

CSCE 5290: NATURAL LANGUAGE PROCESSING

PROJECT REPORT

GROUP – 10

Deeksha Thandra - 11545852

Avinash Chinta – 11523130

Sushmitha Dharmareddy – 11549862

Mavya Tekula – 11511499

Project Title:

Text Summarization and Sentiment Analysis on Amazon Reviews.

Introduction:

Due to the popularity of online marketplaces over the past few decades, online vendors and merchants now request feedback from their customers on the goods they have purchased. As a result, millions of reviews are produced every day, which makes it challenging for a customer to decide whether to purchase the goods or not. For product manufacturers, it is challenging and time-consuming to analyze this massive volume of comments. Our project helps in examining the issue of categorizing reviews according to their overall semantic (positive or negative). To conduct the study two different area are integrated i.e text summarization and sentiment analysis along with supervised machine learning techniques, SVM and logistic regression, has been attempted on product reviews from Amazon. Text summarization and sentiment classification both aim to capture the main ideas of the text but at different levels. Text summarization is to describe the text within a few sentences, while sentiment classification can be regarded as a special type of summarization which “summarizes” the text into an even more abstract fashion, i.e., a sentiment class. Their accuracies have then been compared. The results showed that the SVM approach outperforms the logistic regression approach when the data set is bigger. Both algorithms, however, achieve accuracies of at least 80%, which is encouraging. Finally sentiment analysis is implemented using TextBlob, VADER, Bag of Words Vectorizer, and Transformer based models.

Motivation:

Product reviews are becoming increasingly significant, with the evolution of many retail stores to online shopping. Consumers are posting reviews directly on product pages in real time. The purpose of this project is to investigate how companies can conduct sentiment analysis based on their reviews to gain more insights into customer experiences. Around 100 amazon review files will be analyzed using sentiment analysis along with NLP techniques to understand customer experiences using Amazon reviews and to get the feedback about the products.

There are challenges in the world related to data and organizations are facing issues when analyzing the reviews, they are receiving from the users. These challenges not only arise from the data, but also involve users basically information seekers. These challenges include huge data, users need for different information, informal/ unstructured text. To overcome these challenges, we came up with some of the NLP techniques like text summarization and sentiment analysis.

Significance:

Today's social media platforms and online networks enable businesses to gather honest feedback from clients all around the world. Customers' experiences with prices, value, quality, customer service, ease of shopping, and other aspects of their online purchases are revealed in customer reviews. Since the customer reviews are unstructured, sentiment analysis will make it easier and more affordable to make sense of them.

Most of the current automatic text summarizing systems create a summary using an extraction strategy. To create extraction summaries, sentence extraction algorithms are frequently utilized. Sentence scoring, which assigns a numerical value to each sentence for the summary, is one technique for finding appropriate sentences.

Objectives:

We are extracting the data from all the 100 amazon review files and then we'll implement the tokenization to split the text into individual tokens and then we are removing the stopwords using NLTK libraries.

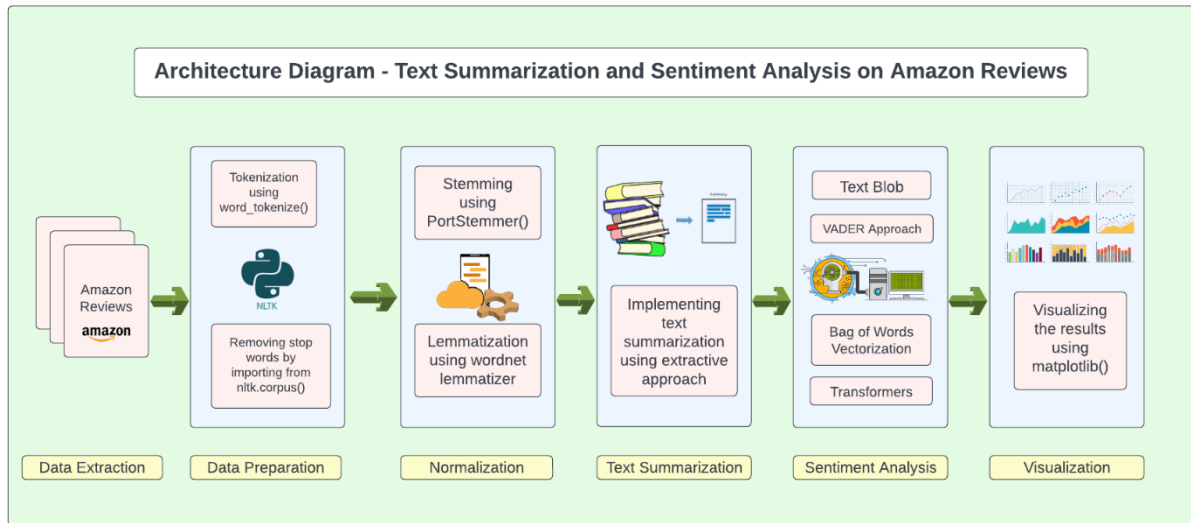
We implemented the normalization techniques like stemming and lemmatization which helps us to extract the root words from the generated tokens. Furthermore, we made use of text summarization and sentiment analysis on our review files there after compared and visualized the results. For most real-life applications, categorizing the sentiment or subjectivity expressed in documents is not sufficient. Practical applications require a precise investigation. Due to the maturity of the field, the problem is better defined and finding the different research directions.

Background/Related Work:

Due to the constantly increasing volume of reviews, it has recently been an active research field in NLP. It has become difficult for people or organizations to effectively process the volume of information included in the corpus, because most people choose to express their opinions on specific items, services, or organizations through online blogging or social networking sites[2]. Sentiment analysis has drawn a lot of attention lately because there are so many online reviews. As a result, this topic has been the subject of numerous studies.

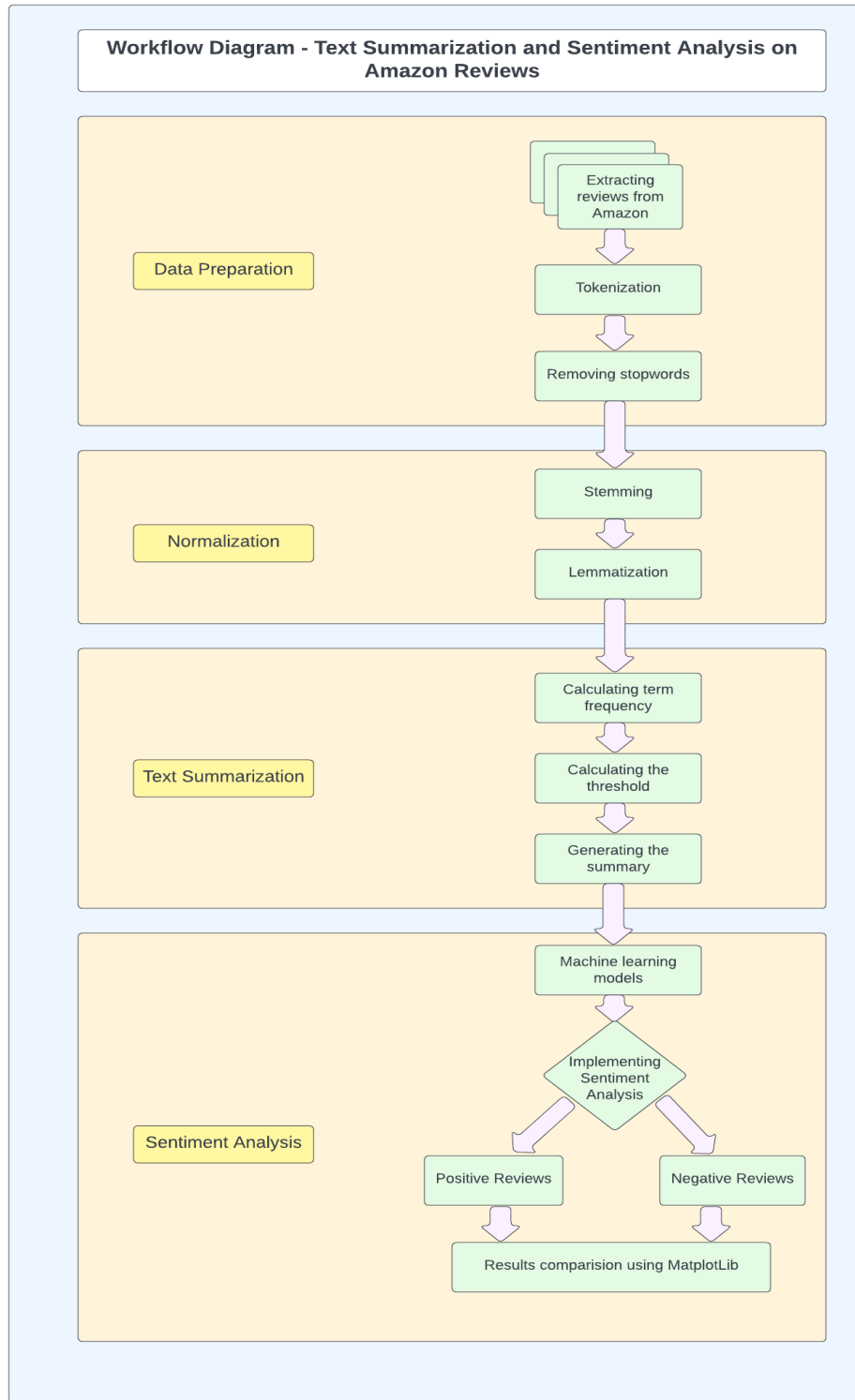
A major research field has emerged around the subject of how to extract the best and most accurate method and simultaneously categorize the customers reviews into positive or negative opinions[4]. Pang, Lee, and Vaidyanathan[13] were the first to propose sentiment categorization using machine learning models on a dataset of movie reviews[15]. On data made up of unigrams and bigrams, they examined the Naive Bayes, Max Entropy, and Support Vector Machine models for sentiment analysis. SVM combined with unigram feature extraction yielded the best outcomes in their trial. They noted an outcome of 82.9% accuracy. Using Facebook and Twitter posts, Hussain et al.[14] carried out an observational study to assess public opinion towards COVID-19 vaccinations in the US and UK. They employed a deep learning model for sentiment analysis and lexicon-based analysis. A weighted average of VADER and TextBlob was merged with the results of the bidirectional encoder representations from transformers (BERT) model in their study.

Architecture Diagram:



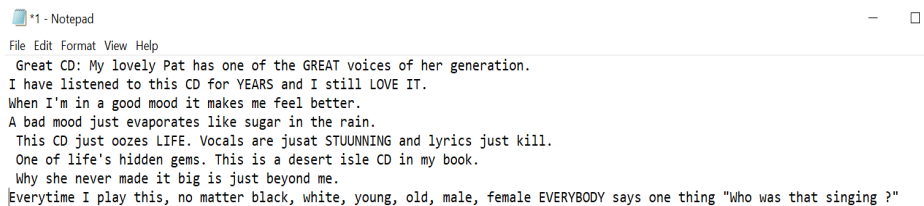
Architecture diagram shows the step-by-step implementation of our project. There are series of steps which are followed, and they are as follows: First we are extracting the amazon review files and then encoded them separately as text and label for further analysis using machine learning models. Next, we preprocessed the data using tokenization, stemming, lemmatization, removal of stop words and punctuation from the review files. Then the processed data is passes through the text summarization using extractive approach where we are extracting the summarized content for the respective review files. After that we analyzed the data before and after text summarization. We the performed sentiment analysis using four different approaches TextBlob, VADER, bag of words vectorizer and transformer-based models. TextBlob and VADER are in-built python libraries easy to implement and predict the sentiment from the given review files. Whereas for bag of words vectorizers, we need to preprocess the data and then pass the preprocessed data through CountVectorizer to generate the bag of words for the respective review files. Then we predicted the accuracy of the model. Transformer based models are most advanced techniques, where we need to generate the pipeline and pass the desired function through the pipeline to predict the label (either positive or negative) along with the sentiment score.

Workflow Diagram:



Dataset:

We are using the 100 amazon review files, each describing about single product. Our dataset contains the metadata like clothing, electronics or any other product reviews from Amazon. We will further preprocess these text files and predict the accurate number of positive and negative words. We are using the dataset where we encoded each review file with their respective labels i.e. either positive or negative. We used this dataset for implementing the machine learning models and predicting their accuracy.



*1 - Notepad

File Edit Format View Help

Great CD: My lovely Pat has one of the GREAT voices of her generation. I have listened to this CD for YEARS and I still LOVE IT. When I'm in a good mood it makes me feel better. A bad mood just evaporates like sugar in the rain. This CD just oozes LIFE. Vocals are just STUNNING and lyrics just kill. One of life's hidden gems. This is a desert isle CD in my book. Why she never made it big is just beyond me. Everytime I play this, no matter black, white, young, old, male, female EVERYBODY says one thing "Who was that singing ?"

snapshot of 1.txt file.

	label	review
1		
2	pos	Stuning even for the non-gamer: This sound track was beautiful! It paints the senery in your mind so well I would reconmend it even to people who hate vid. game music! I have played the game Chrono Cross but out of all of the games I have
3	pos	The best soundtrack ever to anything.: I'm reading a lot of reviews saying that this is the best 'game soundtrack' and I figured that I'd write a review to disagree a bit. This in my opinino is Yasunori Mitsuda's ultimate masterpiece. The music i
4	pos	Amazing!: This soundtrack is my favorite music of all time, hands down. The intense sadness of "Prisoners of Fate" (which means all the more if you've played the game) and the hope in "A Distant Promise" and "Girl who Stole the Star" hav
5	pos	Excellent Soundtrack: I truly like this soundtrack and I enjoy video game music. I have played this game and most of the music on here I enjoy and it's truly relaxing and peaceful.On disk one. my favorites are Scars Of Time, Between Life an
6	pos	Remember, Pull Your Jaw Off The Floor After Hearing it: If you've played the game, you know how divine the music is! Every single song tells a story of the game, it's that good! The greatest songs are without a doubt, Chrono Cross: Time's t
7	pos	an absolute masterpiece: I am quite sure any of you actually taking the time to read this have played the game at least once, and heard at least a few of the tracks here. And whether you were aware of it or not, Mitsuda's music contributed g
8	neg	Buyer beware: This is a self published book, and if you want to know why--read a few paragraphs! Those 5 star reviews must have been written by Ms. Haddon's family and friends--or perhaps, by herself! I can't imagine anyone reading the
9	pos	Glorious story: I loved Whisper of the wicked saints. The story was amazing and I was pleasantly surprised at the changes in the book. I am not normally someone who is into romance novels, but the world was raving about this book and so
10	pos	A FIVE STAR BOOK: I just finished reading Whisper of the Wicked saints. I fell in love with the characters. I expected an average romance read, but instead I found one of my favorite books of all time. Just when I thought I could predict the ox
11	pos	Whispers of the Wicked Saints: This was a easy to read book that made me want to keep reading on and on, not easy to put down.It left me wanting to read the follow on, which I hope is coming soon. I used to read a lot but have gotten aw
12	neg	The Worst!: A complete waste of time. Typographical errors, poor grammar, and a totally pathetic plot add up to absolutely nothing. I'm embarrassed for this author and very disappointed I actually paid for this book.
13	pos	Great book: This was a great book! I just could not put it down, and could not read it fast enough. Boy what a book the twist and turns in this just keeps you guessing and wanting to know what is going to happen next. This book makes you f
14	pos	Great Read: I thought this book was brilliant, but yet realistic. It showed me that to error is human. I loved the fact that this writer showed the loving side of God and not the revengeful side of him. I loved how it twisted and turned and I coul
15	neg	On please: I guess you have to be a romance novel lover for this one, and not a very discerning one. All others beware! It is absolute drivel. I figured I was in trouble when a typo is prominently featured on the back cover, but the first page of
16	neg	Awful beyond belief!: I feel I have to write to keep others from wasting their money. This book seems to have been written by a 7th grader with poor grammatical skills for her age! As another reviewer points out, there is a misspelling on the c
17	neg	Don't try to fool us with fake reviews.: It's glaringly obvious that all of the glowing reviews have been written by the same person, perhaps the author herself. They all have the same misspellings and poor sentence structure that is featured in
18	pos	A romantic zen baseball comedy: When you hear folks say that they don't make 'em like that anymore, they might be talking about "BY THE SEA". This is a very cool story about a young Cuban girl searching for identity who stumbles into a c
19	pos	Fashionable Compression Stockings!: After I had a DVT my doctor required me to wear compression stockings. I wore ugly white TED hose and yucky thick brown stockings. Then I found Jobst UltraSheer. They gave me the compression I n
20	pos	Jobst UltraSheer Thigh High: Excellent product. However, they are very difficult to get on for older people. I feel like I've had a full day workout after getting them on. Also, as the day wears on, they begin to roll down from the top and create
21	neg	sizes recommended in the size chart are not real: sizes are much smaller than what is recommended in the chart. I tried to put it and sheer it!. I guess you should not buy this item in the internet..it is better to go to the store and check it
22	neg	mens ultrasheer: This model may be ok for sedentary types, but I'm active and get around alot in my job - consistently found these stockings rolled up down by my ankles! Not Good!! Solution: go with the standard compression stocking, 20
23	pos	Delicious cookie mix: I thought it was funny that I bought this product without knowing it was a mix. I read the header very quickly and just thought it was packaged cookies. But no, it is cookie MIX and I guess I should have noticed that sinc
24	neg	Another Aysmal Digital Copy: Rather than scratches and insect droppings, this one has random pixelations combined with muddy light and vague image resolution. Probably the cue should have been the packaging is straight out of your s
25	pos	A fascinating insight into the life of modern Japanese teens: I thoroughly enjoyed Rising Sons and Daughters. I don't know of any other book that looks at Japanese society from the point of view of its young people poised as they are betwe
26	pos	I liked this album more then I thought I would: I heard a song or two and thought same o same o, but when I listened to songs like "blue angel", "lanna" and "mama" the hair just rose off my neck.Roy is truly an amazing singer with a talent yo
27	neg	Problem with charging smaller AAAs: I have had the charger for more than two years. It charges AA batteries just fine, but has a huge problem securing smaller AAA batteries. To charge the smaller batteries you need to flip down the little bul
28	neg	Works, but not as advertised: I bought one of these chargers, the instructions say the lights stay on while the battery charges...true. The instructions don't say the lights turn off when its done. Which is also true, 24 hours of charging and the
29	neg	Disappointed: I read the reviews, made my purchase and was very disappointed. The charger is convenient by charging all four batteries at once but the charge only lasts a very short time. I now have to go and find batteries that will give me
30	neg	On dear: I was excited to find a book ostensibly about Muslim feminism, but this volume did not live up to the expectations. One essay, among other things, describes the veil as potentially liberating. It doesn't begin to explain how or why An

Code snippet depicting the review files with their labels.

Detail design of Features:

We collected the product reviews data specifically for the amazon products. Preprocessing is done on the extracted data like tokenization, stemming, lemmatization, removing the stopwords and punctuations to extract the meaningful information from the preprocessed data. We used 'Text Summarization' on the reviews to get summarized data from the respective reviews. After that we implemented the machine learning models, SVM and logistic regression on the reviews dataset and observed that SVM is outperforming the logistic regression, obtaining the accuracy of 87. As we know that, in analyzing the sentiments there are two terms known as polarity and subjectivity. Subjectivity refers to person's ideas, opinions whereas polarity refers to the feelings expressed on a particular thing – maybe positively, negatively. Then we used the sentiment analysis to classify positive and negative reviews. Sentiment analysis is implemented using four different approaches TextBlob, VADER, bag of words vectorizer and transformer based models then later visualized the results.

Analysis:

Here, we implemented the text summarization and sentiment analysis on the given text files. When we implement text summarization technique, it gradually reduces the text size and create a summary of our text data. There are two approaches in the text summarization, extractive and abstractive. Using the extractive method, we summarize the text using the traditional approaches whereas abstractive method uses the deep learning techniques like BERT to summarize the sentences. Here in our project, we used the extractive approach to summarize the sentences and below are the steps involved to summarize the text:

1. **Create the word frequency table:** we are calculating the frequency of tokens, how many times a particular token appeared in respective documents and stores them in a dictionary.
2. **Tokenize the sentences:** There can be multiple sentences in each review text file. So each sentence is tokenized with the help of the `sent_tokenize`.
3. **Score the sentences:** Each sentence is given a score, i.e number of words in a particular sentence. Only first ten characters are analyzed and assigned the respective scores to save the memory.

4. **Find the threshold:** we are calculating the average scores of the sentences as the threshold value. We can set any other threshold value if we want for further usage.
5. **Generate the Summary:** If we give any text files as the input then it summarizes the respective text file. If the sentence score is more than the average, then we will make use of that sentence.

One solution that machine learning provides for sentiment analysis involves two main steps. The first step is to learn the model and the second is to classify the unseen data with the help of the trained model. So, we used two machine learning models SVM and logistic regression, to determine the nature of customer reviews as positive or negative. We also implemented sentiment analysis using four python libraries i.e. Textblob, VADER, bag of words vectorizer, and Transformer based models.

Implementation:

In our project, we are using a dataset consisting of the amazon reviews which in turn contains the metadata of all the product reviews. This raw data is preprocessed using the NLP techniques like tokenization, stemming, lemmatization and removing stopwords and punctuations from the reviews. Then we applied the text summarization technique, to extract summarized content from the respective reviews. The processed data is passed through machine learning models, SVM and logistic regression therefore predicting the customer reviews as either positive or negative. We observed that the accuracy is high i.e. 87 if we use SVM. Although both the models performed well obtaining the accuracy of at least 80%. There after we used in built python libraries like TextBlob, VADER, bag of words vectorizer and transformer based models to perform sentiment analysis.

We preprocessed the data and then this data is passed as the input for text summarization. The detailed explanation of all the steps is as shown below.

1. **Tokenization:** Imported a word_tokenize from NLTK library and created a token_splitting method, to break the sentences into words and store the tokens in respective text files.
2. **Stopwords removal:** From all the tokens generated, we removed the stopwords using the defined method remove_stop_words

3. **Stemming:** We made use of PorterStemmer from NLTK, to perform stemming on the given set of data. All the prefixes are removed from the respective tokens if stemming method is called with the updated tokens.
4. **Lemmatization:** When we call the method lemmatize, it converts the given input token to its meaningful root word. We used the WordNetLemmatizer from the NLTK library to implement lemmatization on the given text files.
5. **Text Summarization:** After preprocessing the data using all the above steps, we performed text summarization using the extractive approach where each review when passed through the method returns the summarized content for the respective review.
6. **Applied machine learning models:** Using the machine learning models we are going to predict sentiment of the reviews i.e. either positive or negative. We made use of SVM and logistic regression models on the dataset and obtained the accuracy of at least 80% or above for both the models.

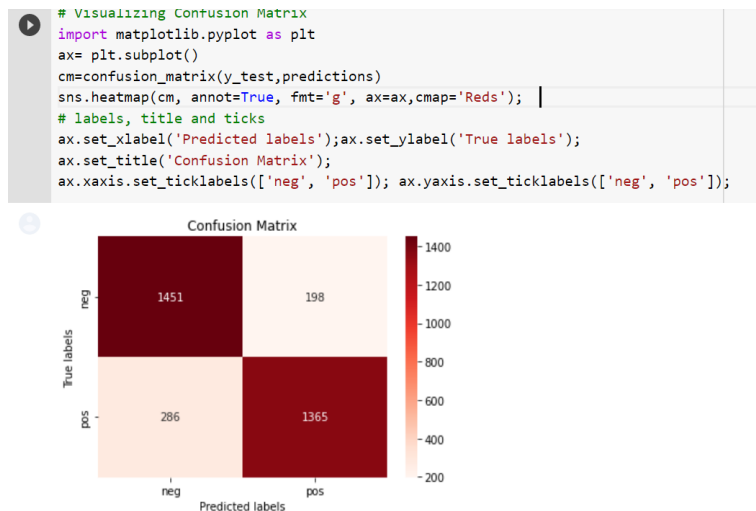
Using the dataset described earlier, where we encoded the text and label separately. We split the whole data into test and train data, and applied the logistic regression and predicted the results out of it.

```
#Visualizing Classification Report
predictions= lr_model.predict(X_test)
report = classification_report(y_test,predictions, output_dict=True)

df_report = pd.DataFrame(report).transpose().round(2)

#df_report.style.background_gradient(cmap='greens').set_precision(2)
cm = sns.light_palette("red", as_cmap=True)
df_report.style.background_gradient(cmap=cm)
```

	precision	recall	f1-score	support
neg	0.840000	0.880000	0.860000	1649.000000
pos	0.870000	0.830000	0.850000	1651.000000
accuracy	0.850000	0.850000	0.850000	0.850000
macro avg	0.850000	0.850000	0.850000	3300.000000
weighted avg	0.850000	0.850000	0.850000	3300.000000



Results obtained using Logistic regression on review files

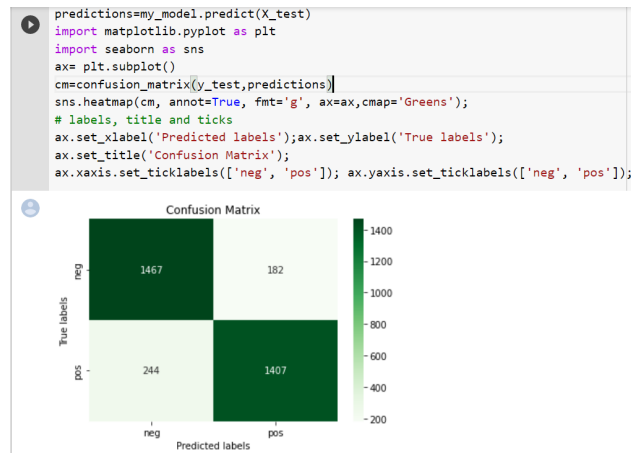
Then Implemented the Support vector machine model on the same train and test data. And visualized the results out of it. By comparing the results, SVM is performing well compared to logistic regression on our dataset.

```
#Visualizing Classification Report
predictions= my_model.predict(X_test)
report = classification_report(y_test,predictions, output_dict=True)

df_report = pd.DataFrame(report).transpose().round(2)

#df_report.style.background_gradient(cmap='greens').set_precision(2)
cm = sns.light_palette("green", as_cmap=True)
df_report.style.background_gradient(cmap=cm)
```

	precision	recall	f1-score	support
neg	0.860000	0.890000	0.870000	1649.000000
pos	0.890000	0.850000	0.870000	1651.000000
accuracy	0.870000	0.870000	0.870000	0.870000
macro avg	0.870000	0.870000	0.870000	3300.000000
weighted avg	0.870000	0.870000	0.870000	3300.000000



Visualizing the classification report and scores obtained using SVM

7. **Sentiment Analysis:** python comes with in-built libraries and tools to perform sentiment analysis. In our project we used 4 different techniques, they are as follows:

TextBlob: It is a python library with which we can perform sentiment analysis easily. It takes the text as the input and returns the sentiment of the text if it is either positive, negative or neutral as output.

```

from textblob import TextBlob

#path to the source files i.e., directory having all the amazon reviews
az_reviews_path= '/content/drive/MyDrive/NLP/amazon_reviews'
reviews_list = os.listdir(az_reviews_path)
pos,neg,neu = 0,0,0

for file in reviews_list:
    #traversing through each file present in the parent directory
    input_files = az_reviews_path + "/" + file
    files_list = open(input_files, "r", encoding="utf-8")
    #reading the content from the file and storing it as text content
    text_content = files_list.read()

    #Determining the Polarity
    polarity = TextBlob(text_content).sentiment.polarity
    #Determining the Subjectivity
    subjectivity = TextBlob(text_content).sentiment.subjectivity
    # print("Polarity :", polarity,"Subjectivity:",subjectivity)
    print(file,end=' ')
    if polarity>0:

```

Code showing the implementation of Textblob.

VADER: It is a rule based sentiment analyzer that is trained on our text files. It's usage is similar to the textblob and very simple to use. At last VaderSentiment object returns dictionary of sentiment scores.

```
[ ] from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
pos,neg = 0,0

for file in reviews_list:
    #traversing through each file present in the parent directory
    input_files = az_reviews_path + "/" + file
    files_list = open(input_files, "r", encoding="utf-8")
    #reading the content from the file and storing it as text content
    text_content = files_list.read()

    sentiment = SentimentIntensityAnalyzer()
    sentiment = sentiment.polarity_scores(text_content)
    print("Sentiment of ",file,"is:", sentiment)
    if(sentiment["pos"]>sentiment["neg"]):
        pos+=1
    else:
        neg+=1

vader_dict = {}
vader_dict["Positive Reviews"] = pos
vader_dict["Negative Reviews"] = neg
```

Implementation of VADER on review files.

Bag of words vectorizer: Instead of using the python libraries, using bag of words we will train our model to predict the sentiment of the respective reviews. First we need to preprocess the review files , then we need to create bag of words using CountVectorizer , there after we trained the model for sentiment classification.

```
#Pre-Processing and Bag of Word Vectorization using Count Vectorizer
from sklearn.feature_extraction.text import CountVectorizer
from nltk.tokenize import RegexpTokenizer
token = RegexpTokenizer(r'[a-zA-Z0-9]+')
cv = CountVectorizer(stop_words='english',ngram_range = (1,1),tokenizer = token.tokenize)

[ ] text_counts = cv.fit_transform(data['review'])
#Splitting the data into trainig and testing
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(text_counts, data['label'], test_size=0.25, random_state=5)
#Training the model
from sklearn.naive_bayes import MultinomialNB
MNB = MultinomialNB()
MNB.fit(X_train, Y_train)
#Caluclating the accuracy score of the model
from sklearn import metrics
predicted = MNB.predict(X_test)
accuracy_score = metrics.accuracy_score(predicted, Y_test)
print("Accuracuy Score: ",accuracy_score)
```

Bag of words vectorizer implementation

Transformer-based models: These models are most advanced Natural Language Processing Techniques. First we imported the pipeline function from the transformers and then called the same function using sentiment analysis. Then it will display the label of the review if it is either positive or negative along with the sentiment score.

```
[ ] from transformers import pipeline

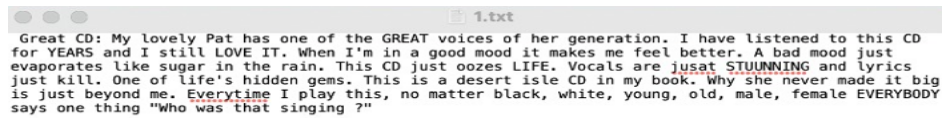
for file in reviews_list:
    #traversing through each file present in the parent directory
    input_files = az_reviews_path + "/" + file
    files_list = open(input_files, "r", encoding="utf-8")
    #reading the content from the file and storing it as text content
    text_content = files_list.read()
    sentiment_pipeline = pipeline("sentiment-analysis")
    output = sentiment_pipeline(text_content)
    print(file,output)
```

Implementing transformer-based models on amazon reviews.

Preliminary Results:

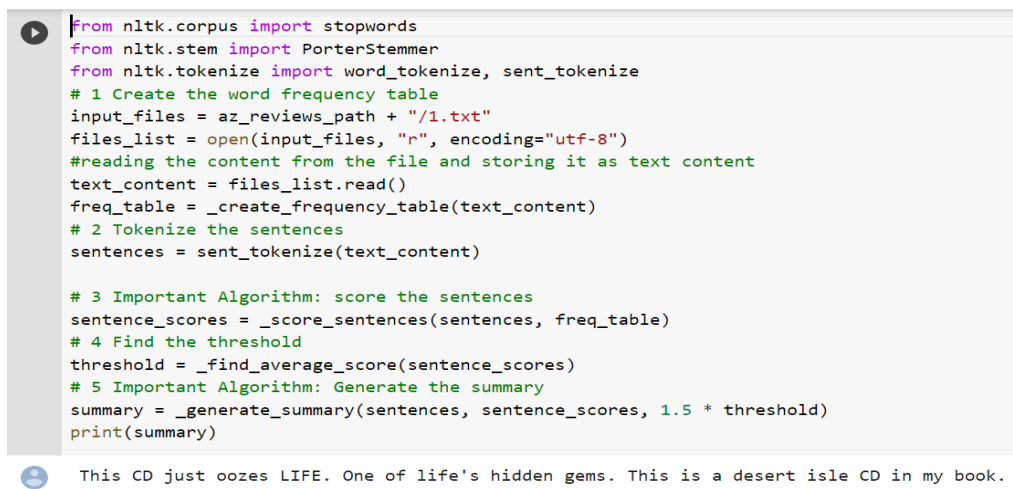
After performing preprocessing on the review files, we implemented the text summarization and sentiment analysis on the preprocessed data following the respective steps. Some of the snapshots of the data before and after performing the text summarization on the respective files.

1. Output snapshot clearly shows the text file 1.txt, before performing the text summarization.



Great CD: My lovely Pat has one of the GREAT voices of her generation. I have listened to this CD for YEARS and I still LOVE IT. When I'm in a good mood it makes me feel better. A bad mood just evaporates like sugar in the rain. This CD just oozes LIFE. Vocals are just STUNNING and lyrics just kill. One of life's hidden gems. This is a desert isle CD in my book. Why she never made it big is just beyond me. Everytime I play this, no matter black, white, young, old, male, female EVERYBODY says one thing "Who was that singing ?"

2. Text summarization on 1.txt file , output in the code snapshot clearly shows the summarized data in the respective file.

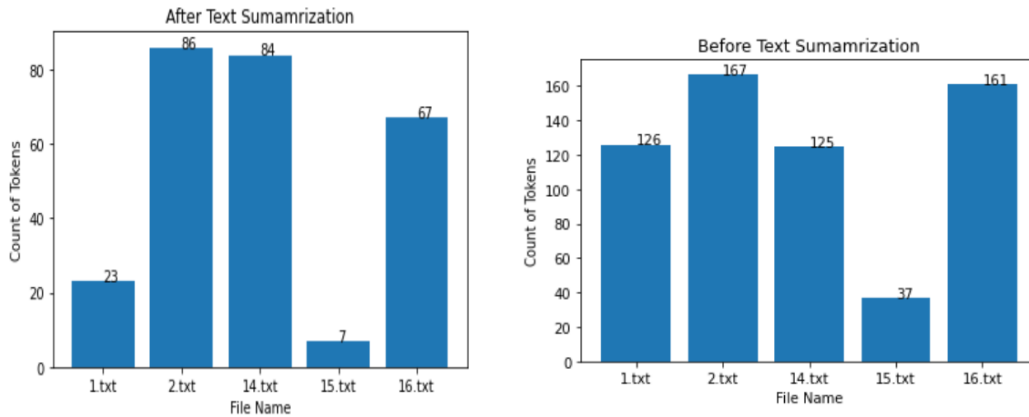


```
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize, sent_tokenize
# 1 Create the word frequency table
input_files = az_reviews_path + "/1.txt"
files_list = open(input_files, "r", encoding="utf-8")
#reading the content from the file and storing it as text content
text_content = files_list.read()
freq_table = _create_frequency_table(text_content)
# 2 Tokenize the sentences
sentences = sent_tokenize(text_content)

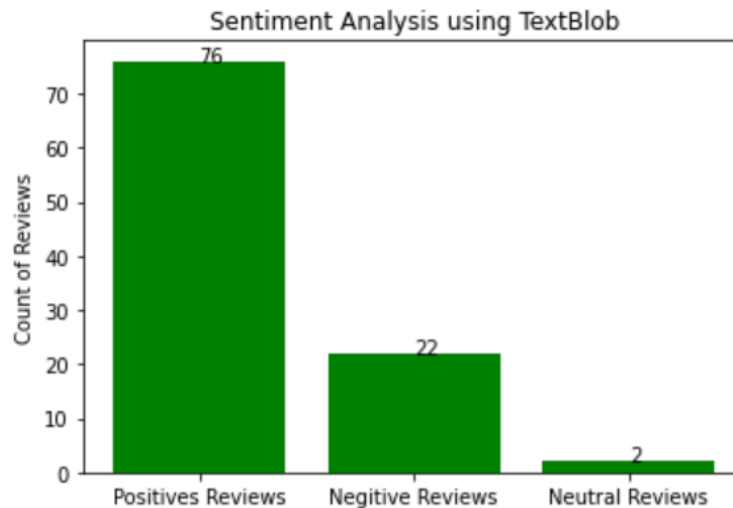
# 3 Important Algorithm: score the sentences
sentence_scores = _score_sentences(sentences, freq_table)
# 4 Find the threshold
threshold = _find_average_score(sentence_scores)
# 5 Important Algorithm: Generate the summary
summary = _generate_summary(sentences, sentence_scores, 1.5 * threshold)
print(summary)
```

This CD just oozes LIFE. One of life's hidden gems. This is a desert isle CD in my book.

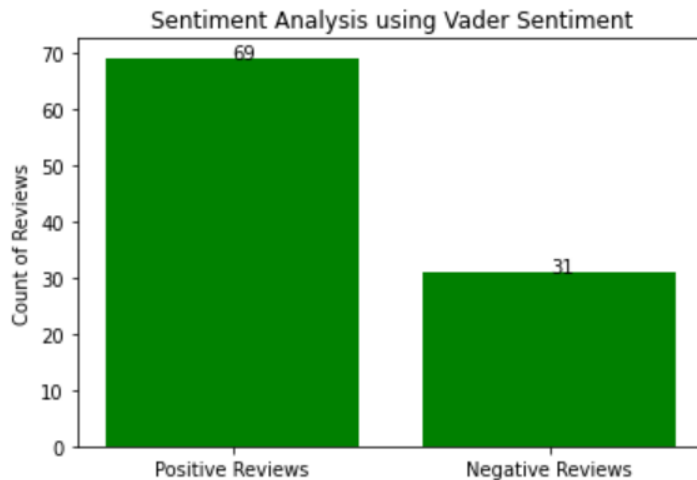
3. We plotted a bar graph between the number of tokens in a respective text file before and after performing the text summarization on certain number of text files.



4. Using the TextBlob, we performed sentiment analysis and predicted the sentiment of review files (positive, negative or neutral). The bar graph is plotted to predict the type of reviews against their count. Below snapshot clearly shows there are maximum number of positive reviews in our dataset.



5. By performing the VADER approach, we classified the review files as either positive or negative. Below bar graph clearly depicts that there are maximum number of positive reviews compared to that of negative ones.



6. Below screenshot depicts the accuracy score determined using bag of words vectorizer approach. Model predicted a very good accuracy score of 81.36%.

```
[ ] text_counts = cv.fit_transform(data['review'])
    #Splitting the data into training and testing
    from sklearn.model_selection import train_test_split
    X_train, X_test, Y_train, Y_test = train_test_split(text_counts, data['label'], test_size=0.25, random_state=5)
    #Training the model
    from sklearn.naive_bayes import MultinomialNB
    MNB = MultinomialNB()
    MNB.fit(X_train, Y_train)
    #Calculating the accuracy score of the model
    from sklearn import metrics
    predicted = MNB.predict(X_test)
    accuracy_score = metrics.accuracy_score(predicted, Y_test)
    print("Accuracy Score: ", accuracy_score)
```

Accuracy Score: 0.8136

7. There after we implemented two machine learning models, SVM and logistic regression. Then compared the results and accuracy score for both the models. BY observing the results, we can say that SVM is giving us good accuracy score compared to that of logistic regression . Both the models performed well and obtained the accuracy above 80%.

```
#Visualizing Classification Report
predictions= lr_model.predict(X_test)
report = classification_report(y_test,predictions, output_dict=True)

df_report = pd.DataFrame(report).transpose().round(2)

#df_report.style.background_gradient(cmap='greens').set_precision(2)
cm = sns.light_palette("red", as_cmap=True)
df_report.style.background_gradient(cmap=cm)
```

	precision	recall	f1-score	support
neg	0.840000	0.880000	0.860000	1649.000000
pos	0.870000	0.830000	0.850000	1651.000000
accuracy	0.850000	0.850000	0.850000	0.850000
macro avg	0.850000	0.850000	0.850000	3300.000000
weighted avg	0.850000	0.850000	0.850000	3300.000000

```
#Visualizing Classification Report
predictions= my_model.predict(X_test)
report = classification_report(y_test,predictions, output_dict=True)

df_report = pd.DataFrame(report).transpose().round(2)

#df_report.style.background_gradient(cmap='greens').set_precision(2)
cm = sns.light_palette("green", as_cmap=True)
df_report.style.background_gradient(cmap=cm)
```

	precision	recall	f1-score	support
neg	0.860000	0.890000	0.870000	1649.000000
pos	0.890000	0.850000	0.870000	1651.000000
accuracy	0.870000	0.870000	0.870000	0.870000
macro avg	0.870000	0.870000	0.870000	3300.000000
weighted avg	0.870000	0.870000	0.870000	3300.000000

Visualized the results from both Logistic regression and SVM

Project Management:

- **Work Completed:**

1. Extracted all the text files, performed the preprocessing on the respective text files like stopwords removal, lemmatization and stemming.
2. On the preprocessed data we implemented the text summarization using the extractive approach.
3. We plotted the bar graph, between the specific text files before and after text summarization.
4. Implemented machine learning models, SVM and logistic regression on our dataset consisting of the review files and predicted the good accuracy score.
5. Performed sentiment analysis on the review files using TextBlob, VADER, bag of words vectorizer and Transformer-based models.

Tasks/Responsibilities/Contributions: we collectively spent some time researching the problem statement and worked on analyzing the problem, steps to be implemented to perform semantic analysis on the review files.

- Avinash: (25%) – worked on text summarization implementation and predicted the results before and after the summarization on the respective text files. Worked on VADER and bag of words approaches to perform sentiment analysis
- Sushmitha: (25%) – worked on creating the workflow diagram and displaying the number of tokens before and after text summarization. Worked on TextBlob sentiment analysis approach.
- Mavya: (25%) – worked on extracting the text files data and visualization of the results. Worked on Transformers based model to predict the sentiment analysis.
- Deeksha: (25%) – worked on preprocessing steps lemmatization, stemming and removal of the stopwords. Worked on implementing the machine learning models, compared and visualized the results.

References:

- 1) <https://techvidvan.com/tutorials/python-sentiment-analysis/>
- 2) https://www.researchgate.net/publication/344869545_Sentiment_Analysis_on_Amazon_reviews
- 3) https://www.researchgate.net/publication/330871275_Natural_Language_Processing_Sentiment_Analysis_and_Clinical_Analytics
- 4) <https://becominghuman.ai/text-summarization-in-5-steps-using-nltk-65b21e352b65>
- 5) <https://www.ijert.org/sentiment-analysis-based-method-for-amazon-product-reviews>
- 6) <https://www.xbyte.io/classifying-amazon-reviews-depending-on-customer-reviews-and-using-nlp.php>
- 7) <https://towardsdatascience.com/sentiment-analysis-on-amazon-reviews-45cd169447ac>
- 8) https://monstott.github.io/sentiment_analysis_and_classification_of_amazon_imdb_and_yelp_reviews
- 9) <https://github.com/RiccardoBonesi/AmazonReview>
- 10) <https://www.analyticsvidhya.com/blog/2021/12/different-methods-for-calculating-sentiment-score-of-text/>
- 11) <https://cs229.stanford.edu/proj2018/report/122.pdf>
- 12) <https://thecleverprogrammer.com/2021/07/20/amazon-product-reviews-sentiment-analysis-with-python/>
- 13) <chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/http://www.diva-portal.org/smash/get/diva2:1241547/FULLTEXT01.pdf>
- 14) <chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://scholar.smu.edu/cgi/viewcontent.cgi?article=1051&context=datasciencereview>
- 15) chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://file.techscience.com/ueditor/files/cm/TSP_CMC-74-1/TSP_CMC_31867/TSP_CMC_31867.pdf

GitHub Link:

<https://github.com/avinashchinta99/nlp-project-group-10>