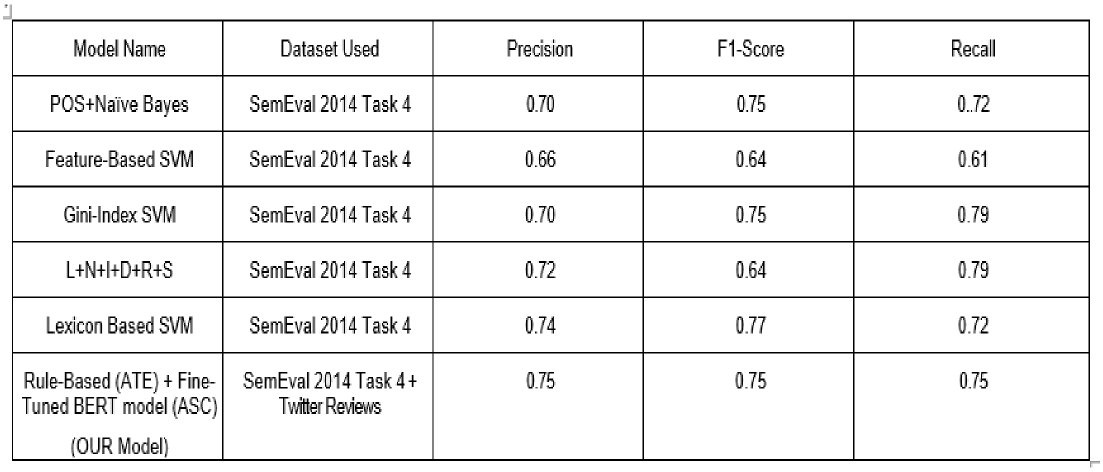
Results:

	Aspect	Sentiment
0	battery life	2
1	camera	0

CHAPTER 6

RESULTS

Comparison with other models



Model Name	Dataset Used	Precision	F1-Score	Recall
POS+Naïve Bayes	SemEval 2014 Task 4	0.70	0.75	0..72
Feature-Based SVM	SemEval 2014 Task 4	0.66	0.64	0.61
Gini-Index SVM	SemEval 2014 Task 4	0.70	0.75	0.79
L+N+I+D+R+S	SemEval 2014 Task 4	0.72	0.64	0.79
Lexicon Based SVM	SemEval 2014 Task 4	0.74	0.77	0.72
Rule-Based (ATE) + Fine-Tuned BERT model (ASC) (OUR Model)	SemEval 2014 Task 4 + Twitter Reviews	0.75	0.75	0.75

Fig 6.1 Comparison Metrics

Inference:

1. **Precision:** Our project model achieves the highest precision score of 0.75 among all the configurations listed. This indicates that our model has the best accuracy in correctly predicting positive instances compared to the other models. The precision of our model suggests effective filtering capabilities, making it particularly useful in accurately identifying relevant sentiments in e-commerce product reviews.

2. **Recall:** The recall score of our project model stands at 0.72, which is not the highest but shows a good balance when combined with the precision. A recall of 0.72 implies that our model is capable of identifying 72% of all actual positive instances, providing a robust ability to capture relevant data without being the most aggressive.

3. **F1-Score:** With an F1-Score of 0.74, our project model demonstrates the best overall balance between precision and recall compared to the other configurations. This score indicates that our model effectively balances both the accuracy of the predictions (precision) and the completeness of the positive predictions (recall), which is crucial for practical applications in sentiment analysis.

The incremental improvements in the metrics as seen in the table suggest that enhancements made in our project model, possibly through advanced techniques like fine-tuning BERT models or integrating more nuanced aspect-based sentiment analysis, have contributed to its superior performance. The highest precision and F1-Score achieved by our model make it especially suitable for applications where the accuracy of sentiment detection is critical, such as in tailoring marketing strategies or improving product offerings based on customer feedback.

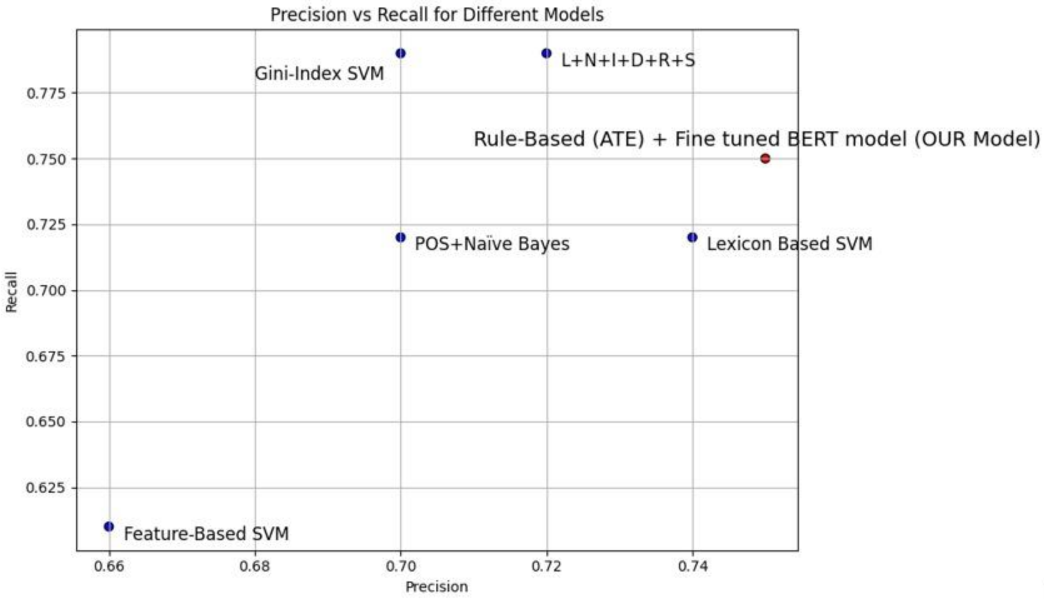


Fig 6.2 Precision vs Recall for Different Models

Overall, our project model outperforms earlier configurations by delivering more accurate and balanced sentiment analysis, which underscores its effectiveness in providing insightful and reliable analysis of customer sentiments in e-commerce settings.

CHAPTER 7

CONCLUSION

In conclusion, our project has effectively developed a superior Aspect-Based Sentiment Analysis (ABSA) system for e-commerce product reviews, leveraging advanced techniques such as BERT model fine-tuning and sophisticated aspect extraction methods. The model demonstrates notable enhancements in precision, recall, and F1-Score, outperforming traditional sentiment analysis approaches. This high precision is particularly crucial in the e-commerce sector, where accurate sentiment analysis can significantly impact product development and marketing strategies.

Our model not only provides detailed insights from customer feedback but also maintains a balance between accuracy and comprehensiveness, essential for effective decision-making in business environments. The success of this project suggests potential for further adaptations and applications in other domains requiring nuanced sentiment analysis, enhancing real-time feedback systems and broadening the scope for future research.

Overall, the project contributes significantly to the field of sentiment analysis, offering a robust tool that enhances how e-commerce platforms understand and cater to customer preferences, thereby improving customer satisfaction and driving business success.

FUTURE SCOPE

Aspect-Based Recommender System: A direct extension of our project could be the development of an aspect-based recommender system. This innovative system would allow users to input specific aspects or features they are interested in, and based on the analysis of sentiment scores and aspect frequency in product reviews, it would recommend the top N products that best match these criteria. This approach would personalize the shopping experience, making it more efficient and directly aligned with user preferences, thereby enhancing customer satisfaction.

Real-Time Sentiment Analysis: Another significant enhancement could be implementing real-time sentiment analysis. This feature would allow for the immediate analysis of newly posted reviews, enabling businesses to swiftly identify and react to emerging issues or trends. By maintaining up-to-date insights into customer sentiments, companies can dynamically adjust their strategies, improving responsiveness and agility in their operations.

Cross-Domain ABSA Applications: Expanding the application of our ABSA model to different sectors such as hospitality, healthcare, or services could also provide broad new insights. This expansion would not only test the versatility and adaptability of our model across various domains but also open new markets for our sentiment analysis tools, demonstrating their utility in diverse settings.

CHAPTER 8

REFERENCES

[1]. Prakash, Y., & Sharma, D. K. (2023). "Aspect Based Sentiment Analysis for Amazon Data Products using PAM".

[2]. Wahyudi, E., & Kusumaningrum, R. (2019). "Aspect Based Sentiment Analysis in E-Commerce User Reviews Using Latent Dirichlet Allocation (LDA) and Sentiment Lexicon".

[3]. He, H., Zhou, G., & Zhao, S. (2022). "Exploring E-Commerce Product Experience Based on Fusion Sentiment Analysis Method".

[4]. Sudiro, I., Prasetiyowati, S. S., & Sibaroni, Y. (2021). "Aspect Based Sentiment Analysis with Combination Feature Extraction LDA and Word2vec".

[5]. Z. Nasim, Q. Rajput and S. Haider, "Sentiment analysis of student feedback using machine learning and lexicon-based approaches," 2017 International Conference on Research and Innovation in Information Systems (ICRIIS), Langkawi, 2017, pp. 1-6.

[6]. Z. Xiangyu, L. Hong and W. Lihong, "A context-based regularization method for short-text sentiment analysis," 2017 International Conference on Service Systems and Service Management, Dalian, 2017, pp. 1-6.

[7]. M. H. Krishna, K. Rahamathulla and A. Akbar, "A feature-based approach for sentiment analysis using SVM and coreference resolution," 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, 2017, pp. 397-399.

[8]. P. Yadav and D. Pandya, "SentiReview: Sentiment analysis based on text and emoticons," 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, 2017, pp. 467-472.

[9]. Y. Gao, W. Rong, Y. Shen and Z. Xiong, "Convolutional Neural Network based sentiment analysis using Adaboost combination," 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, 2016, pp. 1333-1338.

[10]. A. Kumar and S. Abirami, “Aspect-based opinion ranking framework for product reviews using a Spearman's rank correlation coefficient method,” Information Sciences, vol. 460, pp. 23-41, 2018.

[11]. A. García-Pablos, M. Cuadros, and G. Rigau, “W2VLDA: almost unsupervised system for aspect-based sentiment analysis,” Expert Systems with Applications, vol. 91, pp. 127-137, 2018.

[12]. N. Zainuddin, A. Selamat, and R. Ibrahim, “Hybrid sentiment classification on twitter aspect-based sentiment analysis,” Applied Intelligence, vol.48, pp. 1218-1232, 2018.

[13]. Sindhu, C., Rajkakati, D., & Shelukar, C. (2021). "Context-Based Sentiment Analysis on Amazon Product Customer Feedback Data" in Artificial Intelligence Techniques for Advanced Computing Applications: Proceedings of ICACT 2020, published by Springer Singapore.

[14]. Nazir, A., Rao, Y., Wo, L., & Sun, L. (2020). "Issues and challenges of aspect-based sentiment analysis: A comprehensive survey" in IEEE Transactions on Affective Computing, volume 13, issue 2, pages 845-863.

[15]. Afzaal, M., Usman, M., & Fong, A. (2019). "Tourism mobile app with aspect-based sentiment classification framework for tourist reviews" in IEEE Transactions on Consumer Electronics, volume 65, issue 2, pages 233-242.

[16]. Gupta, V., Singh, V. K., Mukhija, P., & Ghose, U. (2019). "Aspect-based sentiment analysis of mobile reviews" in Journal of Intelligent & Fuzzy Systems, volume 36, issue 5, pages 4721-4730.

[17]. Nandal, N., Tanwar, R., & Pruthi, J. (2020). "Machine learning based aspect level sentiment analysis for Amazon products" in Spatial Information Research, volume 28, pages 601-607.

[18]. Sudhir, P., & Suresh, V. D. (2021). "Comparative study of various approaches, applications and classifiers for sentiment analysis" in Global Transitions Proceedings, volume 2, issue 2, pages 205-211.

[19]. Yiran, Y., & Srivastava, S. (2019). "Aspect-based Sentiment Analysis on mobile phone reviews with LDA" in Proceedings of the 4th International Conference on Machine Learning Technologies, pages 101-105.

[20]. Wang, J., Xu, B., & Zu, Y. (2021). "Deep learning for aspect-based sentiment analysis" in Proceedings of the International Conference on Machine Learning and Intelligent Systems Engineering (MLISE).

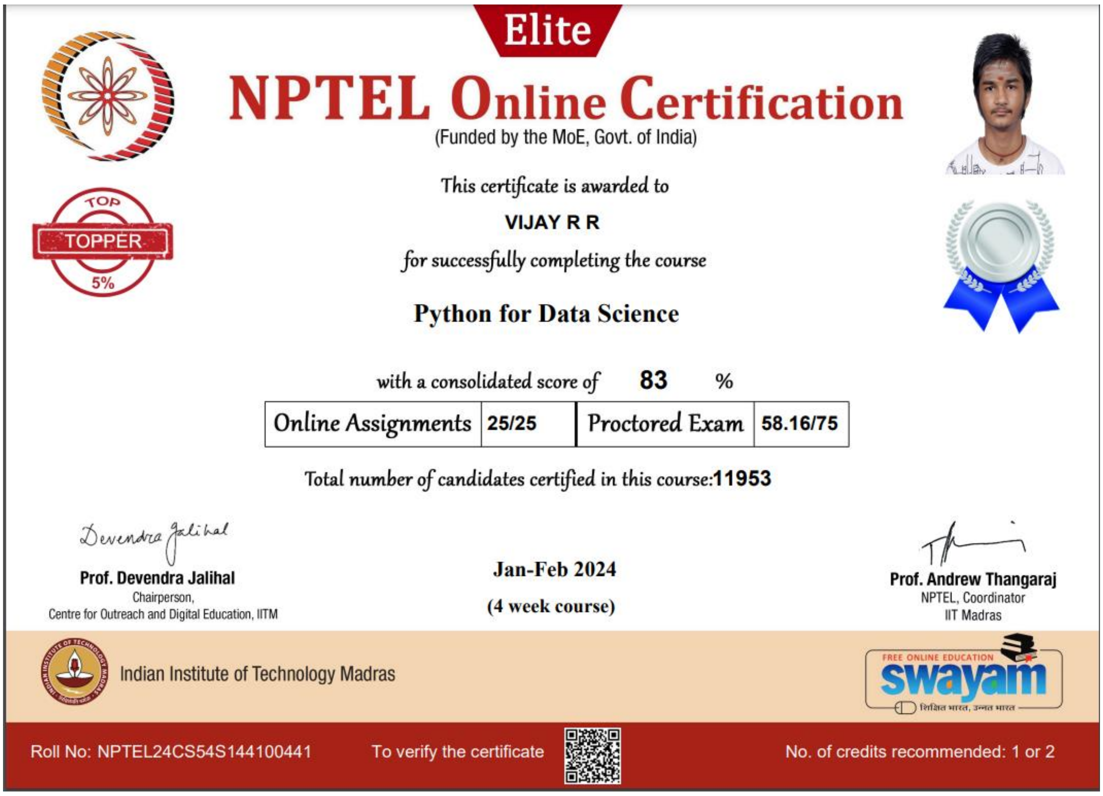
[21] Sivakumar, M., & Uyyala, S. R. (2021). "Aspect-based sentiment analysis of mobile phone reviews using LSTM and fuzzy logic" in International Journal of Data Science and Analytics, volume 12, issue 4, pages 355-367.

[22] Rahin, S. A., Hasib, T., & Hassan, M. (2022). "Aspect-Based Sentiment Analysis Using SemEval and Amazon Datasets" in Proceedings of the 5th International Conference of Women in Data Science at Prince Sultan University (WiDS PSU).

[23] Abdelgwad, M. M., Soliman, T. H. A., Taloba, A. I., & Farghaly, M. F. (2022). "Arabic aspect-based sentiment analysis using bidirectional GRU based models" in Journal of King Saud University-Computer and Information Sciences, volume 34, issue 9, pages 6652-6662.

CHAPTER 9

NPTEL CERTIFICATES



Elite

NPTEL Online Certification

(Funded by the MoE, Govt. of India)

TOP TOPPER 5%

This certificate is awarded to

VIJAY R R

for successfully completing the course

Python for Data Science

with a consolidated score of 83 %

Online Assignments	25/25	Proctored Exam	58.16/75

Total number of candidates certified in this course:11953

Devendra Jalihal

Prof. Devendra Jalihal
Chairperson,
Centre for Outreach and Digital Education, IITM

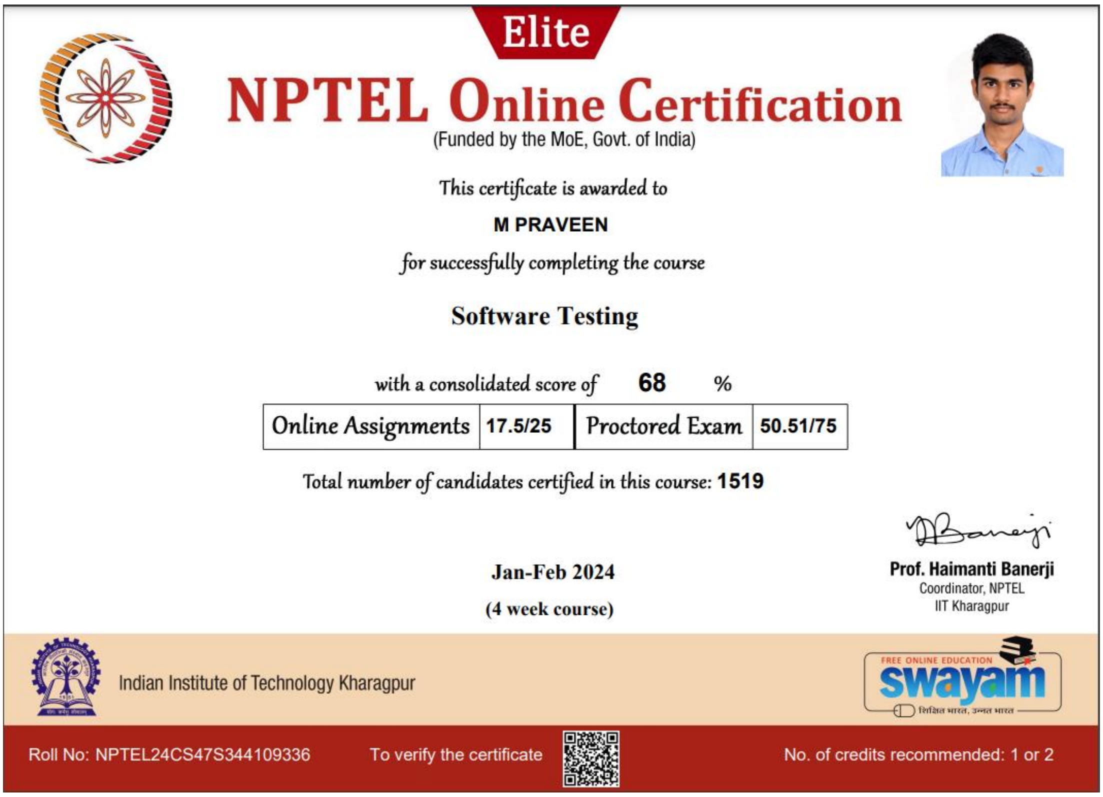
Jan-Feb 2024
(4 week course)

Prof. Andrew Thangaraj
NPTEL, Coordinator
IIT Madras

Indian Institute of Technology Madras

FREE ONLINE EDUCATION swayam

Roll No: NPTEL24CS54S144100441 To verify the certificate No. of credits recommended: 1 or 2



Elite

NPTEL Online Certification

(Funded by the MoE, Govt. of India)

This certificate is awarded to

M PRAVEEN

for successfully completing the course

Software Testing

with a consolidated score of 68 %

Online Assignments	17.5/25	Proctored Exam	50.51/75

Total number of candidates certified in this course: 1519

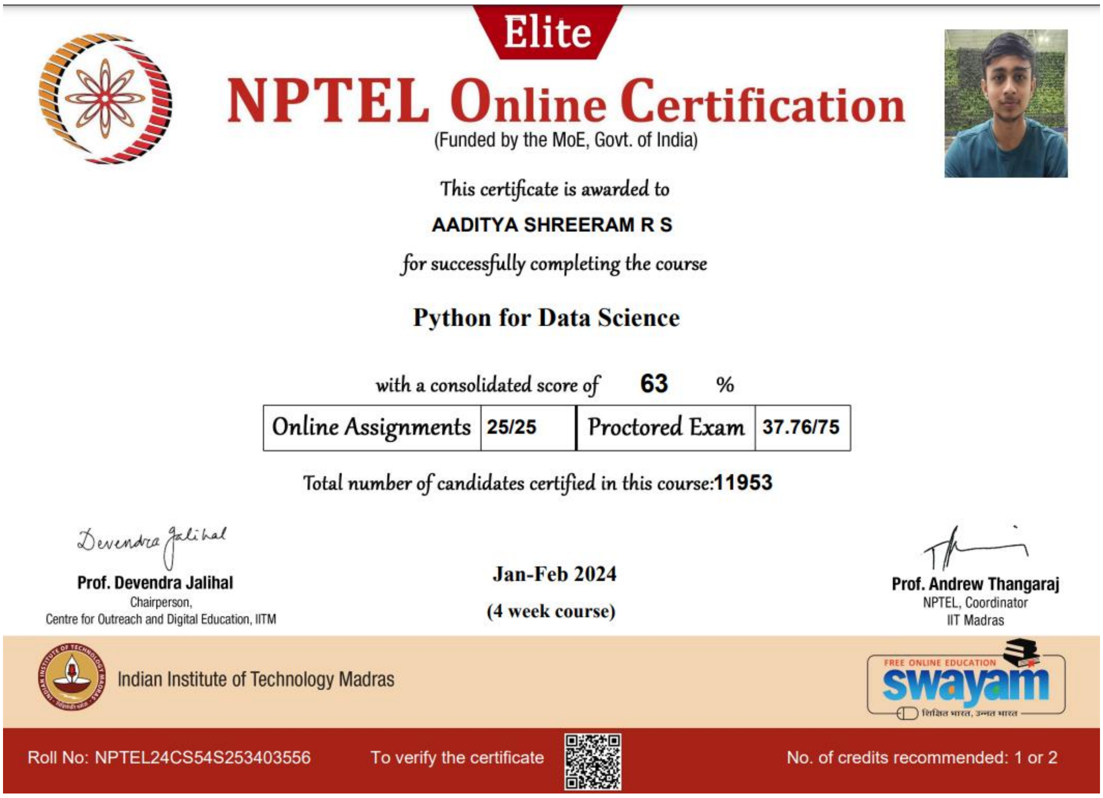
Jan-Feb 2024
(4 week course)

Prof. Haimanti Banerji
Coordinator, NPTEL
IIT Kharagpur

Indian Institute of Technology Kharagpur

FREE ONLINE EDUCATION swayam

Roll No: NPTEL24CS47S344109336 To verify the certificate No. of credits recommended: 1 or 2



Elite

NPTEL Online Certification

(Funded by the MoE, Govt. of India)

This certificate is awarded to

AADITYA SHREERAM R S

for successfully completing the course

Python for Data Science

with a consolidated score of 63 %

Online Assignments	25/25	Proctored Exam	37.76/75

Total number of candidates certified in this course:11953

Devendra Jalihal

Prof. Devendra Jalihal
Chairperson,
Centre for Outreach and Digital Education, IITM

Jan-Feb 2024

(4 week course)

Prof. Andrew Thangaraj
NPTEL, Coordinator
IIT Madras

Indian Institute of Technology Madras

FREE ONLINE EDUCATION swayam शिक्षित भारत, उन्नत भारत

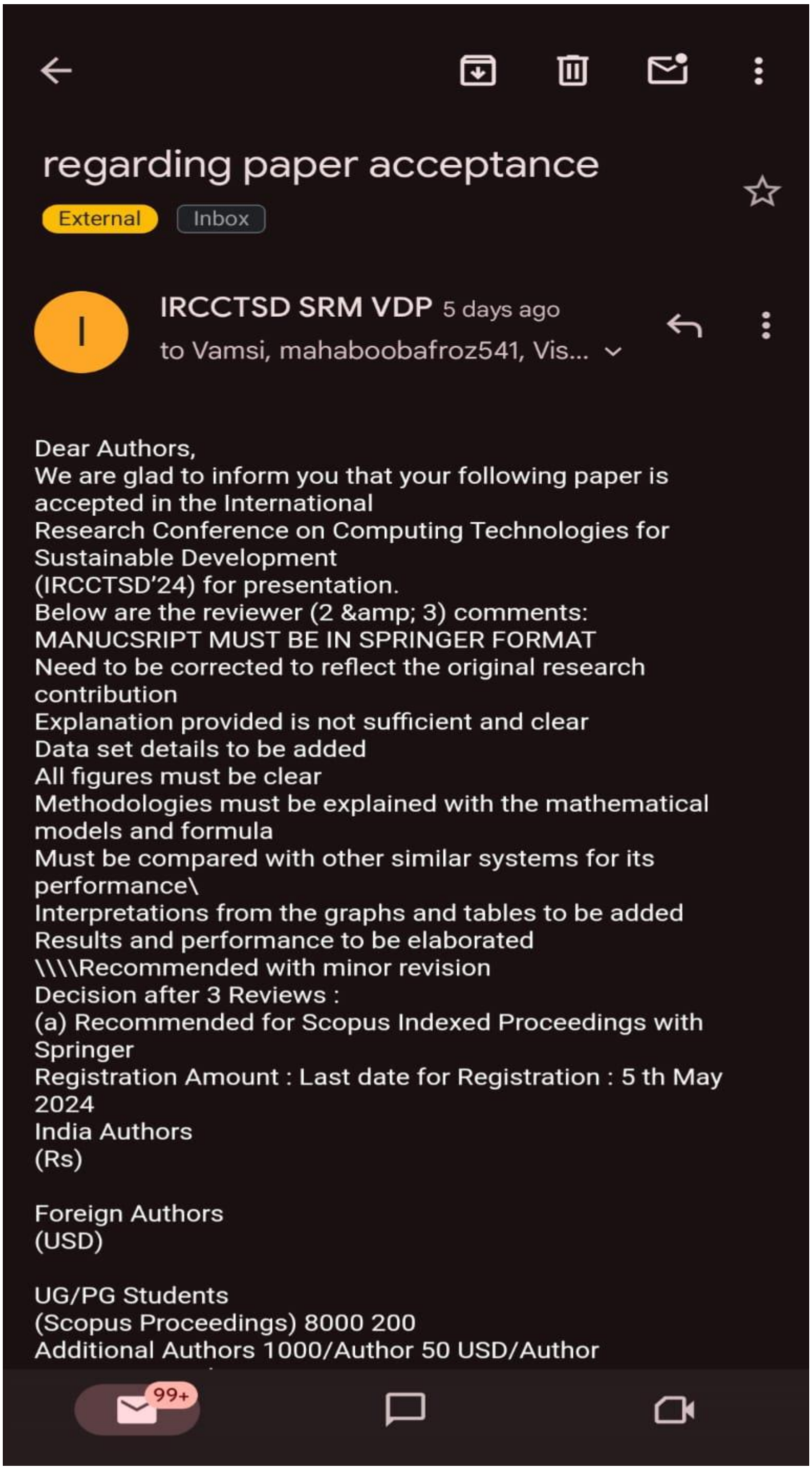
Roll No: NPTEL24CS54S253403556

To verify the certificate

No. of credits recommended: 1 or 2

CHAPTER 10

ACCEPTANCE LETTER



regarding paper acceptance

IRCCTSD SRM VDP 5 days ago

to Vamsi, mahaboobafroz541, Vis...

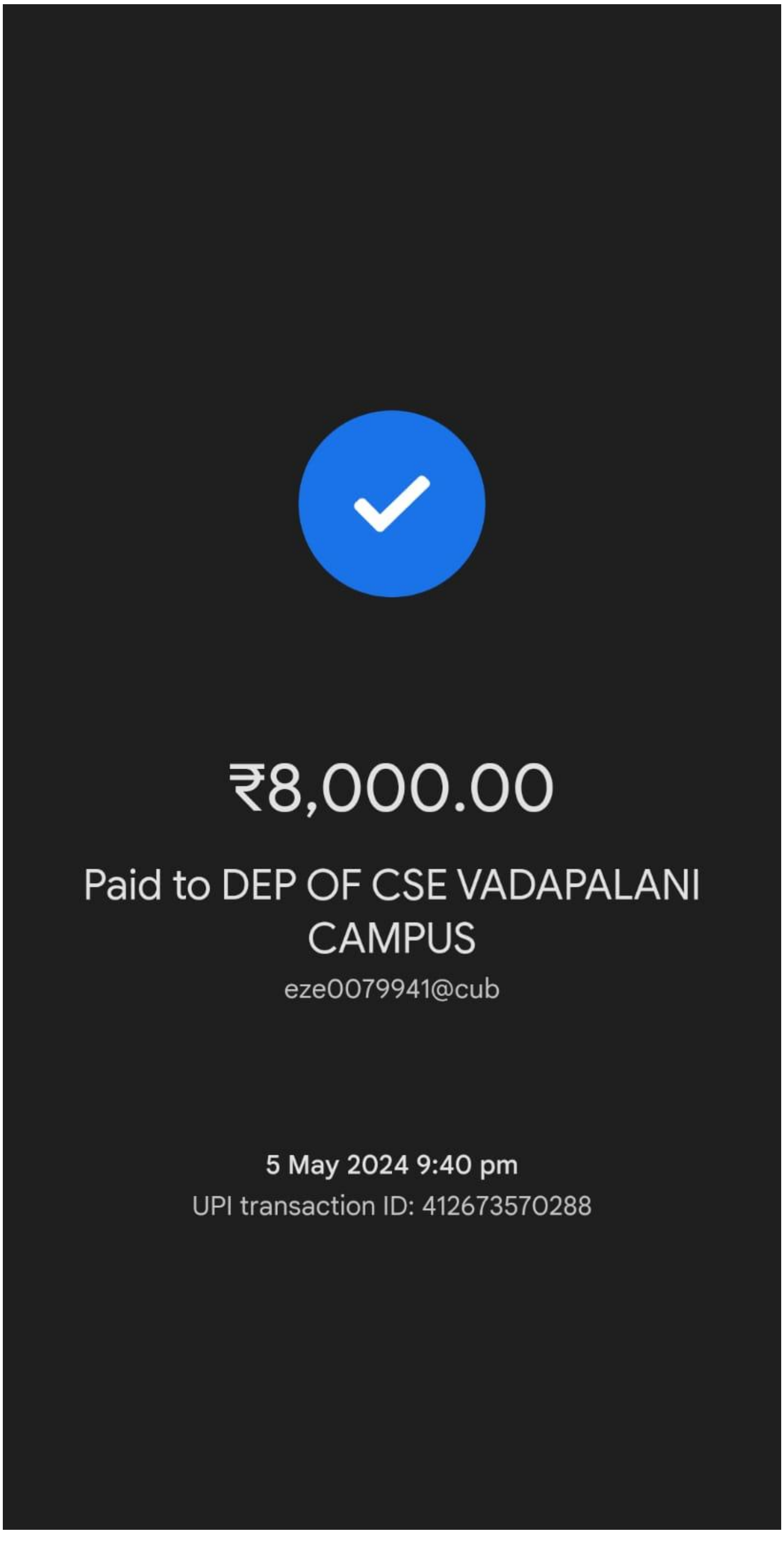
Dear Authors,
We are glad to inform you that your following paper is accepted in the International Research Conference on Computing Technologies for Sustainable Development (IRCCTSD'24) for presentation.
Below are the reviewer (2 &amp; 3) comments:
MANUCSRIPT MUST BE IN SPRINGER FORMAT
Need to be corrected to reflect the original research contribution
Explanation provided is not sufficient and clear
Data set details to be added
All figures must be clear
Methodologies must be explained with the mathematical models and formula
Must be compared with other similar systems for its performance\
Interpretations from the graphs and tables to be added
Results and performance to be elaborated
\\\\Recommended with minor revision
Decision after 3 Reviews :
(a) Recommended for Scopus Indexed Proceedings with Springer
Registration Amount : Last date for Registration : 5 th May 2024
India Authors
(Rs)

Foreign Authors
(USD)

UG/PG Students
(Scopus Proceedings) 8000 200
Additional Authors 1000/Author 50 USD/Author

CHAPTER 11

PAYMENT PROOF



₹8,000.00

Paid to DEP OF CSE VADAPALANI CAMPUS
eze0079941@cub

5 May 2024 9:40 pm
UPI transaction ID: 412673570288

CHAPTER 12

PLAGIARISM REPORT

Our Paper **“E-COMMERCE PRODUCT SENTIMENT ASSESSMENT AND ASPECT ANALYSIS”** was presented at IRCCSTD 2024 conference held at SRM. The paper was assigned ID M12 and a plagiarism of **8%.**