SDM COLLEGE OF ENGINEERING TECHNOLOGY,
DHARWAD-580002
MAY 2019
Department of Computer Science and Engineering

SENTIMENT ANALYSIS OF REVIEWS USING MACHINE LEARNING

A MINI PROJECT REPORT

Submitted by

Apoorva R Upadhya: 2SD16CS020
Isha Singh: 2SD16CS043
Madhura Joshi: 2SD16CS051
Shravya K: 2SD16CS087

In partial fulfillment for the award of the degree
Of

BACHELOR OF ENGINEERING
In

COMPUTER SCIENCE AND ENGINEERING

Under the Guidance of
Prof. R N. Yadawad

6th Semester, A Division **Academic Year: 2018-19**

Batch 1

SDM College of Engineering and Technology, Dharwad-580002

Department of Computer Science & Engineering.

CERTIFICATE

Certified that the project work entitled "SENTIMENT ANALYSIS OF REVIEWS USING MACHINE LEARNING" is an original work carried out by Ms. Apoorva R. Upadhya, Ms. Isha Singh, Ms. Madhura Joshi, Ms. Shravya K, in partial fulfillment for the award of degree of Bachelor of Engineering in Computer Science and Engineering of S.D.M College of Engineering and Technology, Dharwad-580002, during the year 2018-19. The project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for Bachelor of Engineering Degree.

Signature of the Guide

Signature of HOD

Name of the Student and USN

: Ms. Apoorva R Upadhya - 2SD16CS020
Ms. Isha Singh - 2SD16CS043
Ms. Madhura Joshi - 2SD16CS051
Ms. Shravya K - 2SD16CS087

Viva-Voce Committee:

Sl. No.	Name	Designation	Signature with Date
1			

ABSTRACT

Our project focuses on sentiment analysis of the costumer's reviews in one of the most trending e-commerce platform which is women's clothing shopping sites using machine learning.

Machine learning concept helps to improve the shopping experience by considering the personal preferences and recommend the consumer while they do a new purchase based on the history providing personalization. Sentiment analysis allows e-commerce platforms to understand the opinions of customer feedback. Along with understanding the emotions of customer feedback it also analyzes the opinions for a particular reason.

The dataset includes attributes like: Clothing ID, Age, Title, Review Text, Rating, Recommended IND, Positive Feedback Count, Division Name, Department Name, and Class Name.

We attempt to understand the correlation of different variables in customer reviews on a women clothing e-commerce, and to classify each review by the depth meaning of the words and these words further helps us to predict whether the reviewed product is recommended or not and whether it consists of positive, negative, or neutral sentiment. To achieve these goals, we employed Multinomial Naive Bayes algorithm.

To understand the dataset we are using data representation techniques like bar graphs which showed the reviews vs. age and category. We also create confusion matrix to check for the efficiency of our classifier.

TABLE OF CONTENTS

LIST OF FIGURES:

LIST OF TABLES

Chapter 1

INTRODUCTION

Online shopping portals use the reviews as a tool for understanding their customers, in order to further improve their products and/or services. Text analysis has become an active field of research in computational linguistics and natural language processing. One of the most popular problems in the mentioned field is text classification, a task which attempts to categorize documents to one or more classes that may be done manually or computationally.

Public opinion plays a vital role in business organization to market the products, venture new opportunities and for sales prediction. This is achieved by searching suitable information from the accumulated data pertaining to the user's history of visiting particular places. A large amount of data can be analyzed and prediction of opinion is possible using a sentiment analysis technique which saves time of customers and business organization.

Consumers rely on online reviews for direct information to make purchase decisions. But, the presence of huge set of reviews for the same product makes it almost impossible to go through and realize the quality of the product. People nowadays pay more attention to their appearances and comforts especially when it concerns clothes, wearing different clothes depending on with whom they will be meeting or avoiding not wearing the same clothes the next day.

This approach is widely known as sentiment analysis that uses statistics and natural language processing techniques to identify and categorize opinions expressed in a text, particularly, to determine the polarity of attitude (positive, negative, or neutral).

1.1 AIM

Our aim is to analyze and classify the reviews present in the dataset based on category of clothes and age group and hence provide the recommendations and rating based on the customer's new review on the cloth.

1.2 OBJECTIVE

- ✔ To focus on e-commerce reviews on women clothing shopping sites, where our aim is to help in summarizing the product reviews.
- ✔ To help the retailers, the e-commerce platforms will use the summary of customer feedback to improve the quality of the products.
- ✔ To help the existing and prospective customers in deciding the products of their interests.

Chapter 2
THEORETICAL BASIS

2.1 PROBLEM DEFINITION

Sentiment analysis for the given e-commerce dataset on women's clothing and predict the probable favourable suggestions of the clothes.

Also provide with correct recommendation IND and rating values based on the review text.

2.2 BACKGROUND WORK

When choosing a particular cloth the customer at present has to go through the reviews presented to them. When selecting through just a few reviews, it can be most effectively done through simply reading the comments. However, if you have thousands and sometimes even hundreds of thousands of reviews on a consistent basis, reading all the feedback can be difficult if not impossible. Currently, Sentiment analysis can be very useful because it can help businesses to differentiate themselves in certain ways and therefore stand out from the crowd of competitors to vie for the attention of customers.

2.3 PROPSED WORK

We analyse the whole dataset and come up with a classifier by using machine learning concepts like Multinomial Naïve Bayes algorithm. We also analyse the whole dataset to support the correctness of our classifier. Thus, sentiment analysis can be of great help in providing directional insight and for paving the way for further analysis, insight and appropriate action.

Chapter 3

SYSTEM ANALYSIS

3.1 HARDWARE REQUIREMENT

- RAM of minimum configuration.
- Hard disk of minimum configuration or higher.
- Any Intel Processor
- Speed 1Ghz or more

3.2 SOFTWARE REQUIREMENT

- Dataset of an e-commerce platform
- Python-V 3
- Virtual python environment
- Jupyter Notebook
- NLTK Library
- Matplotlib Library
- Pandas and numpy libraries.
- Tkinter library

CHAPTER 4

WORK FLOW

4.1 PRE-PROCESSING

Firstly, we remove the unwanted columns in our dataset. Later we send the review text columns for preprocessing.

Now we remove the unnecessary data such as any non-alphanumeric characters, hyperlinks, stop words, etc. from the Review text and Convert all characters to lowercase, in order to achieve consistency.

Finally lemmatize to obtain the pre-processed data.

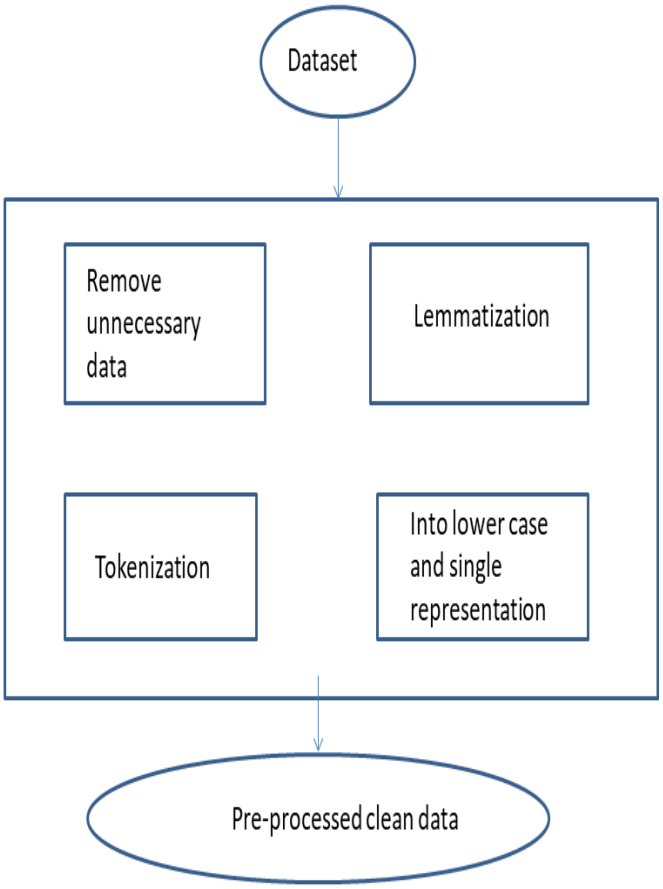


Figure 4.1 Pre-Processing 1

4.2 POLARITY

The pre-processed data is polarized to obtain positive , negetive, neutral emotions of the review.

Now, the analyzer takes the best positive emotions(with value 1) along with user interests to provide with appropriate suggestions.

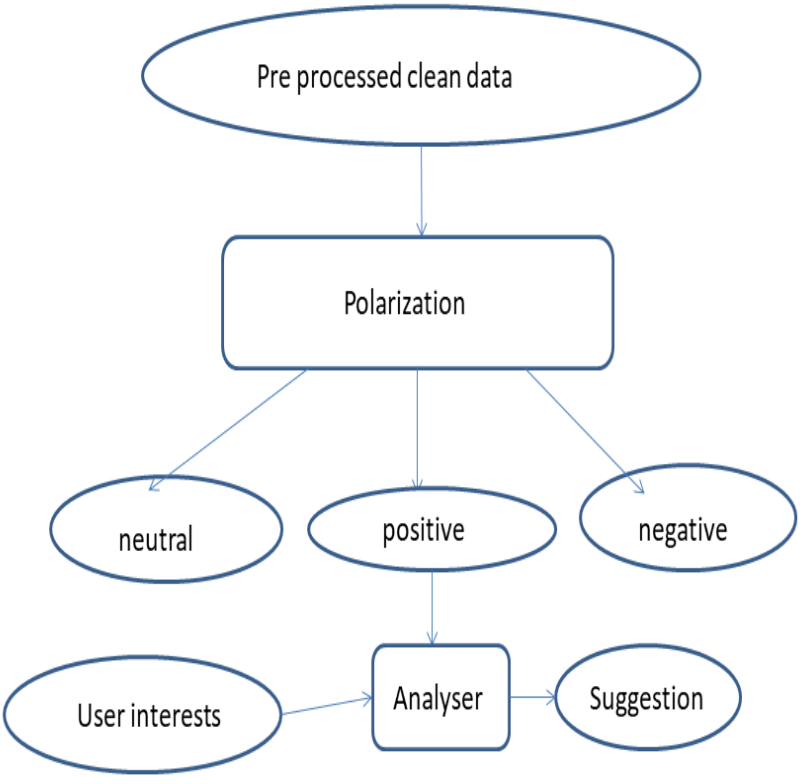


Figure 4.2 Polarizing

4.3 CLASSIFIER MODEL

- In this approach, each element in the vector corresponds to a unique word (*token)* in the corpus vocabulary. Then, if the token at a particular index exists in the document, that element is marked as appropriate weightage, otherwise, its 0. This is essentially bag-of-words.
- These words are then sent into a splitting module where you define a splitting index to split it into Training and Testing data.
- This training data is provided to the Multinomial naïve Bayes algorithm by which we get our classification model. Here, we obtain a trained model (classifier).
- We then send the test data to the classifier which applies the function and provides with the desired result.

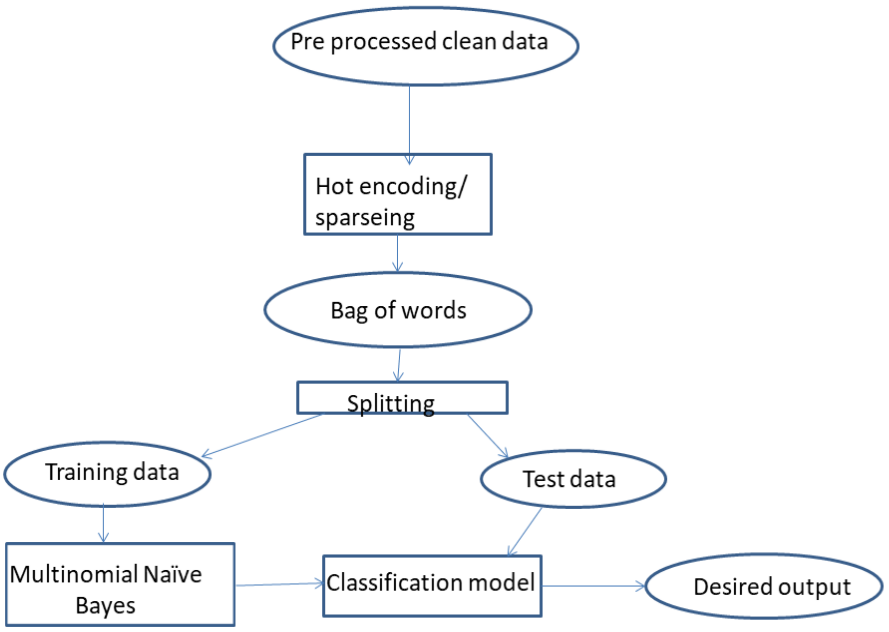


Figure 4.3 Classification Model

Chapter 5

METHODOLOGIES AND IMPLEMENTATIONS

5.1 GATHER THE DATASET

✔ The dataset used is Women's Clothing E-Commerce dataset revolving around the reviews written by customers.

✔ Its nine supportive features offer a great environment to parse out the text through its multiple dimensions.

✔ The attributes includes: independent attributes like clothing ID, Age, Department name, Title.

✔ Dependent attributes like Division name and Class name depends on Department name and Clothing ID. Review, Rating and positive feedback depends on Age and Title.

	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 comf...	4	1	0	Initmates	Intimate	Intimates
1	1	1080	34	NaN	Love this dress! it's sooo pretty. i happene...	5	1	4	General	Dresses	Dresses
2	2	1077	60	Some major design flaws	I had such high hopes for this dress and reall...	3	0	0	General	Dresses	Dresses
3	3	1049	50	My favorite buy!	I love, love, love this jumpsuit. it's fun, fl...	5	1	0	General Petite	Bottoms	Pants
4	4	847	47	Flattering shirt	This shirt is very flattering to all due to th...	5	1	6	General	Tops	Blouses
5	5	1080	49	Not for the very petite	I love tracy reese dresses, but this one is no...	2	0	4	General	Dresses	Dresses
6	6	858	39	Cagrcoal shimmer fun	I aded this in my basket at hte last mintue to...	5	1	1	General Petite	Tops	Knits
7	7	858	39	Shimmer, surprisingly goes with lots	I ordered this in carbon for store pick up, an...	4	1	4	General Petite	Tops	Knits
8	8	1077	24	Flattering	I love this dress. i usually get an xs but it ...	5	1	0	General	Dresses	Dresses
9	9	1077	34	Such a fun dress!	I'm 5"5' and 125 lbs. i ordered the s petite t...	5	1	0	General	Dresses	Dresses

Table 5.1 Dataset

5.2 CLEANING OF DATASET

- We first extract the main attributes from the given dataset which includes Clothing ID, Age, Review Text, Rating, Recommended IND, and Class Name.

	CID	Review Text	Recommend	age	ClassName
0	767	Absolutely wonderful - silky and sexy and comf...	1	33	Intimates
1	1080	Love this dress! it's sooo pretty. i happene...	1	34	Dresses
2	1077	I had such high hopes for this dress and reall...	0	60	Dresses
3	1049	I love, love, love this jumpsuit. it's fun, fl...	1	50	Pants
4	847	This shirt is very flattering to all due to th...	1	47	Blouses

Table 5.2 Necessary Features

- Since, we have to pre-process and clean the text we extract the Review Text attribute column.
- We define a function called remove_noise which consists of the following:
  - Using lower() function to convert the text to lowercase
  - Use stip() function to remove the whitespaces.
  - Use repalce() function along with the **re library** which hs regular expressions to remove repeated numbers, punctuations(Ex. Beautiful!!!! = beautiful(!*))
- Remove stop words like and, are, because, at etc. from review text by comparing it with the stop words list present in stop words library.

Figure 5.1 Stop Words

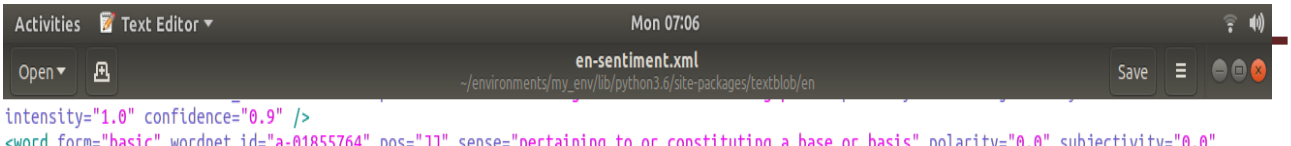
- Tokenize the text by separating it into individual words to the specific tokens (example: word “Beautiful” = adjective = token (JJ)) in order to convert the words to vector form.
- Lemmatize the words in order to get appropriate meaning of the words(example: words such as “studied”, “studies”, and “studying” to simple form of word “study”).

	CID	Review Text	Recommend	age	ClassName	Filtered
0	767	Absolutely wonderful - silky and sexy and comf...	1	33	Intimates	[absolutely, wonderful, silky, sexy, comfortable]
1	1080	Love this dress! it's sooo pretty. i happene...	1	34	Dresses	[love, dress, sooo, pretty, happened, store, i...
2	1077	I had such high hopes for this dress and reall...	0	60	Dresses	[high, hope, dress, really, wanted, work, init...
3	1049	I love, love, love this jumpsuit. it's fun, fl...	1	50	Pants	[love, love, love, jumpsuit, fun, flirty, fabu...
4	847	This shirt is very flattering to all due to th...	1	47	Blouses	[shirt, flattering, adjustable, tie, perfect, ...

Table 5.3 Pre-processed dataset

5.3 COMPUTING POLARITY

- Pre-process data is sent to **textblob library** which has a **sentiment module** which has variable called **polarity**.
- Polarity is a float variable which derives the meaning of the word given by British English and rates the words in the range of -1 to 1 where -1 to 0 being negetive words and 0 to 1 being postive words and 0 being a neutral words.



- As our objective is to provide the customer the clothing IDs which are highly reccomended analysed from the reviews,we consider the clothing ID's which has the highest possible polarity that is 1. Now we send these IDs to the analyser
- Based on the user interests like age group or catogory of clothes, the analyser suggests the clothes.

	CID	Review Text	Recommend	age	ClassName	Filtered	Polarity
0	767	Absolutely wonderful - silky and sexy and comf...	1	33	Intimates	[absolutely, wonderful, silky, sexy, comfortable]	0.633333
1	1080	Love this dress! it's sooo pretty. i happene...	1	34	Dresses	[love, dress, sooo, pretty, happened, store, i...	0.318750
2	1077	I had such high hopes for this dress and reall...	0	60	Dresses	[high, hope, dress, really, wanted, work, init...	0.037400
3	1049	I love, love, love this jumpsuit. it's fun, fl...	1	50	Pants	[love, love, love, jumpsuit, fun, flirty, fabu...	0.500000
4	847	This shirt is very flattering to all due to th...	1	47	Blouses	[shirt, flattering, adjustable, tie, perfect, ...	0.750000

Table 5.4 Polarity of review text

5.4 FINDING A GOOD DATA REPRESENTATION

- We use **CountVectorizer module** of **sklearn library** to achieve the following:
- Build a **vocabulary** of all the unique words in our dataset, and associate a unique index to each word in the vocabulary using **nltk.corpus**.
- Now we compare this vocabulary of words with our filtered dataset and fit the filtered words with the corresponding values based on importance (weightage) present in vocabulary using **fit function**.

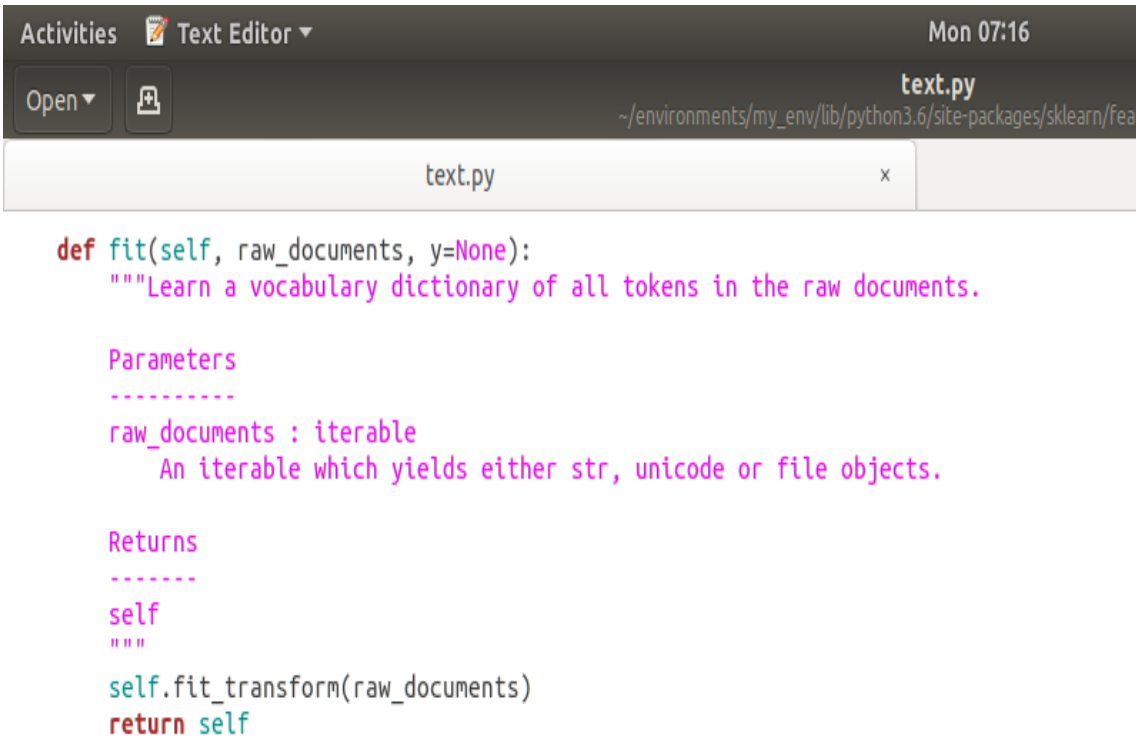


Figure 5.3 fit function

- Next the **transform function** creates a matrix of size 23,486 review text X number of words compute by fit function. This is called **Sparse matrix**.

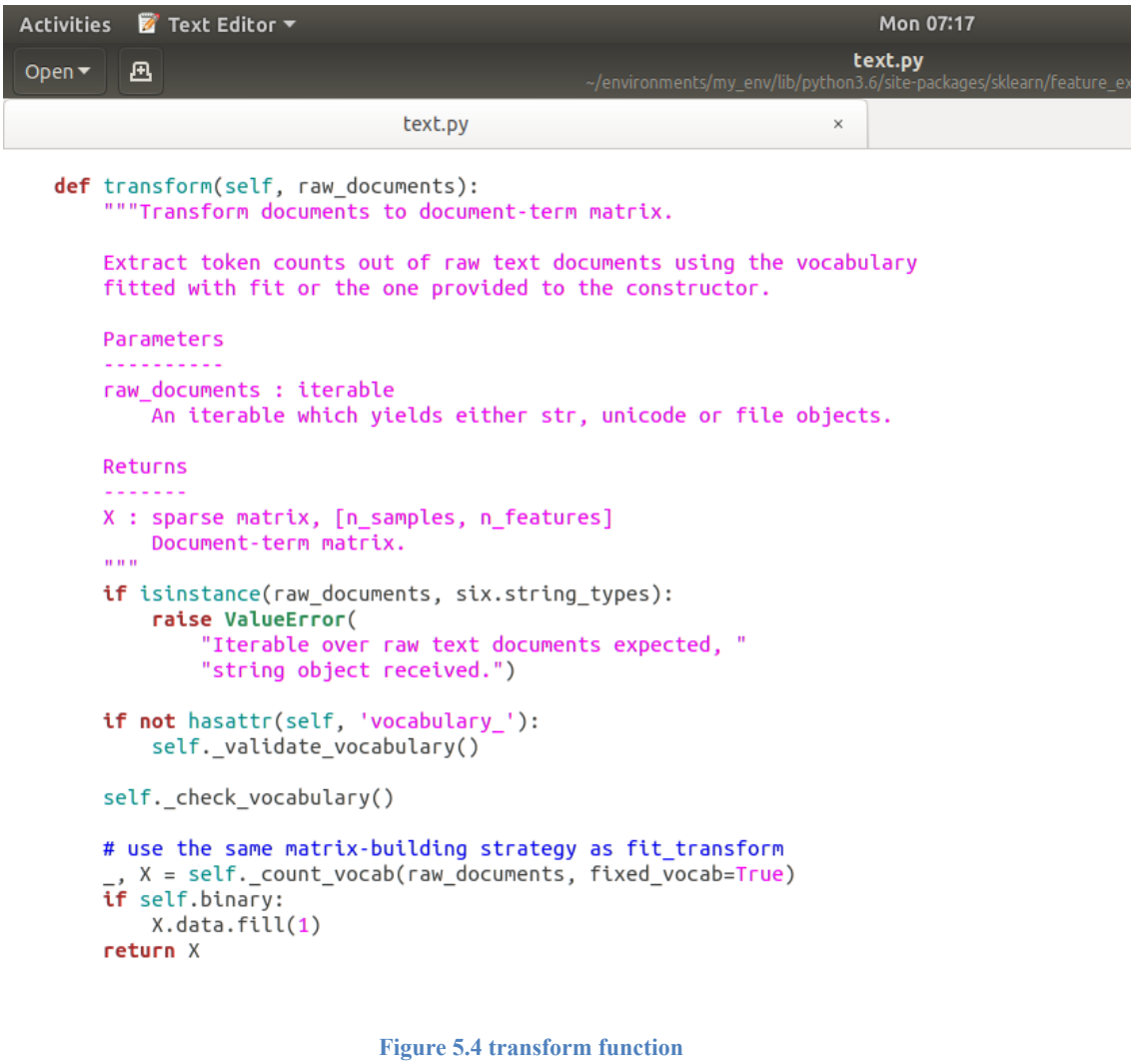


Figure 5.4 transform function

- Now we add the weightage of all the words present in our filtered dataset with its corresponding position in sparse matrix. At each index in this list, we mark how many times the given word appears in our sentence. This process is done using **fit_transform function**.

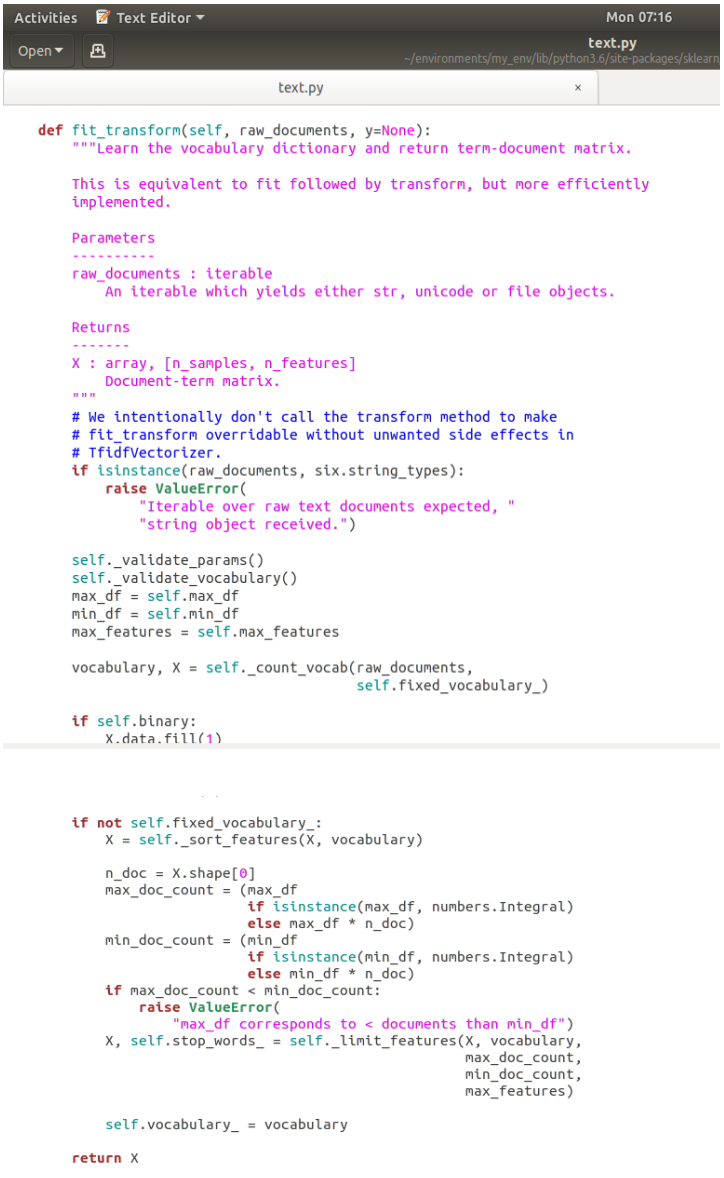


Figure 5.5 fit transform function

- Now we get the final sparse matrix which we use for further processing.

ClassName	Filtered	Polarity	able	absolutely	absolutely love	...	year	yellow	yes	yesterday	youre	zip	zipper	Keyword	Max	Sum
Intimates	[absolutely, wonderful, silky, sexy, comfortable]	0.633333	0.0	0.384864	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.000000	0.000000	silky	0.548067	2.181111
Dresses	[love, dress, sooo, pretty, happened, store, i...	0.318750	0.0	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.000000	0.000000	bc	0.474526	4.486387
Dresses	[high, hope, dress, really, wanted, work, init...	0.037400	0.0	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.178346	0.156682	layer	0.449929	5.224752
Pants	[love, love, love, jumpsuit, fun, flirty, fabu...	0.500000	0.0	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.000000	0.000000	love love	0.647011	2.700876
Blouses	[shirt, flattering, adjustable, tie, perfect, ...	0.750000	0.0	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.000000	0.000000	shirt	0.443253	3.277495

Table 5.5 Sparse matrix

- For visualization purpose we use **get_feature_name function** which finds the word that was occurred maximum number of times and also its weight in the reviews for a particular clothing id.

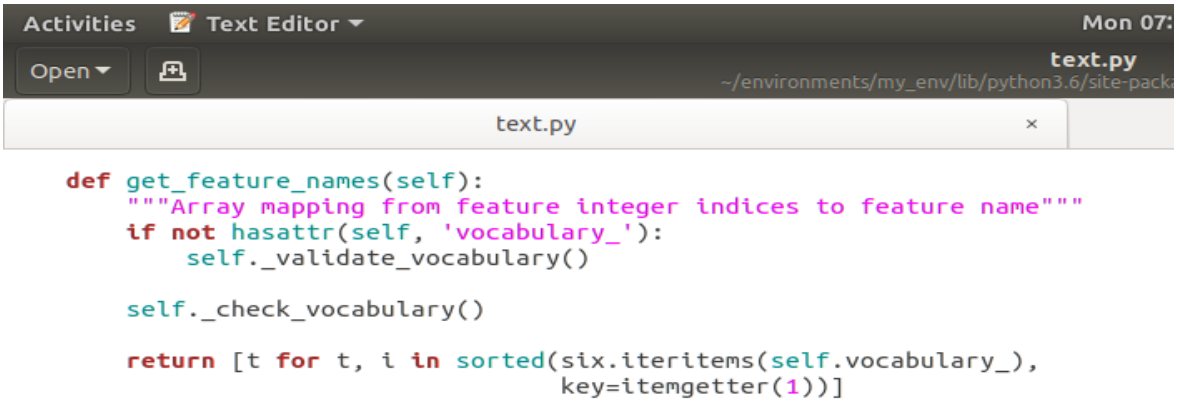


Figure 5.6 get_feature_names function

:

	Term	Occurrences
176	dress	11320
237	fit	10096
668	size	9355
443	love	8968
406	like	7018
110	color	6903
430	look	6873
837	wear	6512
291	great	6076
343	im	5988

	Term	Weight
176	dress	0.051224
443	love	0.041398
237	fit	0.040412
668	size	0.038802
804	unknown	0.035979
291	great	0.033325
110	color	0.033161
430	look	0.032320
406	like	0.031847
837	wear	0.030934

Table 5.6 Occurrence and weight of words

5.5 BUILDING A CLASSIFIER

- By using train_test_split function of model_selection module of sklearn library we split the sparse matrix into training and testing data in the ratio of 80:20 using test_size parameter.
- Now we decide the classification model that we wish to use to train our prediction model.
- We have used Multinomial Naive Bayes because it calculates likelihood to be count of an word/token (random variable):

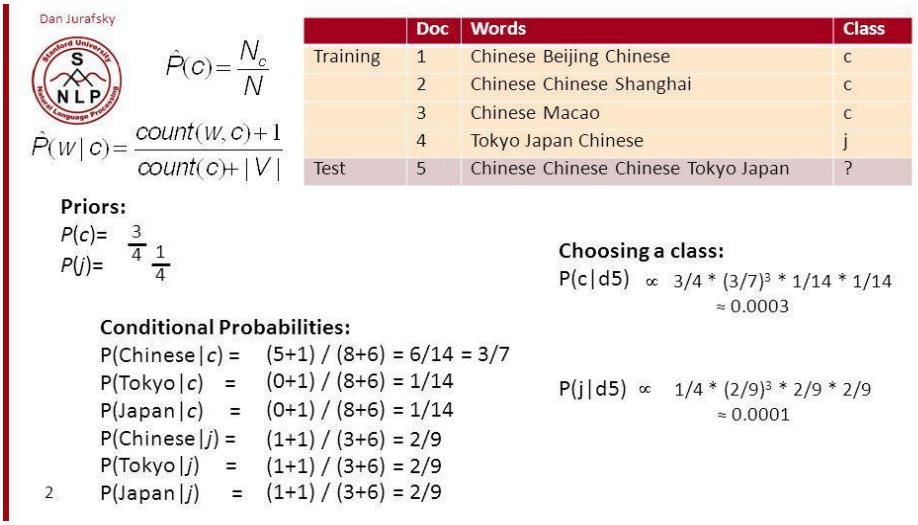


Figure 5.7 MultinomialNB formulae

Where:

- *P(c) indicates the priors of class*
- *P(w|c) is the probability of the word occuring given the class is true.*
- *Count(w,c) is count of the word occuring in that class*
- *Count(c) is count of that word occuring in the training set*
- *V is the toatal vocabulary words in that class*
- *Probabilty of the class given the document , P(c|d5) = Priors * (P(w|c) for each word)*

- To achieve this we use **MultinomialNB() function** from **naïve Bayes module** of **sklearn library.**

```
class MultinomialNB(BaseDiscreteNB):
    """
    Naive Bayes classifier for multinomial models

    The multinomial Naive Bayes classifier is suitable for classification with
    discrete features (e.g., word counts for text classification). The
    multinomial distribution normally requires integer feature counts. However,
    in practice, fractional counts such as tf-idf may also work.

    Read more in the :ref:`User Guide <multinomial_naive_bayes>`.

    Parameters
    ----------
    alpha : float, optional (default=1.0)
        Additive (Laplace/Lidstone) smoothing parameter
        (0 for no smoothing).

    fit_prior : boolean, optional (default=True)
        Whether to learn class prior probabilities or not.
        If false, a uniform prior will be used.

    class_prior : array-like, size (n_classes,), optional (default=None)
        Prior probabilities of the classes. If specified the priors are not
        adjusted according to the data.


    def __init__(self, alpha=1.0, fit_prior=True, class_prior=None):
        self.alpha = alpha
        self.fit_prior = fit_prior
        self.class_prior = class_prior

    def _count(self, X, Y):
        """Count and smooth feature occurrences."""
        if np.any((X.data if issparse(X) else X) < 0):
            raise ValueError("Input X must be non-negative")
        self.feature_count_ += safe_sparse_dot(Y.T, X)
        self.class_count_ += Y.sum(axis=0)

    def _update_feature_log_prob(self, alpha):
        """Apply smoothing to raw counts and recompute log probabilities"""
        smoothed_fc = self.feature_count_ + alpha
        smoothed_cc = smoothed_fc.sum(axis=1)

        self.feature_log_prob_ = (np.log(smoothed_fc) -
                                  np.log(smoothed_cc.reshape(-1, 1)))

    def _joint_log_likelihood(self, X):
        """Calculate the posterior log probability of the samples X"""
        check_is_fitted(self, "classes_")

        X = check_array(X, accept_sparse='csr')
        return (safe_sparse_dot(X, self.feature_log_prob_.T) +
                self.class_log_prior_)
```

Figure 5.8 MultinomialNB code

- Our ultimate goal is to train our model to learn the probabilities needed in order to make a classification decision. We achieve this by training the model by giving it the training set of data.

5.6 Using the trained model for prediction

- Now we use the prediction model that we obtained after training, to compute further predictions. We pass the test data to the **predict function** which gives us the prediction results like recommendation IND or rating.

Chapter 6

RESULT

6.1 DATA VISUALIZATION

We use **seaborn library** and **pyplot module** of **matplotlib library** to visualize our data.

6.1.1 NUMBER OF REVIEWS PER AGE

The histogram is drawn for **age vs. number of reviews**. With its help we realized that maximum reviews were given by people of age 30-40.

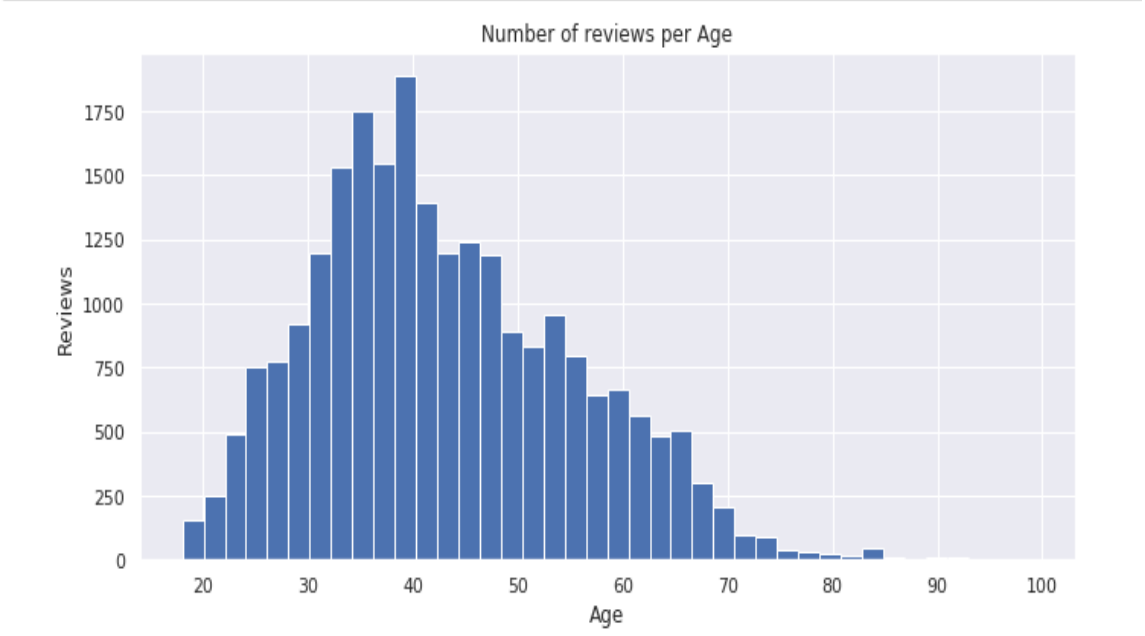


Figure 6.1 Histogram of Review vs. Age

6.1.2 NUMBER OF REVIEWS PER CATEGORY

This bar graph is drawn for **category vs. number of reviews** helped us to realize that **dresses** are most debated category of the shopping site.

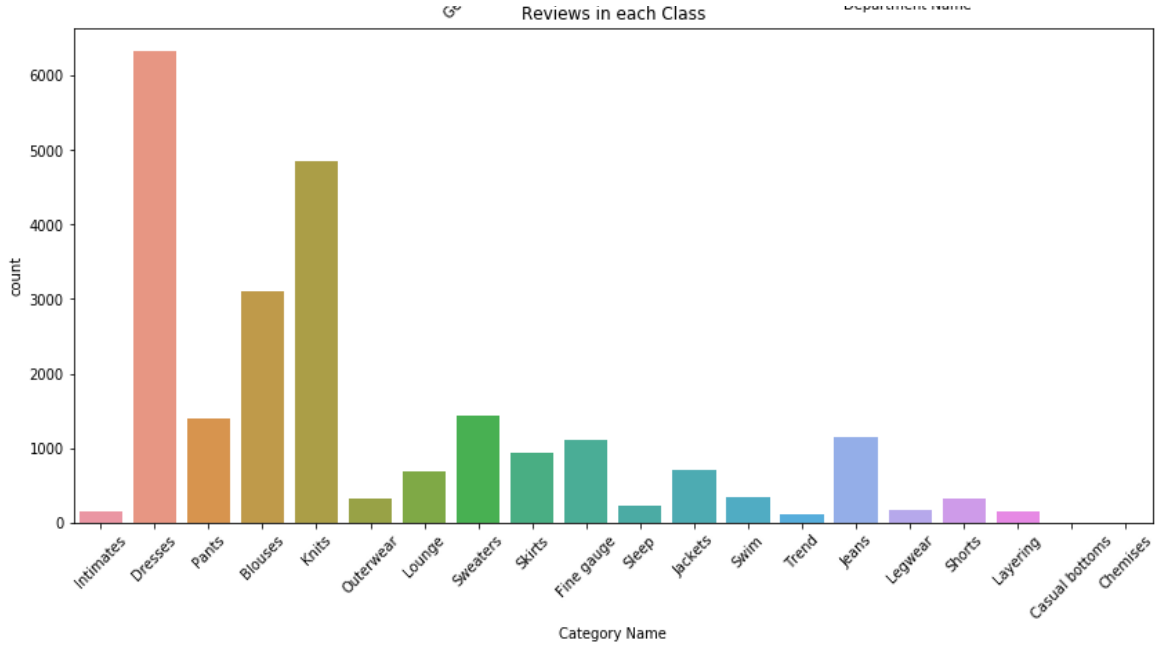


Figure 6.2 Review vs. Category name

6.1.3 TOP 50 POPULAR ITEMS

This bar graph is drawn for **clothing ID vs. popularity** which helped us realize that the clothing ID 1078 has the most reviews and thus, making it the most popular items.

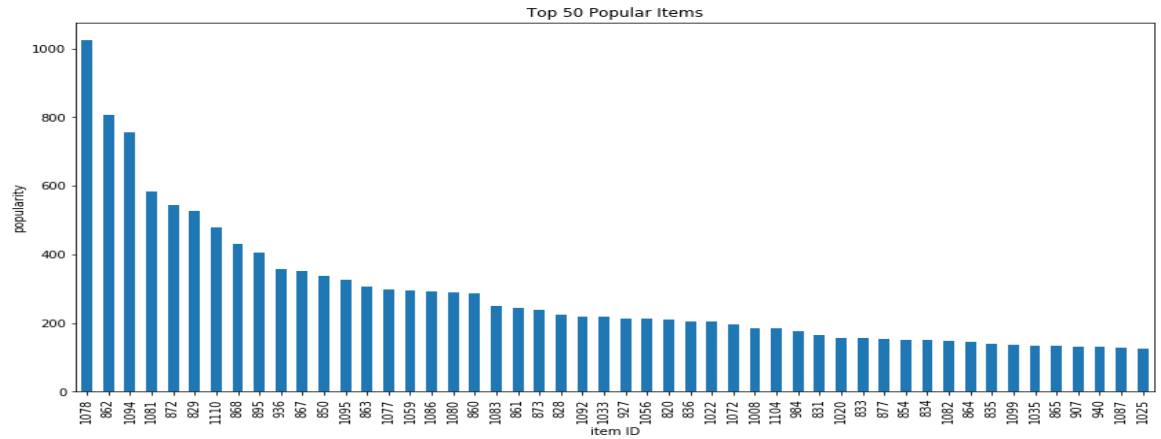
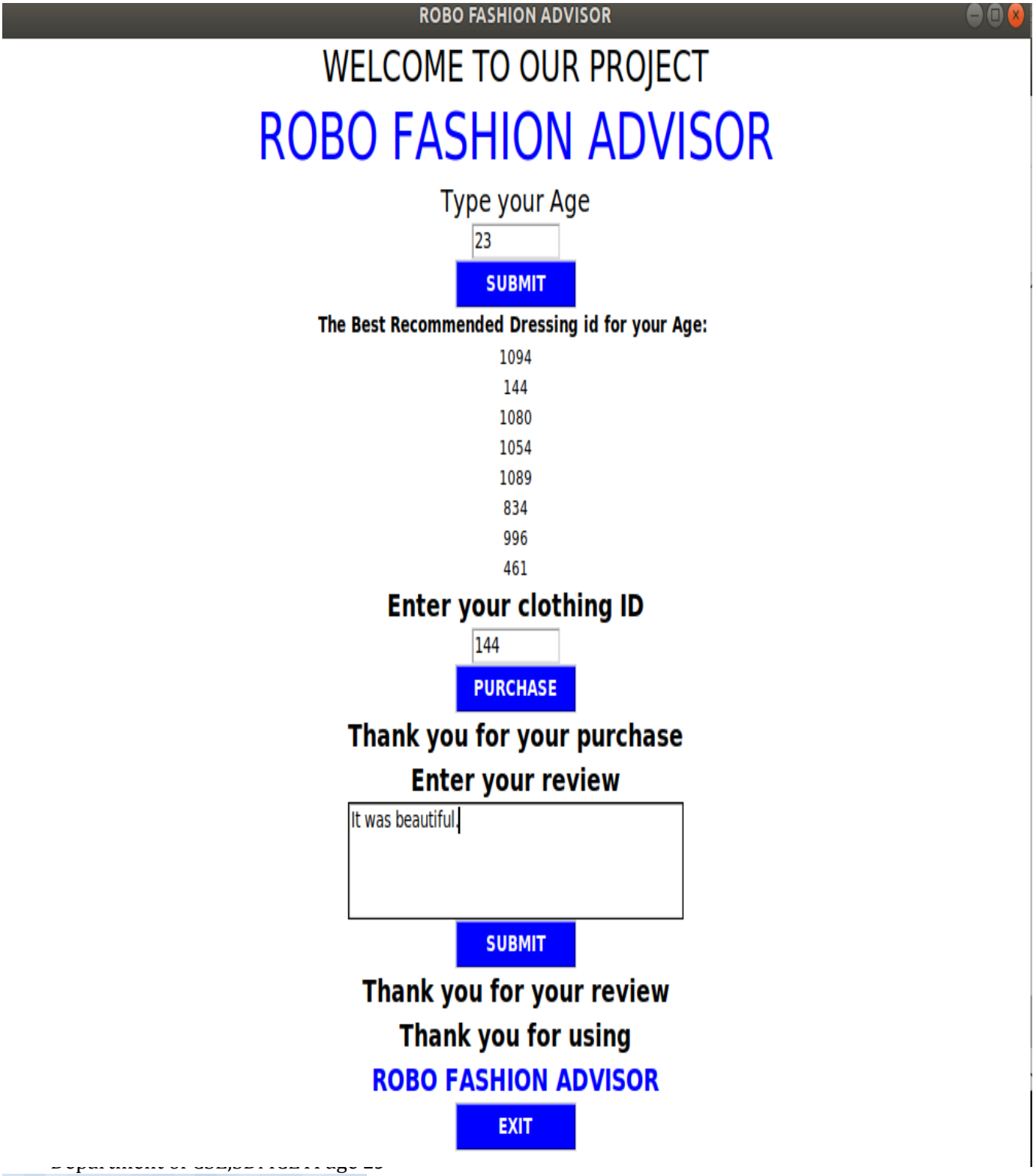


Figure 6.3 Item Id vs. Popularity

6.2 RESULT SNAPSHOTS

6.2.1 RECCOMENDATION BASED ON AGE



6.2.1 RECCOMENDATION BASED ON CATEGORY

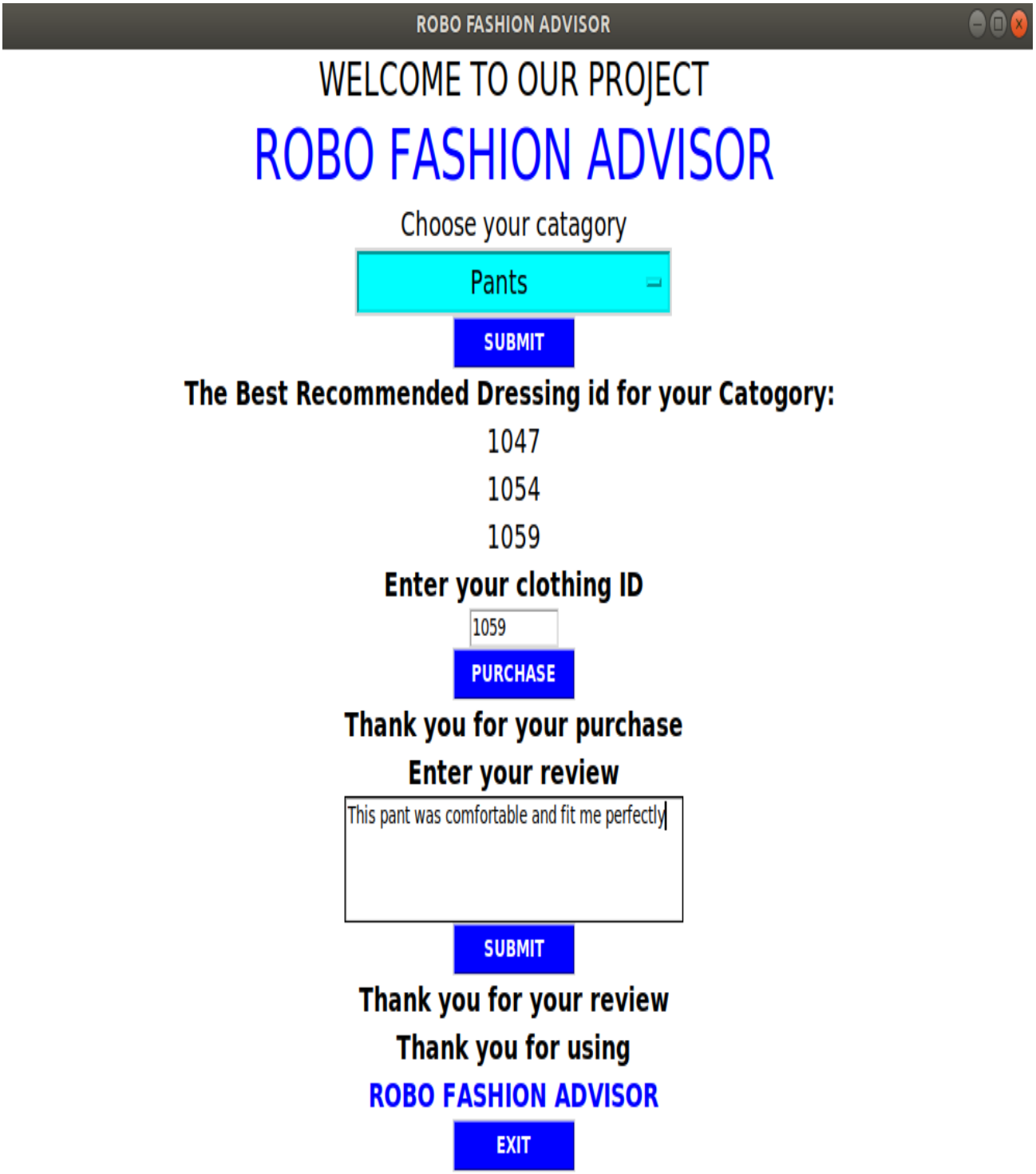


Figure 6.5 Recommendations by Category

6.2.1 PREDICTION OF RECOMMENDED IND

```
In [50]: rating_positive

Out[50]: 'This dress in a lovely platinum is feminine and fits perfectly, easy to wear and comfy, too! highly recommend!'

In [51]: rating_positive_transformed = bow_transformer.transform([rating_positive])
         nb.predict(rating_positive_transformed)[0]

Out[51]: 1

In [52]: rating_negative

Out[52]: "3 tags sewn in, 2 small (about 1'' long) and 1 huge (about 2'' x 3''). very itchy so i cut them out. then the thr
         ead left behind was plasticy and even more itchy! how can you make an intimates item with such itchy tags? not com
         fortable at all! also - i love bralettes and wear them all the time including to work. i am a b cup. however, this
         one is so thin and flimsy that it gives no support even to a b cup - so for me this would only be a lounging brale
         tte - if it wasn't so itchy!"

In [53]: rating_negative_transformed = bow_transformer.transform([rating_negative])
         nb.predict(rating_negative_transformed)[0]

Out[53]: 0

In [54]: rating

Out[54]: 'I bought this item from online... the fit on the model looked a little loose but when i got mine it seemed a bit
         tight! so i took it back to the store & ordered a larger size. for the sale price this is a great top.'

In [55]: rating_transformed = bow_transformer.transform([rating])
         nb.predict(rating_transformed)[0]

Out[55]: 1
```

Figure 6.6 Prediction for recommend IND

6.2.1 PREDICTION OF RATING

```
In [43]: rating_positive=df['Review Text'][23485]
         rating_positive

Out[43]: 'This dress in a lovely platinum is feminine and fits perfectly, easy to wear and comfy, too! highly recommend!'

In [44]: rating_positive_transformed = bow_transformer.transform([rating_positive])
         nb.predict(rating_positive_transformed)[0]

Out[44]: 5

In [45]: rating_negative=df['Review Text'][61]
         rating_negative

Out[45]: "3 tags sewn in, 2 small (about 1'' long) and 1 huge (about 2'' x 3''). very itchy so i cut them out. then the thr
         ead left behind was plasticy and even more itchy! how can you make an intimates item with such itchy tags? not com
         fortable at all! also - i love bralettes and wear them all the time including to work. i am a b cup. however, this
         one is so thin and flimsy that it gives no support even to a b cup - so for me this would only be a lounging brale
         tte - if it wasn't so itchy!"

In [46]: rating_negative_transformed = bow_transformer.transform([rating_negative])
         nb.predict(rating_negative_transformed)[0]

Out[46]: 1

In [47]: rating=df['Review Text'][45]
         rating

Out[47]: 'I bought this item from online... the fit on the model looked a little loose but when i got mine it seemed a bit
         tight! so i took it back to the store & ordered a larger size. for the sale price this is a great top.'

Out[48]: 5
```

Figure 6.7 Prediction for rating

6.3 CONFUSION MATRIX AND PERFORMANCE MEASUREMENT

6.3.1 CONFUSION MATRIX OF RECCOMANDATION IND

```
[[ 588  448]
 [ 248 4615]]
```

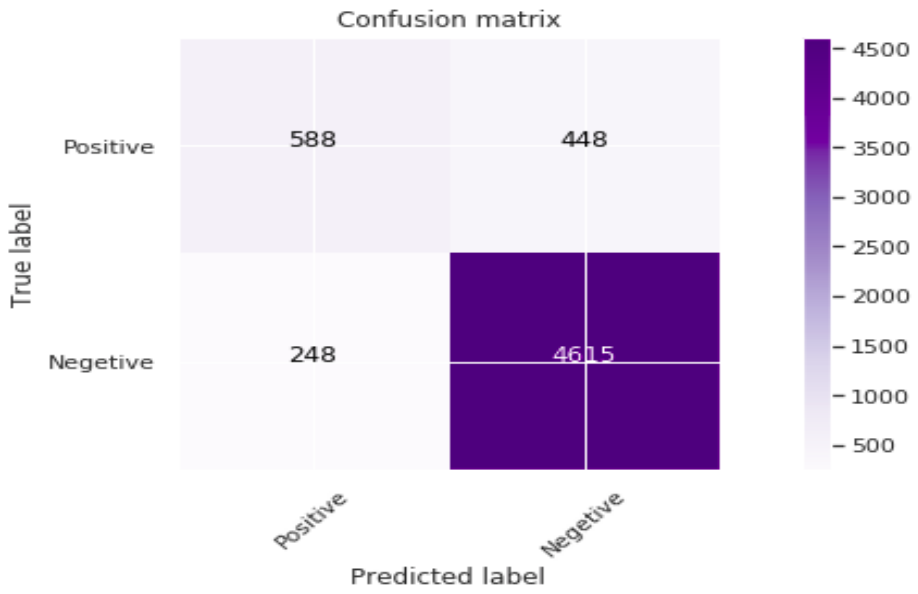


Figure 6.8 Confusion matrix for recommend ind

6.3.2 CONFUSION MATRIX OF RATINGS

```
[[   65  134]
 [   14 3252]]
```

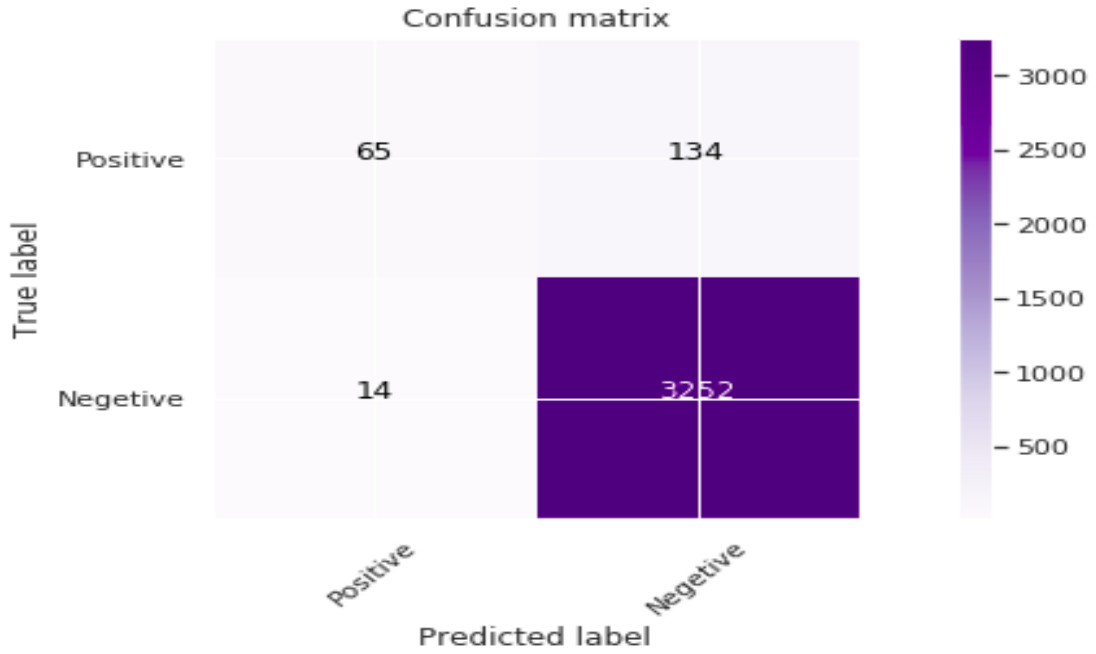


Figure 6.9 Confusion matrix for rating

6.3.2 PERFORMANCE MEASUREMENT OF RECCOMENDED IND

	precision	recall	f1-score	support
0	0.70	0.57	0.63	1036
1	0.91	0.95	0.93	4863
micro avg	0.88	0.88	0.88	5899
macro avg	0.81	0.76	0.78	5899
weighted avg	0.87	0.88	0.88	5899

Figure 6.10 Performance measurements for recommended IND

6.3.2 PERFORMANCE MEASUREMENT OF RATING

	precision	recall	f1-score	support
1	0.82	0.33	0.47	199
5	0.96	1.00	0.98	3266
micro avg	0.96	0.96	0.96	3465
macro avg	0.89	0.66	0.72	3465
weighted avg	0.95	0.96	0.95	3465

Figure 6.11 Performance measurements for rating

Chapter 7
CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

Through this project we were able to explore the vast libraries and modules present in NLP and also understand and implement few of machine learning algorithms. Sentiment analysis helped us to provide the users with best recommended clothing ID. Also, Multinomial Naïve Bayes algorithm helped us to provide the retailers an easy and efficient way to know whether the users have genuine interests in their products by providing them with true ratings and recommended IND. Thus, we were able to achieve our goals of improving user experiences and retailers service.

7.2 FUTURE SCOPE

- Differentiating between fake and honest reviews.
- Provision for removal of redundant and old reviews.
- Provision for adding of new reviews, updating the dataset.
- Increasing the capacity of training set and hence increasing the efficiency of the prediction model for future data.

REFERENCES

1) Abien Fred M. Agarap, Department of Computer Science Adamson University Manila, Philippines abien.fred.agarap@adamson.edu.ph ,Paul M. Grafilon, Ph.D.† Department of Computer Science Adamson University Manila, Philippines grafilonpaul@yahoo.com, Statistical Analysis on E-Commerce Reviews, with Sentiment Classification using Bidirectional Recurrent Neural Network
2) Geoff Hulten, Apress Media LLC publishers, Building Intelligent Systems, A Guide to Machine Learning Engineering
3) Minqing Hu and Bing Liu ,Mining and Summarizing Customer Reviews , Department of Computer Science University of Illinois at Chicago 851 South Morgan Street Chicago, IL 60607-7053 {mhu1, liub}@cs.uic.edu
4) Paul Barry (2nd Edition), Head First Python, O'Reilly publications.
5) Sasikala P*1 , L.Mary Immaculate Sheela#2 *Research Scholar, Department of Computer science, Mother Teresa Women's University, Kodaikanal, India, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 14 (2018) pp. 11525-11531 © Research India Publications. http://www.ripublication.com, Sentiment Analysis and Prediction of Online Reviews with Empty Ratings
6) Vishal A. Kharde, S S Sonawane, (April 2016), Sentiment Analysis of Twitter Data: A Survey of Techniques, International Journal of Computer Applications (0975 – 8887) Volume 139 – No.11.
7) https://www.digitalocean.com/community/tutorials/how-to-set-up-jupyter-notebook-for-python-3
8) https://www.digitalocean.com/community/tutorials/how-to-work-with-language-data-in-python-3-using-the-natural-language-toolkit-nltk
9) https://pythonprogramming.net/tokenizing-words-sentences-nltk-tutorial/
10) https://www.youtube.com/watch?v=3Pzni2yfGUQ&feature=youtu.be
11) https://www.kaggle.com/