Sentimental Analysis of eCommerce Platform

Team – 16

Sunny Sumanth Dodda

Sai Krishna Boinapally

Jatin Raj Thodupunuri

Poojitha Achanta

GITHUB - https://github.com/saikrishnaBoinapally/NLP-PROJECT

INTRODUCTION

Consumers in today's world place a great deal of importance on the practice of shopping online since it allows them to reduce the amount of time and effort required to acquire a product. Because of the enormous growth of e-commerce, gathering feedback from customers has become an increasingly important part of identifying their areas of interest and activity. The purpose of doing a sentiment analysis is to ascertain how customers feel about a certain product. This assists other consumers in making judgments about whether or not to purchase the goods. A recommender system that is built on this can offer recommendations to other consumers or show them things that are connected to what they are browsing for. In recent years, sentiment analysis has garnered a considerable deal of interest, as has text categorization based on customer testimonials. Textual reviews, star ratings, and emojis are the many forms that the reviews are presented (Zikang et al., 2020). The shops or service providers may more easily accomplish their goals with the aid of sentiment analysis, which is used to assess the massive amounts of data they collect. Opinion and characteristics based on the information provided on the product's characteristics. A vast quantity of material relating to a certain topic that may be found via social media. People express their thoughts and opinions on social media sites like Twitter, Facebook, and others. After purchasing or utilizing the product, the customer provides feedback about its usefulness. They uploaded a massive amount of information to a variety of different sites. The ability to interpret these reviews provides a huge competitive advantage for businesses, since it enables suppliers to make various judgments concerning the quality of the services or goods being offered. Additionally, the recommender system may be improved with the aid of these evaluations (Yang et al., 2020). We also give information on often purchased items or items that are frequently purchased together, and this is based on the reviews and purchases made by the customer.

Goals and Objectives

The purpose of this study is to provide a system that will be built using a hybrid method that combines context-based engine functionality with stochastic learning. The framework that has been suggested will attempt to create a hybrid recommendation algorithm by combining the many algorithms that are now in use. It will boost performance by overcoming the disadvantages of standard recommendation systems. In addition, the customer sentiment analysis that was carried out for the purpose of this research is an essential instrument for any contemporary company because it enables the business to obtain insights that can be put into action, identify and resolve critical issues with reoccurring patterns that cause customers to feel dissatisfied, strengthen the aspects of a product or service that are responsible for customers' positive emotions, and make decisions that are more data-driven and efficient in general. On a more granular level, the purpose of the customer sentiment analysis carried out on this platform is to provide users with the ability to enhance customer service and, as a result, customer experiences.

Motivation

As the number of online platforms grows quickly, businesses are falling behind and are unable to maintain their competitive advantage over well-funded platforms like Amazon and others. Promoting the application of sentiment analysis is the major factor that propels the platforms that are now the most competitive. Additionally, the majority of the models developed in the earlier study mostly failed to adopt a hybrid technique of stochastic learning, necessitating the use of such a framework in the current study.

Aims and Purposes

The purpose of this study is to provide a system that will be built using a hybrid method that combines context-based engine functionality with stochastic learning. The framework that has been suggested will attempt to create a hybrid recommendation algorithm by combining the many algorithms that are now in use. It will boost performance by overcoming the disadvantages of standard recommendation systems. In addition, the customer sentiment analysis that was carried out for the purpose of this research is an essential instrument for any contemporary company because it enables the business to obtain insights that can be put into action, identify and resolve critical issues with reoccurring patterns that cause customers to feel dissatisfied, strengthen the aspects of a product or service that are responsible for customers' positive emotions, and make decisions that are more data-driven and efficient in general. On a more granular level, the purpose of the customer sentiment analysis carried out on this platform is to provide users with the ability to enhance customer service and, as a result, customer experiences.

Significance

In this day and age of big data, an overwhelming amount of consumer product reviews have been published across various online social media platforms. Consequently, mining the sentiment of customers regarding items can yield significant business knowledge that can enhance the decision-making process of management. Therefore, with the help of the suggested framework for the model, sentiment analysis can be used to investigate a wide range of possibilities, such as the influence sale behavior as well as important brand strategies. Customer analytics, in addition to helping businesses better understand their clients' behaviors, enabling the business to shifts in their clients' requirements. In addition to this, it offers a method for determining which methods of acquiring new consumers and keeping existing ones are successful, as well as which methods are unsuccessful.

Features

The Amazon product reviews that are accessible online are the major source of information for this project. Amazon provides its users with an online option that allows them to review the company's products and services using a star-based scale after making a purchase through the marketplace. Customers also have the option to leave comments, which allow them to describe more explicitly what they took into consideration while assessing the goods. For the sake of this investigation, a data collection consisting of many of these product evaluations will be analyzed

using sentiment analysis. The Amazon product reviews area of this website is where the data that will be used in this research will be acquired from. The dataset that was utilized for the research includes features include:

- ✓ reviews. title,
- ✓ brand,
- ✓ reviews.text,
- ✓ categories,
- ✓ primary categories,
- ✓ And the sentiment – contains negative and positive labels.

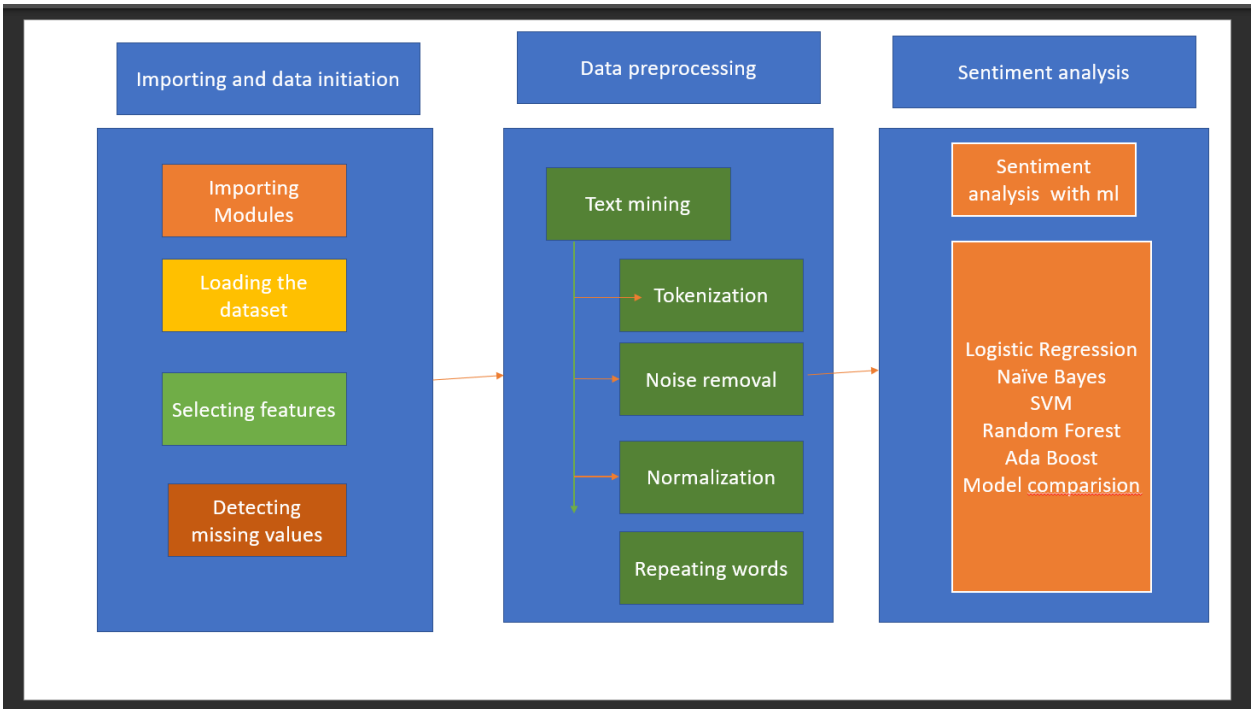
Background

Businesses that are involved in electronic commerce such as Amazon and Flipkart employ a variety of recommendation algorithms to provide customers various choices. Amazon now employs item-item collaborative filtering, which is capable of scaling to enormous datasets and producing high-quality recommendation systems in real time. This system is a sort of data filtering system that attempts to forecast the "rating" or preferences of the person who is interested in the content being filtered. Amazon's Recommendation System adheres to the notion of creating product-based suggestions. This entails determining the degree to which two items are comparable to one another and then advising individual users on which products are most comparable to those examples (Jin et al., 2013). Researchers' primary interest has always been in developing new metrics for comparing the degree to which two different things are same. However, when it comes to a website such as Amazon, it needs to incorporate more criteria in order to propose things to its visitors, such as the product's quality. Because a product of high quality will almost always have a sizeable number of reviews, we are able to provide suggestions based on both the similarity score and the reviews of individual products (Jiang, 2016). E-commerce websites and other online companies, such as social networking and movie/music rendering sites, now typically include recommendation algorithms as an essential component of their offerings. They have a significant influence on the amount of income that these companies bring in and also provide consumers with benefits by lowering the amount of mental effort required to conduct searches and sort through an excessive amount of data (Hu & Zhang, 2012). A customer's experience may be made more personalized by recommender systems, which analyze a user's interactions with a system and then provide suggestions about other products the user would find helpful.

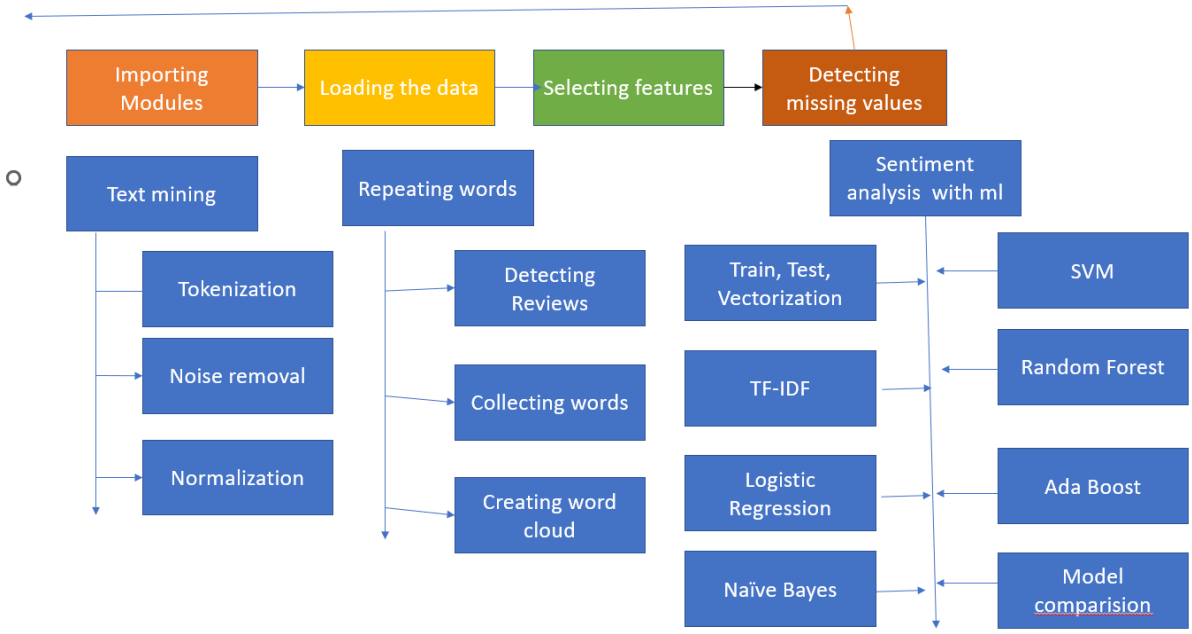
We had referred many articles and scholary articles, journals, and documents for this topic selection. In the references section, we had mentioned some of the journals which are helpful for the project purpose and code development.

Model:

Architecture diagram:



Workflow diagram:



Dataset

The data set we got from the website that is Kaggle and where we can find it here:

https://www.kaggle.com/code/kadirduran/nlp-sentiment-classification-with-ml-and-dl-models/data

In this dataset we have eleven columns and where each column and the dataset give the reviews of the products and by using this dataset. Analysis of data, customer needs to give judgement for the product. We must change the text file to numeric features. That is because machine learning models only use these numeric features.

Data consists of 22641 rows and 11 columns. Each row consists of a comment from the customer that he feels about the project.

Detail description of Features of Dataset:

Coming to dataset there are ten features, and these features are listed in the columns of the dataset and then the features have their own data in it which was given by the customer and got from the customer like details about the product.

Clothing Id:

We have a unique Id for every cloth that refer to the product by which we can say that this product is brought by a customer and the review is in the context of product and depending on this Id we can refer to the product and we can tell the customer have given a genuine review of the product. Then this Id value we take is numerical number and this integer will extend till 22642.

Age:

We take age as a positive because age cannot be number and then this value is taken from customer which customer uses the product and the product can be used by the customer or the product is related to the same age and the product really suits to the customer or not can be known by this age factor most of the time we don't use it but it might help in deep understanding about the product.

Title:

The title of the product is shown here where we need to verify the Id correctly matches to the product and then the title really helps in finding out the product. We take a String value for giving the product name. The customer needs to give the product review in a single word before abbreviating it because, based on the title, we can know whether the review is positive or negative.

Review Text:

This is the place we are going to store the review of the product and where the product in depth review is seen and then the review of the text is stored in a string value. This is the place where the customers will give their reviews and these reviews are stored and this is the important in the features because it is positive then the product is recommended for others and sale may also increases.

Rating:

This is the place where the customer ratings are stored, where the reviews are stored from 1 to 5. If it is 1 then it will be worst and if it is 5 then it will be best so we can say that it is going to be the meter about the product and then we use only one single positive integer to display the product.

Recommended IND:

This is the place where the customer wants to recommend it to others and this area also increases the sales of the product. If he recommends it, then other customers buy it by the way sales are increased. We use a binary digit or 0 and 1 only because they can denote whether to recommend or not that means two variables can be able to tell the product is being reviewed or not.

Positive feedback count:

This is the place where we get the other customers who got this review is useful for them and then the customers found it useful and it shows the count of the people who used this and tells and it tells the number of customers, this is helpful, so we use positive integer.

Division Name:

We are going to use string where there are words like General, General petite and intimate which tell about the quality and the fitting of the product for the customer. So we use string value to store the value.

Department name:

We are going store these values in string format and we are going to group these dresses in many classes like Bottoms, intimates, Dresses. etc. These help us to find out or classify the products.

Class name:

We are going store these values in string format and we are going to group these dresses in many classes like Tops, pants, Dresses. etc. These help us to find out or classify the products.

Dataset:

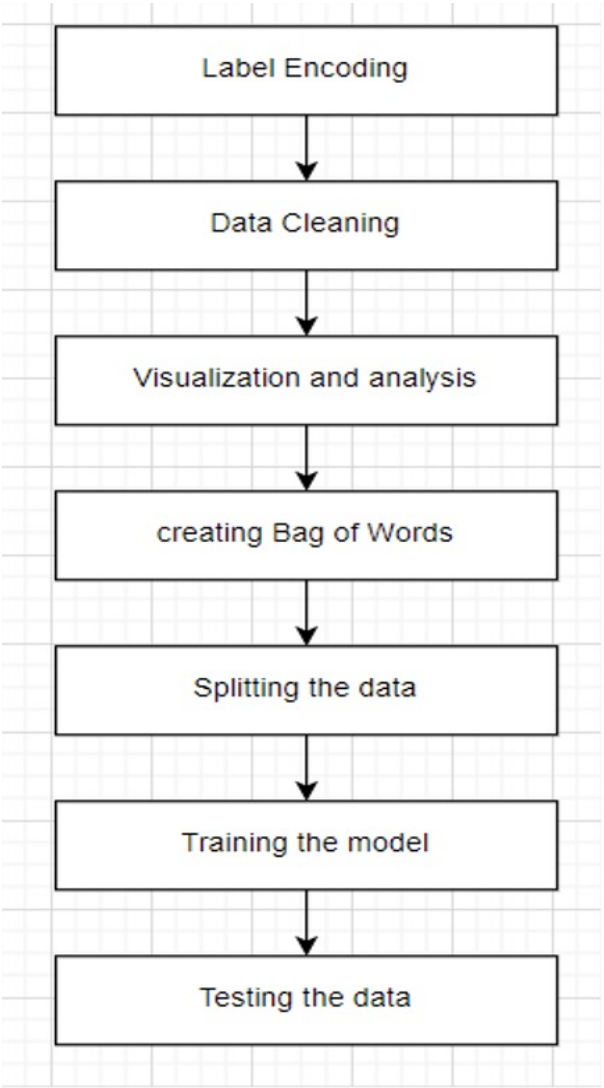
	Clothing ID	Age	Title	Review Te	Rating	Recomme	Positive Fe	Division N	Departme	Class Name
0	767	33		Absolutely	4	1	0	Initmates	Intimate	Intimates
1	1080	34		Love this c	5	1	4	General	Dresses	Dresses
2	1077	60	Some maj	I had such	3	0	0	General	Dresses	Dresses
3	1049	50	My favorit	I love, love	5	1	0	General P	Bottoms	Pants
4	847	47	Flattering	This shirt i	5	1	6	General	Tops	Blouses
5	1080	49	Not for th	I love trac	2	0	4	General	Dresses	Dresses
6	858	39	Cagrcoal s	I aded this	5	1	1	General P	Tops	Knits
7	858	39	Shimmer,	I ordered	4	1	4	General P	Tops	Knits
8	1077	24	Flattering	I love this	5	1	0	General	Dresses	Dresses
9	1077	34	Such a fun	I'm 5"5' ar	5	1	0	General	Dresses	Dresses
10	1077	53	Dress look	Dress runs	3	0	14	General	Dresses	Dresses
11	1095	39		This dress	5	1	2	General P	Dresses	Dresses
12	1095	53	Perfect!!!	More and	5	1	2	General P	Dresses	Dresses
13	767	44	Runs big	Bought	5	1	0	Initmates	Intimate	Intimates
14	1077	50	Pretty par	This is a ni	3	1	1	General	Dresses	Dresses
15	1065	47	Nice, but r	I took thes	4	1	3	General	Bottoms	Pants
16	1065	34	You need	Material a	3	1	2	General	Bottoms	Pants

Analysis of Data:

Feature Selection:

We are now using feature selection then we must store them into a single data frame and then we should process them and then it should contain two column names that are review text, Recommended ID. Then we must find missing values.

Data preprocessing:



Text mining:

Because text is the least organized of all the data kinds, it contains a variety of noise. This indicates that without any pre-processing, the data cannot be easily analyzed. Text preprocessing is the process of cleaning and standardizing text to remove noise and prepare it for analysis.

Tokenization:

One of the main considerations while working on text mining is this stage. Tokenization is the process of breaking down a phrase, sentence, paragraph, or even an entire text document into simpler components, such individual words, or phrases. Tokens are the name for each of these smaller components**.**

Stop words removal:

Any language that does not make sense in the context of the data or the final product might be classified as noise. Language stopwords, such as "is," "am," "the," "of," and "in," URLs or links, upper- and lowercase distinction, punctuation, and words peculiar to a certain business are a few examples. This stage deals with removing all different kinds of distracting text elements.

Normalization:

One word might have numerous representations, which is another sort of textual noise. Examples of alternative spellings of the word "play" include "play," "player," "played," "plays," and "playing." Even though they all have various meanings, they are all comparable in their context. This stage transforms a word's discrepancies into their normalized form (also known as lemma). Stemming and lemmatization are the two processes used for lexicon normalization. For this situation, lemmatization is advised since it will return each word's root form (rather than just stripping suffixes, which is stemming).

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
0
absolutely wonderful silky sexy comfortable
1                    love dress sooo pretty happened find store im glad bc never would ordered online bc petite bought petite love length hit little knee
would definitely true midi someone truly petite
2    high hope dress really wanted work initially ordered petite small usual size found outrageously small small fact could zip reordered petite medium
overall top half comfortable fit nicely bottom ha...
3                                                                                                                  love love love jumpsuit fun flirty
fabulous every time wear get nothing great compliment
4                                                                                shirt flattering due adjustable front tie perfect length wear
legging sleeveless pair well cardigan love shirt
Name: text, dtype: object
```

```
[111] df_ml[df_ml["recommend"] == 0].head(3)
```

	text	recommend
2	high hope dress really wanted work initially ordered petite small usual size found outrageously small small fact could zip reordered petite medium overall top half comfortable fit nicely bottom ha...	0
5	love tracy reese dress one petite foot tall usually wear brand dress pretty package lot dress skirt long full overwhelmed small frame stranger alteration shortening narrowing skirt would take away...	0
10	dress run small esp zipper area run ordered sp typically fit tight material top look feel cheap even pulling cause rip fabric pretty disappointed going christmas dress year needle say going back	0

```
df_ml[df_ml["recommend"] == 1].head(3)
```

	text	recommend
0	absolutely wonderful silky sexy comfortable	1
1	love dress sooo pretty happened find store im glad bc never would ordered online bc petite bought petite love length hit little knee would definitely true midi someone truly petite	1
3	love love love jumpsuit fun flirty fabulous every time wear get nothing great compliment	1

Repeated words:

The most frequent terms in each target class will now be represented in a Word Cloud that will be used for reviews. The magnitude of each word in a word cloud, a data visualization approach for expressing text data, shows its frequency or relevance. Using a word cloud, significant textual data points may be emphasized. We will split the word clouds for favorable and unfavorable assessments. By examining a review's recommendation status, we may determine if it is good or negative.

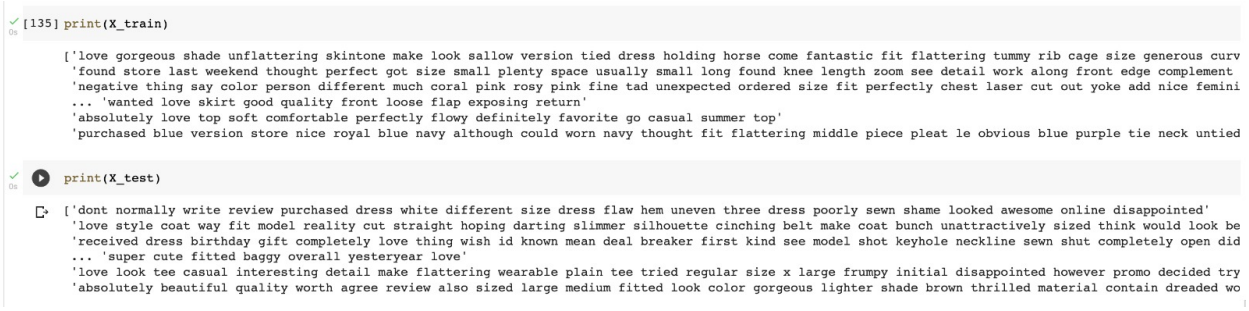
```
[135] print(X_train)

['love gorgeous shade unflattering skintone make look sallow version tied dress holding horse come fantastic fit flattering tummy rib cage size generous curv
 'found store last weekend thought perfect got size small plenty space usually small long found knee length zoom see detail work along front edge complement
 'negative thing say color person different much coral pink rosy pink fine tad unexpected ordered size fit perfectly chest laser cut out yoke add nice femini
 ... 'wanted love skirt good quality front loose flap exposing return'
 'absolutely love top soft comfortable perfectly flowy definitely favorite go casual summer top'
 'purchased blue version store nice royal blue navy although could worn navy thought fit flattering middle piece pleat le obvious blue purple tie neck untied

print(X_test)

['dont normally write review purchased dress white different size dress flaw hem uneven three dress poorly sewn shame looked awesome online disappointed'
 'love style coat way fit model reality cut straight hoping darting slimmer silhouette cinching belt make coat bunch unattractively sized think would look be
 'received dress birthday gift completely love thing wish id known mean deal breaker first kind see model shot keyhole neckline sewn shut completely open did
 ... 'super cute fitted baggy overall yesteryear love'
 'love look tee casual interesting detail make flattering wearable plain tee tried regular size x large frumpy initial disappointed however promo decided try
 'absolutely beautiful quality worth agree review also sized large medium fitted look color gorgeous lighter shade brown thrilled material contain dreaded wo
```

Data distribution:

We are going to distribute the data into two files where we are going to distribute into testing and training. Because, we must do vectorization and train-test split as data pretreatment stages before going on to modeling. The most common input for machine learning algorithms is feature vectors of numbers. We thus require a method for turning each text document into a numeric vector while working with text documents. Text vectorization is the method in question. The method of vectorization that will be used in this instance is to represent each text as a vector of word counts

Training data: It is the dataset used to train machine learning algorithms. Train data is used to train the model and then we are going to the train data into numerical data as 0,1 which helps in plotting the graphs for the required machine learning models.

Testing Data: It is the dataset used to test machine learning algorithms. Test data is used to test the model and then we are going to the test data into numerical data as 0,1 which helps in plotting the graphs for the required machine learning models.



Implementation

Despite the numerous other kinds of recommendation systems, such as ones based on quality, classification models, feature recommendation systems, as well as more, the sort of recommendation system that will be the topic of this paper is an overview recommendation system in machine learning, and we will explore how to create it using python code. In the past, recommendations were determined by looking at product patterns, which meant that the product that was being used most frequently was the one that was suggested to practically everyone. Other methods of determining recommendations made use of rating histories.

Jupiter notebook/ Google Colab

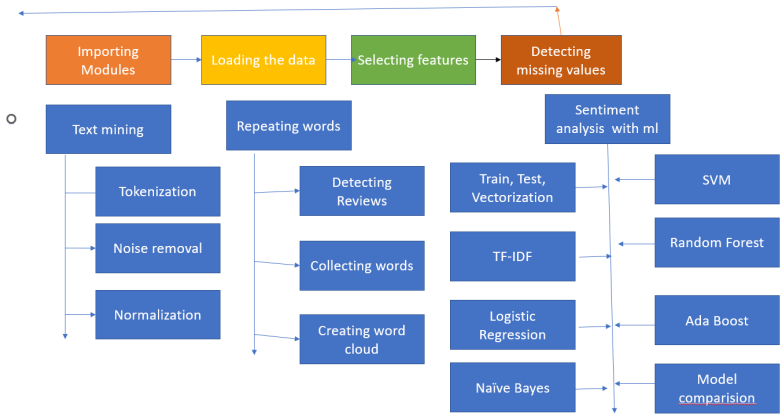
To this recommender, the Jupiter notebook implementation that Anaconda supplies is utilized. The scalable machine learning library known as Jupiter notebook. The Jupiter notebook is particularly adept at doing computations in an iterative fashion, which enables it to operate quickly.

The Anaconda technique with Weight Regularization is what the Jupiter notebook implements. It is a distributed version of the approach. At the moment, model-based collaborative filtering is supported by google colab.

The selection of the model and the hyper-parameters

The main basic parameters that are specified for the recommendation system are the amount of iterations, lambda, as well as rank (the number of latent components). When there was no split in the dataset, the values for rank were set to [2, 5, 10, 20], but they were [8, 10, 20] when there was a split of [0.6, 0.2, 0.2]. The value of lambda might run from [0.001 to 50]. Iterations range from 5 to 20 when there is no split in the dataset, but they are always set to 20 when there is a [0.6, 0.2, 0.2] split in the dataset.

Algorithm:



In this model we have three states were

1. Importing data and data initiation.

2. Data preprocessing.

3. Sentimental analysis using ML and DL.

The main theme of the project is to give predictions whether a customer is going to suggest the product or not. And these products are useful for the customer who bought it and how many people are going to get influenced by the reviews. In this we are mainly focusing on the review text and ignoring the other features.

Project is divided into five tasks:

1. Importing and data initialization.
2. Data cleaning.
3. Text mining.
4. Repetition word removing.
5. Sentiment Classification with ml.

Tasks:

T1: In the first step we are going to import the libraries.

```
# !pip install pyforest
# 1-Import Libraies
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as stats
%matplotlib inline
import statsmodels.api as sm
import statsmodels.formula.api as smf
import missingno as msno

from sklearn.compose import make_column_transformer

# Scaling
from sklearn.preprocessing import scale
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import PolynomialFeatures
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import PowerTransformer
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import RobustScaler


# Importing plotly and cufflinks in offline mode
import plotly.express as px
import cufflinks as cf
import plotly.offline
cf.go_offline()
cf.set_config_file(offline=False, world_readable=True)

# Ignore Warnings
import warnings
warnings.filterwarnings("ignore")
warnings.warn("this will not show")
```

T2: We must load the data.

```
df = pd.read_csv("Womens Clothing E-Commerce Reviews.csv")
df.head()
```

	Unnamed: 0	Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Department Name	Class Name
0	0	767	33	NaN	Absolutely wonderful - silky and sexy and comfortable	4	1	0	Initmates	Intimate	Intimates
1	1	1080	34	NaN	Love this dress! it's sooo pretty. i happened to find it in a store, and i'm glad i did bc i never would have ordered it online bc it's petite. i bought a petite and am 5'8". i love the length...	5	1	4	General	Dresses	Dresses
2	2	1077	60	Some major design flaws	I had such high hopes for this dress and really wanted it to work for me. i initially ordered the petite small (my usual size) but i found this to be outrageously small. so small in fact that i co...	3	0	0	General	Dresses	Dresses
3	3	1049	50	My favorite buy!	I love, love, love this jumpsuit. it's fun, flirty, and fabulous! every time i wear it, i get nothing but great compliments!	5	1	0	General Petite	Bottoms	Pants
4	4	847	47	Flattering shirt	This shirt is very flattering to all due to the adjustable front tie. it is the perfect length to wear with leggings and it is sleeveless so it pairs well with any cardigan. love this shirt!!!	5	1	6	General	Tops	Blouses

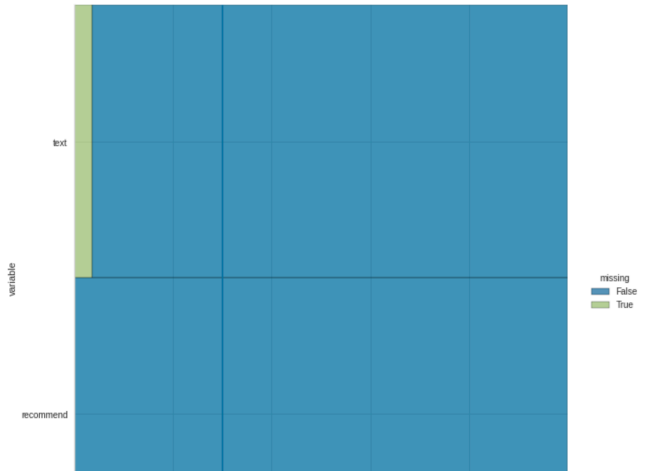
T3: In this we are going to select all the required features and naming it again for columns.

```
RangeIndex: 23486 entries, 0 to 23485
Data columns (total 11 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   Unnamed: 0               23486 non-null  int64 
 1   Clothing ID              23486 non-null  int64 
 2   Age                      23486 non-null  int64 
 3   Title                    19676 non-null  object
 4   Review Text              22641 non-null  object
 5   Rating                   23486 non-null  int64 
 6   Recommended IND          23486 non-null  int64 
 7   Positive Feedback Count  23486 non-null  int64 
 8   Division Name            23472 non-null  object
 9   Department Name          23472 non-null  object
 10  Class Name               23472 non-null  object
dtypes: int64(6), object(5)
memory usage: 2.0+ MB
None
--------------------------------------------------------------------------------
Number of Uniques:
Unnamed: 0                 23486
Clothing ID                 1206
Age                           77
Title                      13993
Review Text                22634
Rating                         5
Recommended IND                2
Positive Feedback Count       82
Division Name                  3
Department Name                6
Class Name                    20
dtype: int64
--------------------------------------------------------------------------------
Missing Values:
                Missing_Number  Missing_Percent
Title                     3810            0.162
Review Text                845            0.036
Division Name               14            0.001
```

T4: Find the missing values.

```
plt.figure(figsize = (10, 5))
sns.displot(data = df_ml.isnull().melt(value_name = "missing"),y = "variable",hue = "missing",multiple = "fill",height = 9.25)
plt.axvline(0.3, color = "b");
```

```
<Figure size 720x360 with 0 Axes>
```



T5: Text mining which involves Data preprocessing techniques, In this we had done tokenization.

Text Mining

Tokenization, Noise Removal and Lexicon Normalization

```
df_ml.head()
```

	text	recommend
0	Absolutely wonderful - silky and sexy and comfortable	1
1	Love this dress! it's sooo pretty. i happened to find it in a store, and i'm glad i did bc i never would have ordered it online bc it's petite. i bought a petite and am 5'8". i love the length...	1
2	I had such high hopes for this dress and really wanted it to work for me. i initially ordered the petite small (my usual size) but i found this to be outrageously small. so small in fact that i co...	0
3	I love, love, love this jumpsuit. it's fun, flirty, and fabulous! every time i wear it, i get nothing but great compliments!	1
4	This shirt is very flattering to all due to the adjustable front tie. it is the perfect length to wear with leggings and it is sleeveless so it pairs well with any cardigan. love this shirt!!!	1

```
import nltk
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('omw-1.4')
stop_words = stopwords.words('english')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
```

```
[205] def cleaning(data):
```

T6: Removing noise words is seen here.

```
def cleaning(data):

    text_tokens = word_tokenize(data.replace("'", "").lower()) #1. Tokenize
    tokens_without_punc = [w for w in text_tokens if w.isalpha()]  #2. Remove Puncs
    tokens_without_sw = [t for t in tokens_without_punc if t not in stop_words] #3. Removing Stopwords
    text_cleaned = [WordNetLemmatizer().lemmatize(t) for t in tokens_without_sw]  #4. lemma
    return " ".join(text_cleaned) #joining
```

```
[206] nltk.download('wordnet')
df_ml["text"] = df_ml["text"].apply(cleaning)
df_ml["text"].head()
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
0
absolutely wonderful silky sexy comfortable
1                    love dress sooo pretty happened find store im glad bc never would ordered online bc petite bought petite love length hit little knee
would definitely true midi someone truly petite
2    high hope dress really wanted work initially ordered petite small usual size found outrageously small small fact could zip reordered petite medium
overall top half comfortable fit nicely bottom ha...
3                                                                                                                    love love love jumpsuit fun flirty
fabulous every time wear get nothing great compliment
4                                                                                                shirt flattering due adjustable front tie perfect length wear
legging sleeveless pair well cardigan love shirt
Name: text, dtype: object
```

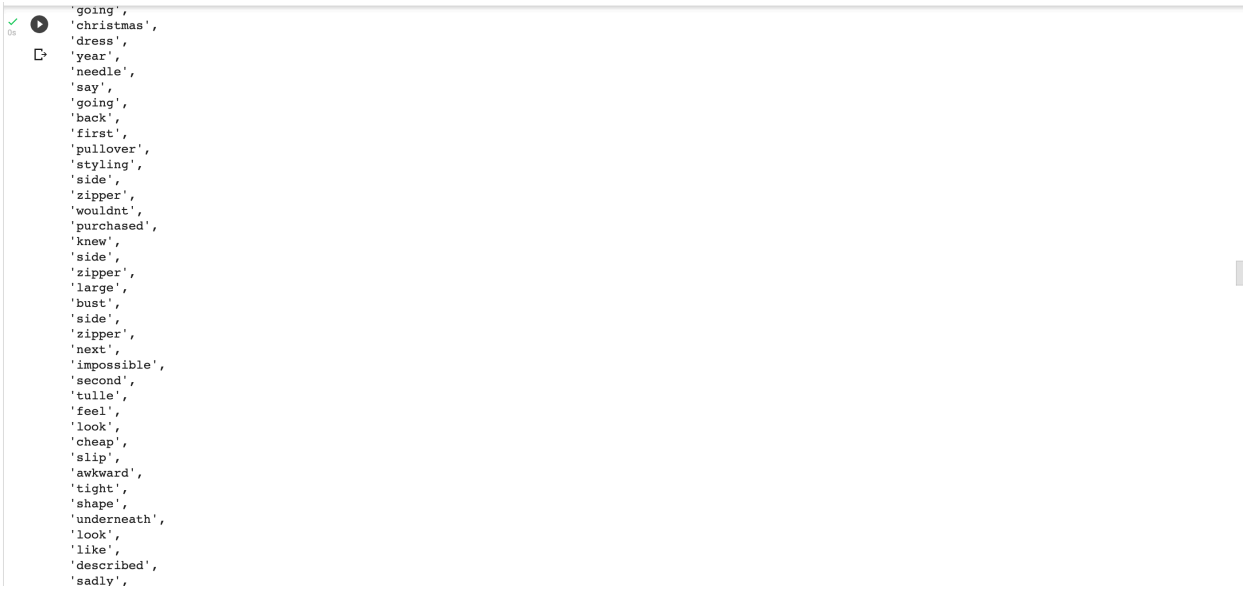
WordCloud - Repetition of Words

```
[207] df_ml[df_ml["recommend"] == 0].head(3)
```

	text	recommend

T7: We are going to detect all the reviews.

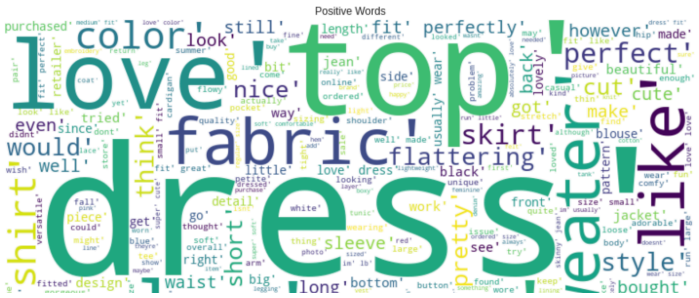
T8: Collect all the individual so that we can find the repeated words.



T9: Create word cloud which deletes all the positive/ negative words.

```
[216]
```

```
[217] wordcloud = WordCloud(width = 800, height = 400, background_color = "white", max_words = 250).generate(str(positive_words))

      plt.figure(figsize = (14, 14))
      plt.imshow(wordcloud)
      plt.title('Positive Words')
      plt.axis("off")
      plt.show()
```



T10: Splitting the data and converting it into numeric data.

```
[230] 18112 rows × 10956 columns
```

```
[231] print(X_train)

      ['love gorgeous shade unflattering skintone make look sallow version tied dress holding horse come fantastic fit flattering tummy rib cage size generous curv
       'found store last weekend thought perfect got size small plenty space usually small long found knee length zoom see detail work along front edge complement
       'negative thing say color person different much coral pink rosy pink fine tad unexpected ordered size fit perfectly chest laser cut out yoke add nice femini
       ... 'wanted love skirt good quality front loose flap exposing return'
       'absolutely love top soft comfortable perfectly flowy definitely favorite go casual summer top'
       'purchased blue version store nice royal blue navy although could worn navy thought fit flattering middle piece pleat le obvious blue purple tie neck untied
```

```
      print(X_test)

      ['dont normally write review purchased dress white different size dress flaw hem uneven three dress poorly sewn shame looked awesome online disappointed'
       'love style coat way fit model reality cut straight hoping darting slimmer silhouette cinching belt make coat bunch unattractively sized think would look be
       'received dress birthday gift completely love thing wish id known mean deal breaker first kind see model shot keyhole neckline sewn shut completely open did
       ... 'super cute fitted baggy overall yesteryear love'
       'love look tee casual interesting detail make flattering wearable plain tee tried regular size x large frumpy initial disappointed however promo decided try
       'absolutely beautiful quality worth agree review also sized large medium fitted look color gorgeous lighter shade brown thrilled material contain dreaded wo
```

```
[233] print(y_train)

      [0 0 0 ... 1 0 0]
```

```
[234] print(y_test)

      [1 0 0 ... 0 0 0]
```

```
[235]: import plot_confusion_matrix, confusion_matrix, classification_report, accuracy_score, f1_score, recall_score, precision_score, average_precision_score
```

T11: Implementation of naïve bayes model.

NaiveBayes

```
[245] from sklearn.naive_bayes import MultinomialNB

[246] nbmulti_count = MultinomialNB()
      nbmulti_count.fit(X_train_count,y_train)

      MultinomialNB()

      print("NBMulti_Count Model")
      print("-------------------")
      eval(nbmulti_count, X_train_count, X_test_count)

      NBMulti_Count Model
      -------------------
      [[3466  243]
       [ 258  562]]
      Test_Set
                    precision    recall  f1-score   support

                 0       0.93      0.93      0.93      3709
                 1       0.70      0.69      0.69       820

          accuracy                           0.89      4529
         macro avg       0.81      0.81      0.81      4529
      weighted avg       0.89      0.89      0.89      4529

      Train_Set
                    precision    recall  f1-score   support

                 0       0.95      0.94      0.95     14831
                 1       0.75      0.78      0.76      3281
```

T12: Implementation of random forest model.

RandomForest

```
      from sklearn.ensemble import RandomForestClassifier

      rf_count = RandomForestClassifier(n_estimators = 200, max_depth = 11, class_weight = "balanced", random_state = 101, n_jobs = -1)
      rf_count.fit(X_train_count, y_train)

      RandomForestClassifier(class_weight='balanced', max_depth=11, n_estimators=200,
                             n_jobs=-1, random_state=101)

[255] print("RF_Count Model")
      print("--------------")
      eval(rf_count, X_train_count, X_test_count)

      RF_Count Model
      --------------
      [[3175  534]
       [ 164  656]]
      Test_Set
                    precision    recall  f1-score   support

                 0       0.95      0.86      0.90      3709
                 1       0.55      0.80      0.65       820

          accuracy                           0.85      4529
         macro avg       0.75      0.83      0.78      4529
      weighted avg       0.88      0.85      0.86      4529

      Train_Set
                    precision    recall  f1-score   support

                 0       0.97      0.88      0.93     14831
                 1       0.63      0.89      0.74      3281
```

T13: Implementation of Logistic regression.

Logistic Regression

```
[237] from sklearn.linear_model import LogisticRegression

      logreg_count = LogisticRegression(C = 0.1, max_iter = 1000, class_weight = 'balanced', random_state = 101)
      logreg_count.fit(X_train_count,y_train)
```

```
LogisticRegression(C=0.1, class_weight='balanced', max_iter=1000,
                   random_state=101)
```

```
print("LogReg_Count Model")
print ("------------------")
eval(logreg_count, X_train_count, X_test_count)
```

```
LogReg_Count Model
------------------
[[3208  501]
 [ 122  698]]
Test_Set
              precision    recall  f1-score   support

           0       0.96      0.86      0.91      3709
           1       0.58      0.85      0.69       820

    accuracy                           0.86      4529
   macro avg       0.77      0.86      0.80      4529
weighted avg       0.89      0.86      0.87      4529

Train_Set
              precision    recall  f1-score   support

           0       0.98      0.89      0.93     14831
           1       0.65      0.93      0.77      3281
```

T14: Model comparison.

```
[262] compare = compare.sort_values(by="Recall_Score", ascending=True)
      compare
```

	Model	F1_Score	Recall_Score	Average_Precision_Score
0	NaiveBayes(Multi)_Count	0.676	0.681	0.728
2	Random Forest_Count	0.655	0.788	0.702
1	LogReg_Count	0.913	0.873	0.732

```
[263] compare = compare.sort_values(by="F1_Score", ascending=True)
      compare
```

	Model	F1_Score	Recall_Score	Average_Precision_Score
2	Random Forest_Count	0.655	0.788	0.702
0	NaiveBayes(Multi)_Count	0.676	0.681	0.728
1	LogReg_Count	0.913	0.873	0.732

```
compare = compare.sort_values(by="Average_Precision_Score", ascending=True)
compare
```

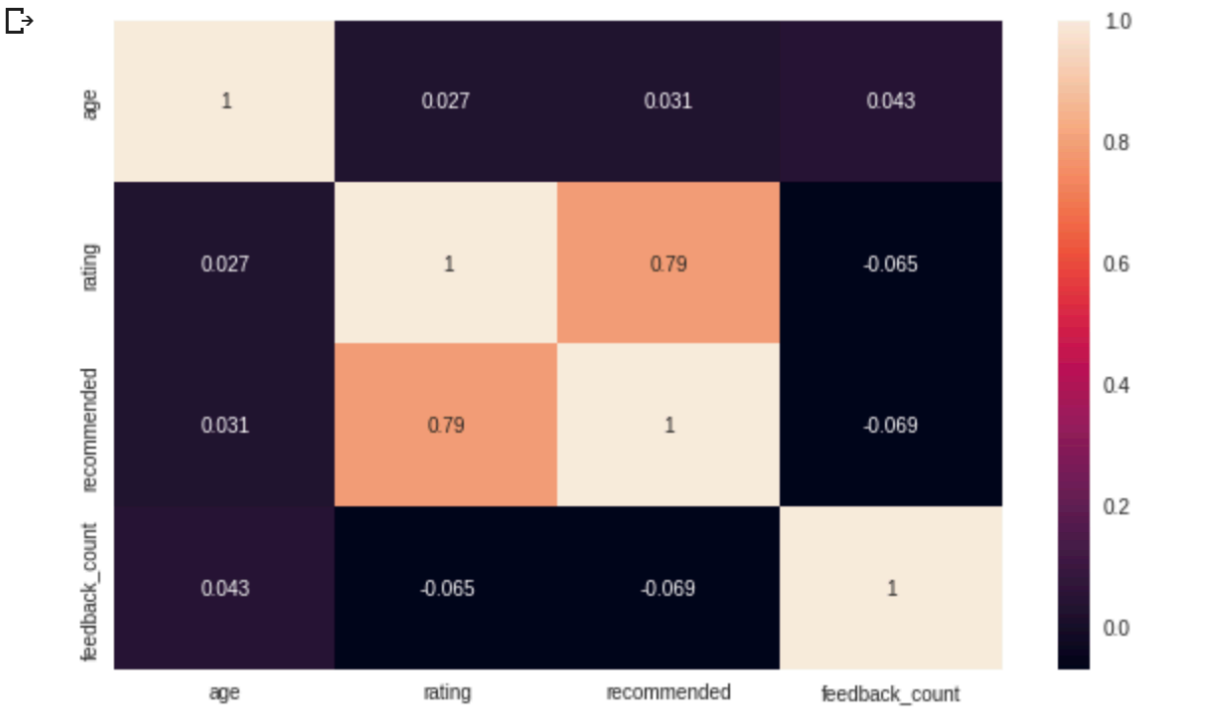
	Model	F1_Score	Recall_Score	Average_Precision_Score
2	Random Forest_Count	0.655	0.788	0.702
0	NaiveBayes(Multi)_Count	0.676	0.681	0.728
1	LogReg_Count	0.913	0.873	0.732

Results

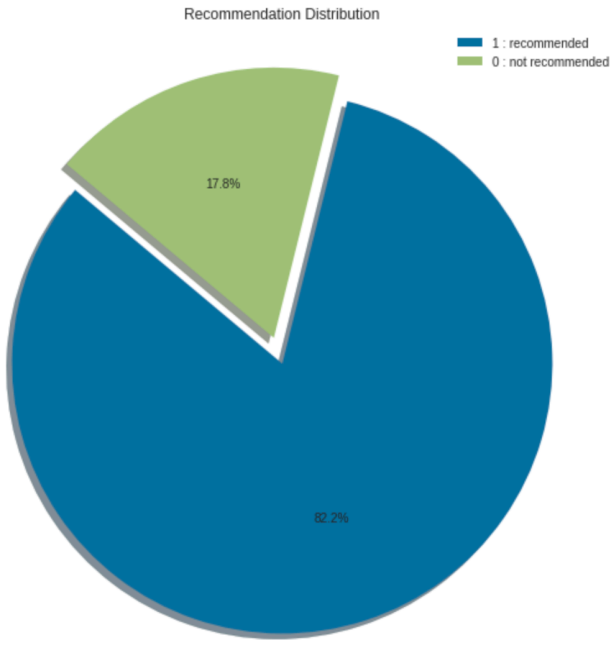
- Dataset initiation

	Unnamed: 0	Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Department Name	Class Name
0	0	767	33	NaN	Absolutely wonderful - silky and sexy and comfortable	4	1	0	Initmates	Intimate	Intimates
1	1	1080	34	NaN	Love this dress! it's sooo pretty. i happened to find it in a store, and i'm glad i did bc i never would have ordered it online bc it's petite. i bought a petite and am 5'8". i love the length...	5	1	4	General	Dresses	Dresses
2	2	1077	60	Some major design flaws	I had such high hopes for this dress and really wanted it to work for me. i initially ordered the petite small (my usual size) but i found this to be outrageously small. so small in fact that i co...	3	0	0	General	Dresses	Dresses
3	3	1049	50	My favorite buy!	I love, love, love this jumpsuit. it's fun, flirty, and fabulous! every time i wear it, i get nothing but great compliments!	5	1	0	General Petite	Bottoms	Pants
4	4	847	47	Flattering shirt	This shirt is very flattering to all due to the adjustable front tie. it is the perfect length to wear with leggings and it is sleeveless so it pairs well with any cardigan. love this shirt!!!	5	1	6	General	Tops	Blouses

- SNS Heatmap



- Recommended Distribution



- Cleaning the data

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
0
absolutely wonderful silky sexy comfortable
1                    love dress sooo pretty happened find store im glad bc never would ordered online bc petite bought petite love length hit little knee
would definitely true midi someone truly petite
2    high hope dress really wanted work initially ordered petite small usual size found outrageously small small fact could zip reordered petite medium
overall top half comfortable fit nicely bottom ha...
3                                                                                                                love love love jumpsuit fun flirty
fabulous every time wear get nothing great compliment
4                                                                                shirt flattering due adjustable front tie perfect length wear
legging sleeveless pair well cardigan love shirt
Name: text, dtype: object
```

- Positive and Negative Differentiation

```
[207] df_ml[df_ml["recommend"] == 0].head(3)
```

	text	recommend
2	high hope dress really wanted work initially ordered petite small usual size found outrageously small small fact could zip reordered petite medium overall top half comfortable fit nicely bottom ha...	0
5	love tracy reese dress one petite foot tall usually wear brand dress pretty package lot dress skirt long full overwhelmed small frame stranger alteration shortening narrowing skirt would take away...	0
10	dress run small esp zipper area run ordered sp typically fit tight material top look feel cheap even pulling cause rip fabric pretty disappointed going christmas dress year needle say going back	0

```
df_ml[df_ml["recommend"] == 1].head(3)
```

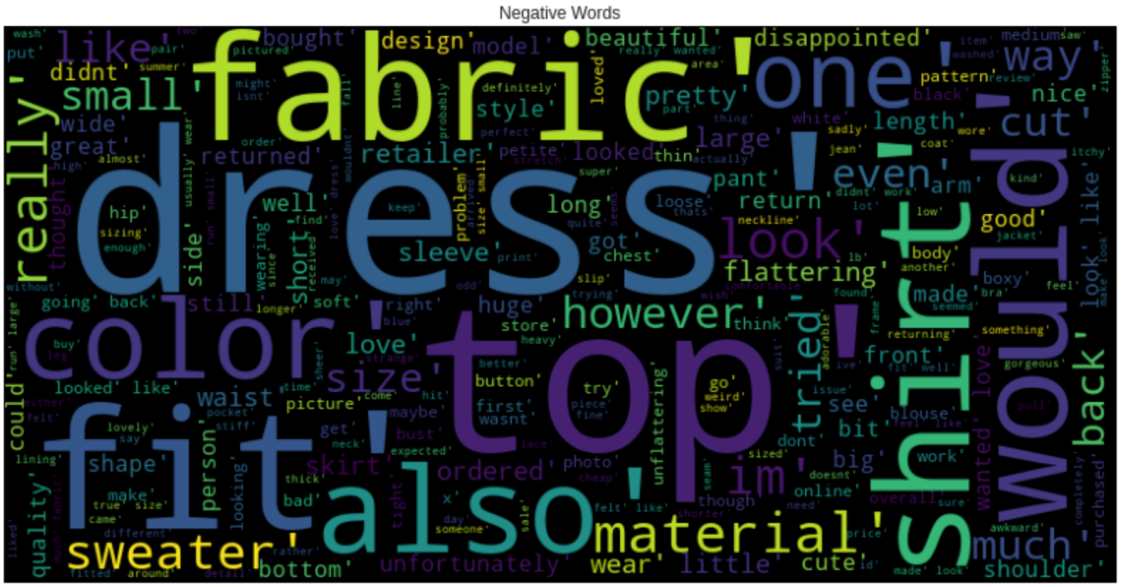
	text	recommend
0	absolutely wonderful silky sexy comfortable	1
1	love dress sooo pretty happened find store im glad bc never would ordered online bc petite bought petite love length hit little knee would definitely true midi someone truly petite	1
3	love love love jumpsuit fun flirty fabulous every time wear get nothing great compliment	1

```
[209] " ".join(df_ml["text"]).split()
```

- Word cloud for all words

- Word cloud for positive words

- Word cloud for negative words



- X_train, X_test, y_train, y_test

```
[231] print(X_train)

['love gorgeous shade unflattering skintone make look sallow version tied dress holding horse come fantastic fit flattering tummy rib cage size generous curv
 'found store last weekend thought perfect got size small plenty space usually small long found knee length zoom see detail work along front edge complement
 'negative thing say color person different much coral pink rosy pink fine tad unexpected ordered size fit perfectly chest laser cut out yoke add nice femini
 ... 'wanted love skirt good quality front loose flap exposing return'
 'absolutely love top soft comfortable perfectly flowy definitely favorite go casual summer top'
 'purchased blue version store nice royal blue navy although could worn navy thought fit flattering middle piece pleat le obvious blue purple tie neck untied

[232] print(X_test)

['dont normally write review purchased dress white different size dress flaw hem uneven three dress poorly sewn shame looked awesome online disappointed'
 'love style coat way fit model reality cut straight hoping darting slimmer silhouette cinching belt make coat bunch unattractively sized think would look be
 'received dress birthday gift completely love thing wish id known mean deal breaker first kind see model shot keyhole neckline sewn shut completely open did
 ... 'super cute fitted baggy overall yesteryear love'
 'love look tee casual interesting detail make flattering wearable plain tee tried regular size x large frumpy initial disappointed however promo decided try
 'absolutely beautiful quality worth agree review also sized large medium fitted look color gorgeous lighter shade brown thrilled material contain dreaded wo

[233] print(y_train)

[0 0 0 ... 1 0 0]

print(y_test)

[1 0 0 ... 0 0 0]
```

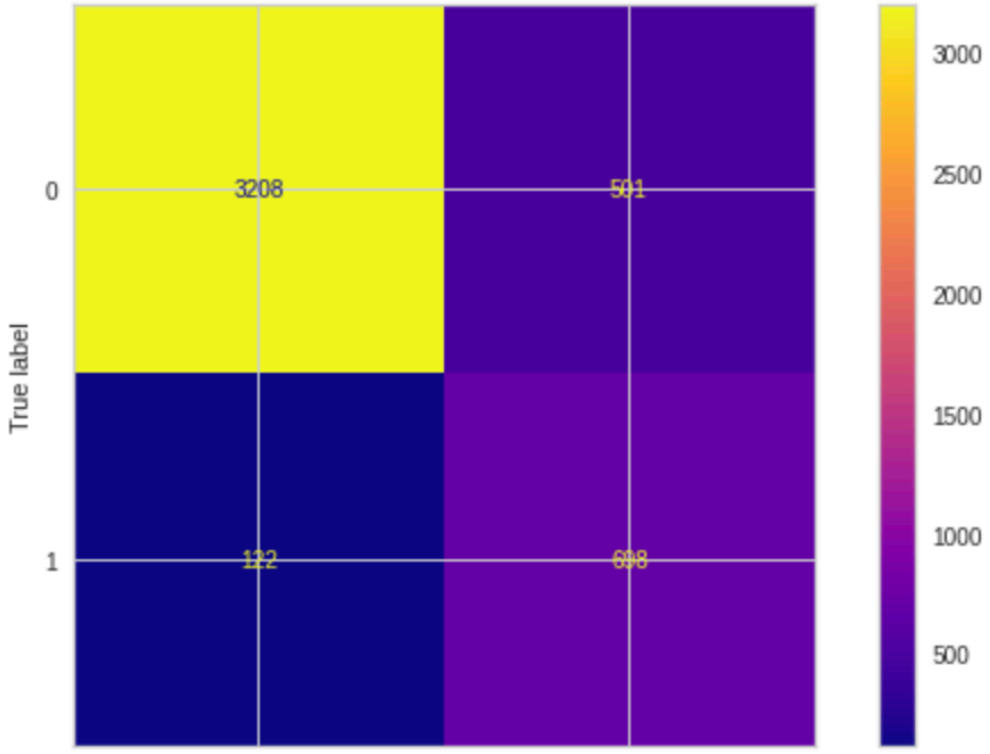
- Logistic Regression

0s

	precision	recall	f1-score	support
0	0.96	0.86	0.91	3709
1	0.58	0.85	0.69	820
accuracy			0.86	4529
macro avg	0.77	0.86	0.80	4529
weighted avg	0.89	0.86	0.87	4529

Train_Set

	precision	recall	f1-score	support
0	0.98	0.89	0.93	14831
1	0.65	0.93	0.77	3281
accuracy			0.90	18112
macro avg	0.82	0.91	0.85	18112
weighted avg	0.92	0.90	0.90	18112



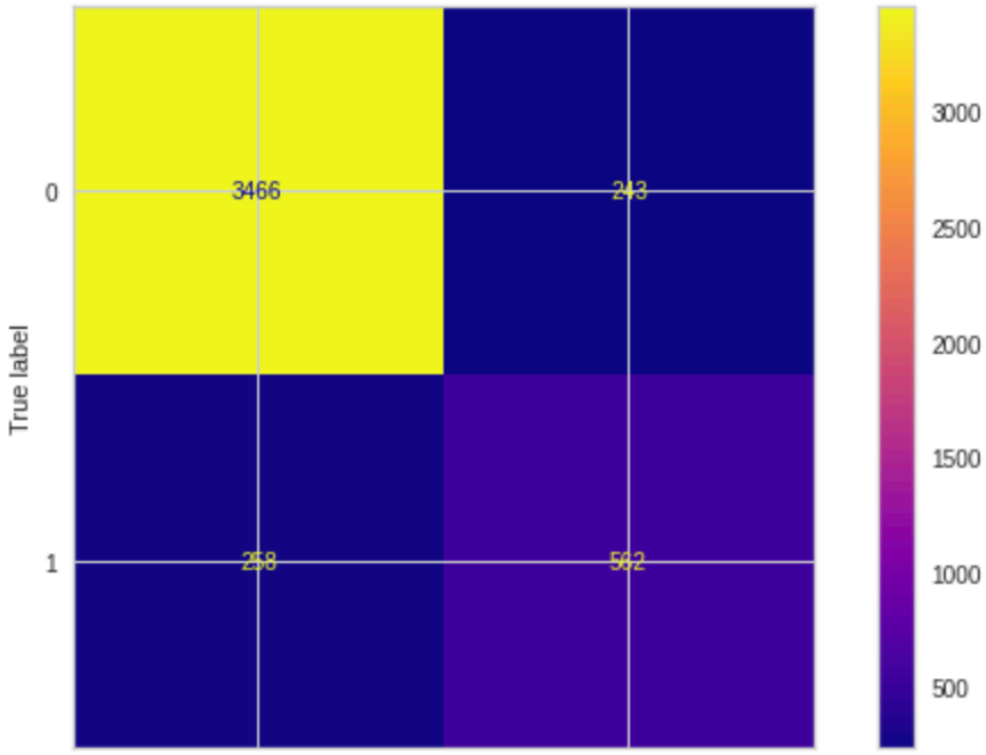
- Naïve Bayes

0s

	precision	recall	f1-score	support
0	0.93	0.93	0.93	3709
1	0.70	0.69	0.69	820
accuracy			0.89	4529
macro avg	0.81	0.81	0.81	4529
weighted avg	0.89	0.89	0.89	4529

Train_Set

	precision	recall	f1-score	support
0	0.95	0.94	0.95	14831
1	0.75	0.78	0.76	3281
accuracy			0.91	18112
macro avg	0.85	0.86	0.85	18112
weighted avg	0.91	0.91	0.91	18112



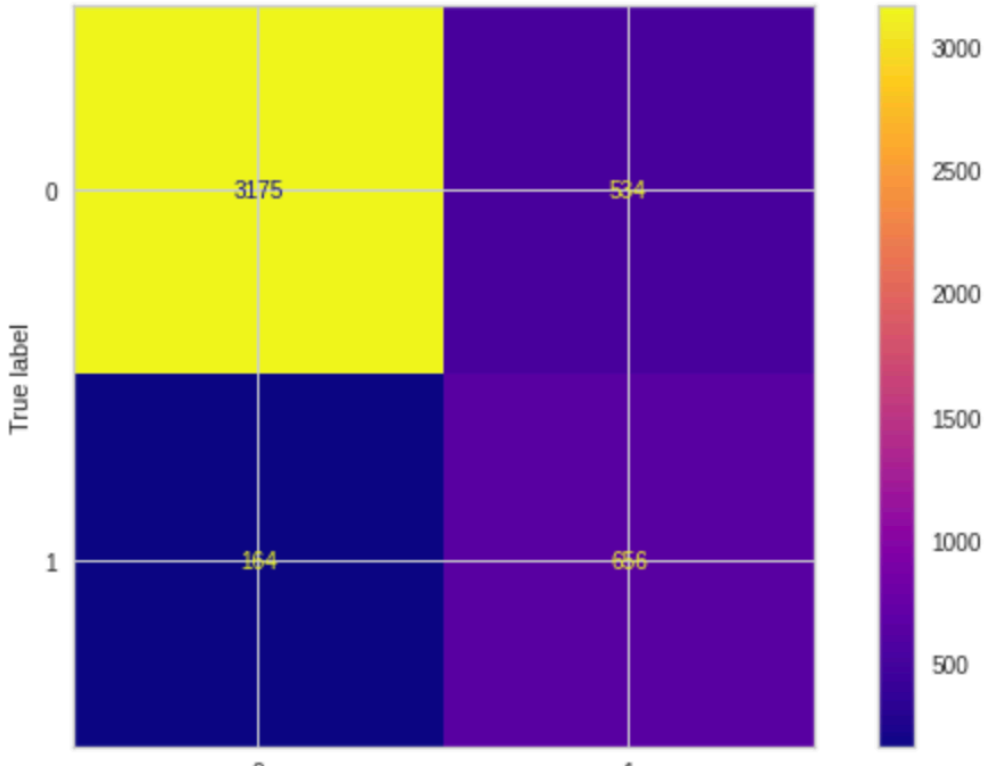
- Random Forest

1s

	precision	recall	f1-score	support
0	0.95	0.86	0.90	3709
1	0.55	0.80	0.65	820
accuracy			0.85	4529
macro avg	0.75	0.83	0.78	4529
weighted avg	0.88	0.85	0.86	4529

Train_Set

	precision	recall	f1-score	support
0	0.97	0.88	0.93	14831
1	0.63	0.89	0.74	3281
accuracy			0.88	18112
macro avg	0.80	0.89	0.83	18112
weighted avg	0.91	0.88	0.89	18112



- Comparison of model scores between all three

```
[262] compare = compare.sort_values(by="Recall_Score", ascending=True)
      compare
```

	Model	F1_Score	Recall_Score	Average_Precision_Score
0	NaiveBayes(Multi)_Count	0.676	0.681	0.728
2	Random Forest_Count	0.655	0.788	0.702
1	LogReg_Count	0.913	0.873	0.732

```
[263] compare = compare.sort_values(by="F1_Score", ascending=True)
      compare
```

	Model	F1_Score	Recall_Score	Average_Precision_Score
2	Random Forest_Count	0.655	0.788	0.702
0	NaiveBayes(Multi)_Count	0.676	0.681	0.728
1	LogReg_Count	0.913	0.873	0.732

```
compare = compare.sort_values(by="Average_Precision_Score", ascending=True)
compare
```

	Model	F1_Score	Recall_Score	Average_Precision_Score
2	Random Forest_Count	0.655	0.788	0.702
0	NaiveBayes(Multi)_Count	0.676	0.681	0.728
1	LogReg_Count	0.913	0.873	0.732

Project Management

Sunny Sumanth Dodda: Took the responsibility of selecting the required classifiers and dataset for the project. Performed the coding operations for the data frame initiation and data cleaning. Selected the required methods need to be implemented for the accurate output execution. Selected the classifiers which will be suitable for the best comparison. Helped in the documentation process of introduction, model, and results part.

Sai Krishna Boinapally: Took the responsibility of data preprocessing and cleaning tasks. Performed all the required operations for the data to be processed with no obstacles for sentimental analysis operations. Performed the coding operations for the data frame initiation and data cleaning. Took responsibility of finding correct algorithms need to be implemented and helped in the code development part.

Jatin Raj Thodupunuri: Took the responsibility of documentation part and screenshots part. Gathered all the necessary screenshots required for the implementation and results part. Helped in the code development and involved in all the required tasks. Performed the logistic classifier operations method, generated graphs, and tables for comparison purpose.

Poojitha Achanta: Performed the naïve bayes and random forest classifier methods and obtained the required tables and graphs for the comparison purpose. Helped in the coding part and documentation report part. Performed the training and testing data operations.

References

Hu, J., & Zhang, B. (2012). Product recommendation system. *CS224W Project Report*.

Jiang, J. (2016). Comparative Analysis of E-commerce Recommendation Strategies. In *2016 2nd International Conference on Education Technology, Management and Humanities Science* (pp. 493-496). Atlantis Press.

Jin, Y., Yang, S. B., Rhee, C., & Lee, K. Y. (2013). An Exploratory Study of the Effects of Price Decreases on Online Product Reviews: Focusing on Amazon's Kindle 2.

Yang, L., Li, Y., Wang, J., & Sherratt, R. S. (2020). Sentiment analysis for E-commerce product reviews in Chinese based on sentiment lexicon and deep learning. *IEEE access*, *8*, 23522-23530.

Zikang, H., Yong, Y., Guofeng, Y., & Xinyu, Z. (2020). Sentiment analysis of agricultural product ecommerce review data based on deep learning. In *2020 International Conference on Internet of Things and Intelligent Applications (ITIA)* (pp. 1-7). IEEE.

P. D. Turney, "Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews", Proceedings of the 40th annual meeting of the association for computational linguistics, pp. 417-424, Dec. 2002, [online] Available: https://arxiv.org/abs/cs/0212032.

P. V. Rajeev and V. S. Rekha, "Recommending products to customers using opinion mining of online product reviews and features", 2015 International Conference on Circuits Power and Computing Technologies (ICCPCT-2015), pp. 1-5, 2015.

"Aamzon review data", Julian McAuley UCSD, 2019, [online] Available: http://jmcauley.ucsd.edu/data/.