E-commerce reviews analysis

September 9, 2018

1 Introduction

A Women's Clothing E-commerce dataset consists in reviews written by customers. This is an interesting dataset to analyze because we can match customers and interests with products; it involves query understanding but also scoring and ranking of results, and everything else we need to do to deliver the right results to the right customers. Target the right people and make predictions around behavior and preferences. Understanding the consumer's preferences can help also the managers to make decisions and improve their offer.

In marketing strategies, the companies need to understand their clients and the customer sentiments in order to improve their services and also their products. Through my analysis the companies can identify and categorize opinions towards a product. The sales can be more accurately and increase e-commerce sales by displaying a larger assortment of similar products to each customer. Determine the potential buyers for a certain group of products. My client are all the e-commerce company, from the smallest to the biggest ones.

The data will be acquired from Kaggle (https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews) and will include 23486 rows and 10 feature variables, which:

- Clothing ID: unique identifier of the product;
- Age: age of the reviewer;
- Title: title of the review;
- Review text: product review;
- Rating: product score granted by the customer from 1 Worst, to 5 Best;
- Recommended IND: customer recommends the product where 1 is recommended, 0 is not recommended;
- Positive feedback count: number of positive feedback on the review;
- Division name: name of the division product is in;
- Department name: name of the department product is in;
- Class name: type of product.

This report is organized as follows: In Sect. 2 I explain the procedure to inspect and clean the data. In Sect. 3 I present the Exploratory Data Analysis (EDA) carried out and I describe the Natural Language Processin (NLP) and the work done with the text. In Sect. 4 I show the correlations between different features and in Sect. 5 I explain the machine learning algorithms that I use to predict, in particular, the rating. Finally, in Sects. 6 I summarize and discuss the steps that can be done in the future.

2 Data Wrangling

The original dataset is written into a file called 'Womens Clothing E-Commerce Reviews.csv', which can be used to make the data analysis. My dataset is composed not only from numerical values but also from text data. From this task, I manage my dataset using Pandas.

My first step is to import relevant libraries and them load the data file into a DataFrame (*rev*). After that, I inspect it using the **.head()** method. But to can see everything, I need to look at the data in another way. The **.shape** and **.columns** attributes let me see the shape of my DataFrame and obtain a list of its columns.

Then, I continue to diagnose my data with the useful **.info()** method. The **.info()** method provides important information about a DataFrame, such as the number of rows, number of columns, number of non-missing values in each column, and the data type stored in each column.

From the results, I am be able to see whether or not all columns have complete data in them; and here I can see that I have some columns with missed values (Title, Review Text, Division Name, Department Name and Class Name). In particular, I can know exactly the number of NaN values per columns using **.isnull().sum()** method.

After this check, I decided to remove missing data only during the Natural Language Processing (NLP) but not for the Exploratory Data Analysis (EDA). This decision is taken to not loose important features during the EDA.

Now, I use **.describe()** method to explore statistically the data and I discovered that the maximum age is 99; so I decided to print the rows where the age of the shoppers is 99: it seems quite abnormal and it could be due to a bug.

As mentioned before, in my original dataset I have text data and in that data I have NaN values; for this reason, I need to filter out data removing the NaN values when I start to analyse it (NLP). So I rename my dataframe as *rev_new* removing the row with NaN values.

The script that makes the data wrangling and cleaning (when necessary) can be found in this link:

https://github.com/LisaB83/Analysis-on-e-commerce-reviews/blob/LisaB83-patch-1/CP_Data_Wrangling.ipynb

3 Exploratory Data Analysis (EDA) and Natural Language Processing (NLP)

3.1 EDA

I start conducting an observational study of the dataset to understand and characterize the features. The code for the EDA part includes the basic data manipulation tools for scientific computation (numpy) and dataframes (pandas) and the creation of simple graphics in the visualization section using matplotlib and seaborn.

First, I analyse the dataset by studying the age of the reviewers: I plot the distribution of their age (Fig 1). The age spreads out over 80 years (from 18 to 99) with the major of the reviewers around 30-50 years and the median age around 40.

After that, I check the products and in particular if the products are recommended or not and how the ratings are distributed, also depending of recommended IND feature.

Fig 2, 3 and 4 show the results: 19314 cloths are recommended, against 4172 not recommended and the major part of products have a good rating, as expected from the previous result; the reviews are very highly positive with a score of five and four. It means that the people writing the

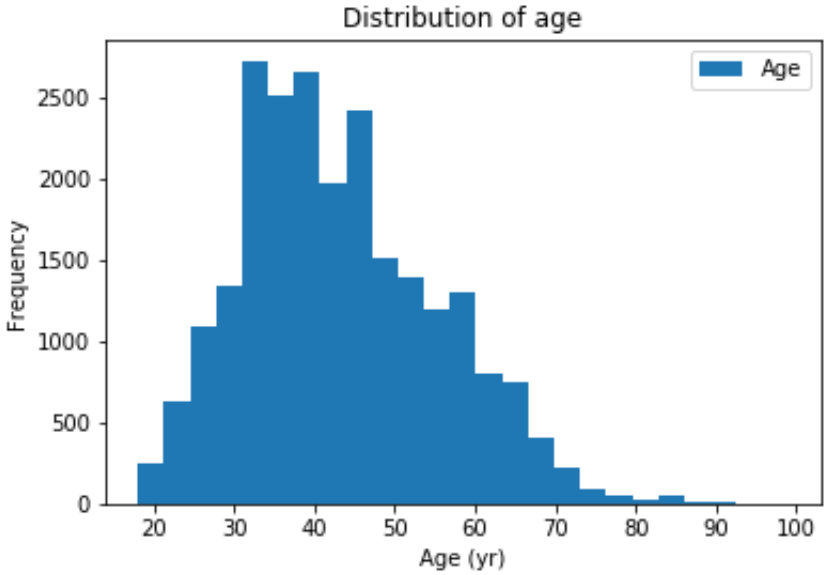

Fig. 1 Histogram to see the distribution of the age of the reviewers

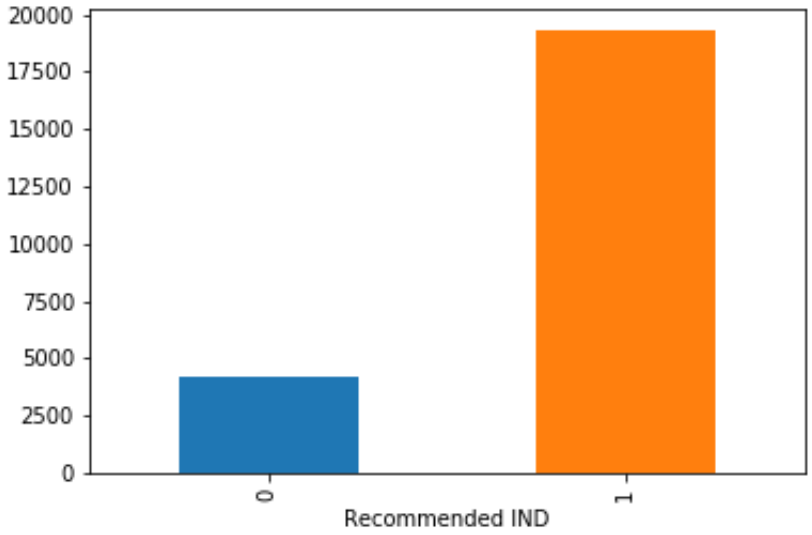

Fig. 2 Distribution of Recommended IND

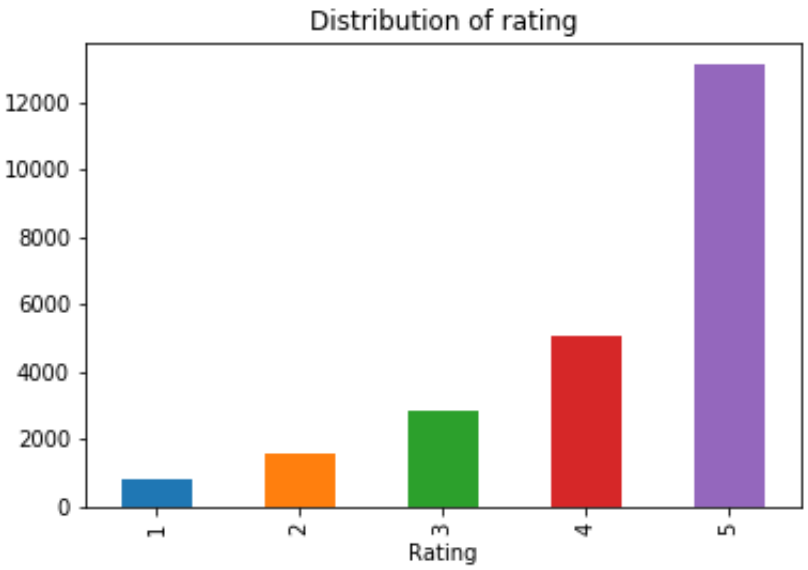
Distribution of rating
12000
10000
8000
6000
4000
2000
0
1
2
3
4
5
Rating

Fig. 3

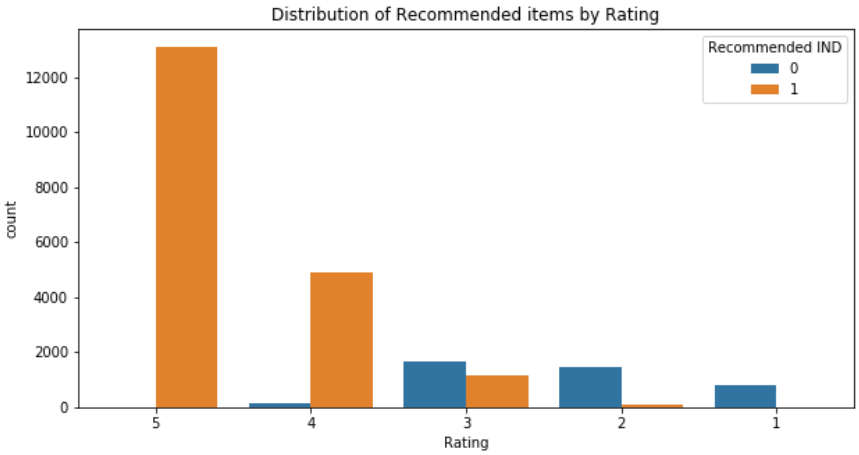
Distribution of Recommended items by Rating
Recommended IND
0
1
count
12000
10000
8000
6000
4000
2000
0
5
4
3
2
1
Rating

Fig. 4

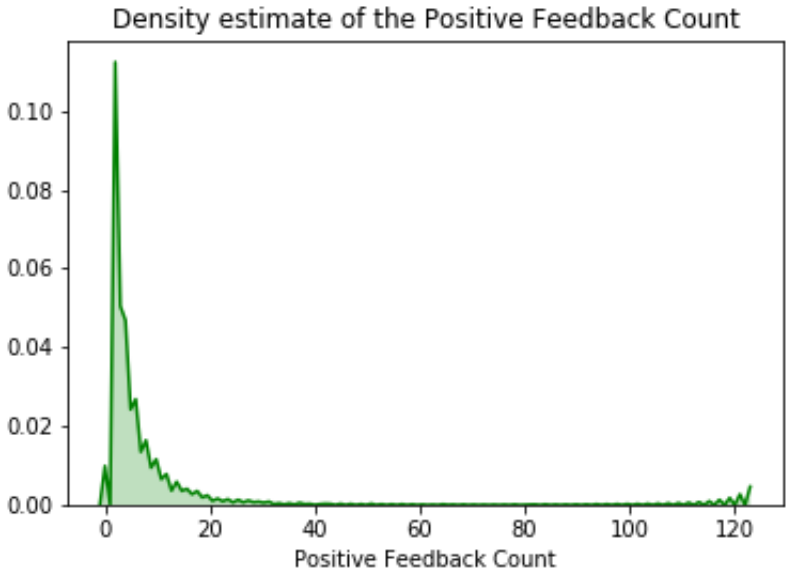

Fig. 5

reviews express good opinions on the product that they bought. But when I plot the ratings for the recommended and not recommended items (Fig 4), we note that there are low rating (1 and 2) also for recommended products and high ratings (4 and 5) for not recommended items. This result could be explained by the very personal clothing experience: for example, in the case of recommended items with low rating it seems that the customers love the product but they have fit issue or product handling problem.

Then, I consider the density of number of positive feedback (Fig. 5) showing a result close to 0. It means that there are few customers that find the reviews very positive: the most of items don't have positive feedback (0) or very few one (only 1,2 or 3).

In Fig. 6, I can see the 10 most recommended and not recommended cloths: I can observe that the first 3 Clothing ID are the same (recommended and not): 1078, 862 and 1094.

Now, I consider the features 'Division Name', 'Department Name' and 'Class Name':

- first, I plot the distribution of recommended and not products by these features and the trend is confirmed;
- second, I calculate the percentage and plot the distribution for the categories of these features and I can conclude that the General Category (59.01 %) has most reviews; Tops (44.6 %) and Dresses (26.92 %) are the most commonly reviewed products and the most popular clothing types are dresses (26.92 %), knits (20.63 %) and blouses (13.19 %).

Then, I study the distribution of rating divided by Department Name and Division Name. The result is consistent with the entire distribution of Rating, so, as expected, the ratings 5 and 4 have always more reviews.

To finish, I decided to divide the age in 3 subgroups by the age:

1. Age < 34;
2. 34 <= Age <= 52 and
3. Age > 52

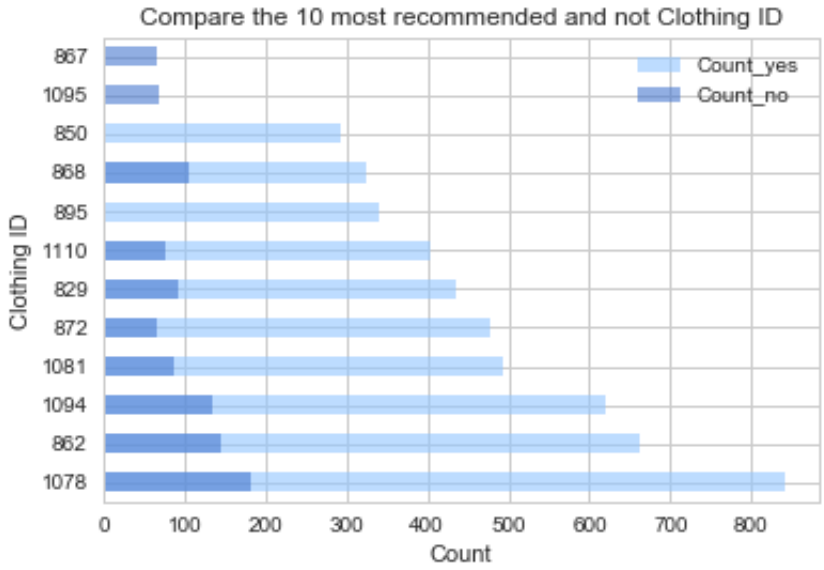

Fig. 6

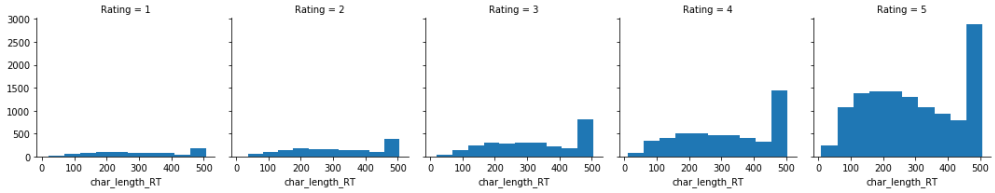

Fig. 7

and see the number of reviews per Department Name, Division Name and Class Name. In this case I would like to see if the age influences the distribution of reviews: from the results it seems that each subgroups presents the same most commonly reviewed products with the conclusion that the age doesn't influence the reviews distribution.

3.2 Working with Text.

My dataset contains also text features. This category includes, in addition to numpy, pandas, matplotlib and seaborn, also Natural Language Processing (NLTK), TextBlob and Scikit-learn (sklearn) libraries.

In this case I use the new dataframe created removing *NaN* values (*rev_new*) and in particular, starting to consider the *Review Text* feature. First, I calculate the number of characters in each review text: from the result I can gather that the text probably has limit of around 500 characters length (Fig. 7).

Then, I extract the number of words: the Table 1 shows the summary of statistical description of the character and word length for the Review Text.

Table 1

	char_length_RT	word_count_RT
count	22641	22641
mean	308.69	60.22
std	143.94	28.50
min	9	2
25%	186	36
50%	301	59
75%	459	88
max	508	115

I did the same process also for *Title* feature and the Table 2 shows its summary statistics.

Table 2

	char_length_Title	word_count_Title
count	19676	19676
mean	19.10	3.35
std	9.58	1.80
min	2	1
25%	12	2
50%	17	3
75%	24	4
max	52	12

I calculate also the number of exclamation points ('!') by rating and, as expected, I find more exclamation points for high rating and in particular for rating 5; it means that the exclamation points express enthusiasm.

After that, I do some basic pre-processing steps for *Review Text* to study the most frequent words as remove stopwords, punctuations and keep only alphabetic words. I tokenize the data. Tokenizing means splitting our text into minimal meaningful units (tokens). These tokens could be paragraphs, sentences, or individual words.

I start separating the dataset into 5 groups based on their rating and extracting the 20 most frequent words for each group. I can notice that the two words more frequent for Rating 1, 2 and 3 are the same : 'dress' and 'like'; while for Rating 4 the first is 'dress' but the second 'size'. For Rating 5 the first two are 'love' and 'dress'.

Instead, in Fig 8, 9 I divide the dataset in only 2 groups, the highly (>= 3) rating and the low rating (<3), and check the words with highly and low rated comments: as expected from the previous analysis, the most frequent words for high rating are 'love' and 'dress', instead for the low rating are 'dress' and 'like'.

In Fig 10, 11 I do the same with the words for *Title* feature.

Without divided by rating the most common words used for *Review Text* are shown in Table 3.

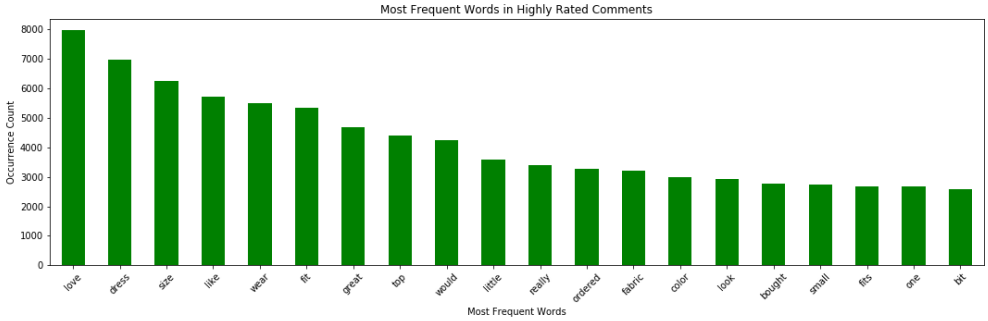

Fig. 8

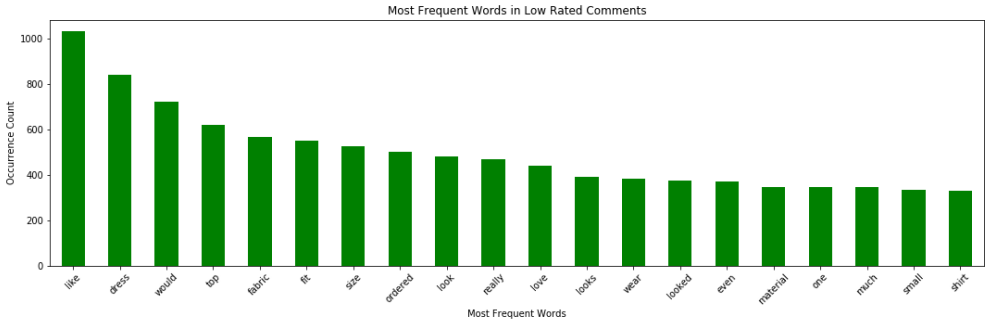

Fig. 9

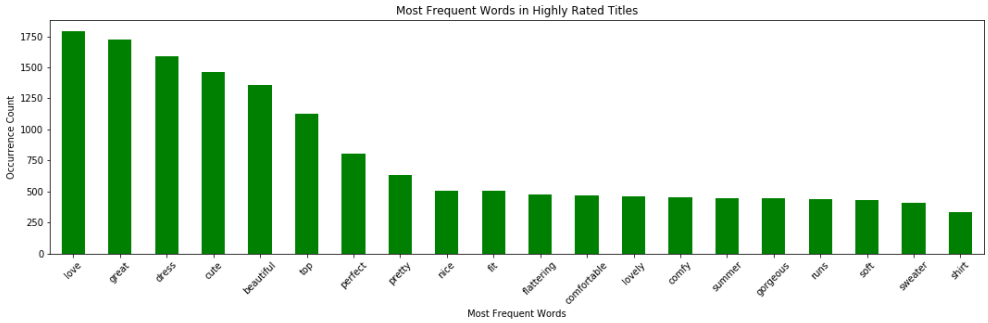

Fig. 10

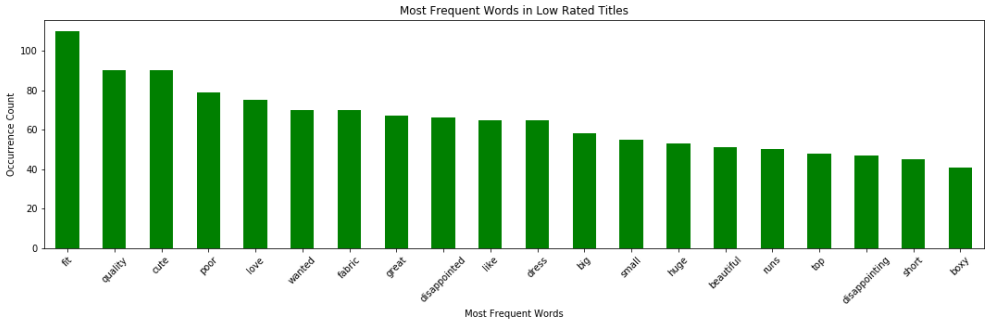

Fig. 11

Table 3

word	frequency
dress	10460
love	8916
size	8687
fit	7238
like	6992
wear	6414
great	6076
im	5974
fabric	4755
color	4557

This result can be explained because the reviews with high rating are more than the other ones.

I use, also, the **TF-IDF** (Term Frequency and Inverse Document Frequency).

The Term Frequency (tf) measures how frequently a word occurs in a document; however, there are words in a document that occur many times but may not be important. An approach is to look at a term's inverse document frequency (idf), which decreases the weight for commonly used words and increases the weight for words that are not used very much in a collection of documents. This can be combined with term frequency to calculate a term's tf-idf, the frequency of a term adjusted for how rarely it is used. It is intended to measure how important a word is to a document in a collection (or corpus) of documents computing a weight. The importance increases proportionally to the number of times a word appears in the document.

A TfidfVectorizer can be accessed through the attribute idf_, which will return an array of length equal to the feature dimension. I sort the features by this weighting to get the top weighted features and I determine the 20 words with highest TF-IDF scores for each rating group.

I note that the first 2 words more frequent are the same as in the previous analysis for every rating.

Instead, the first 10 words in general for *Review Text* based on Tf-Idf score are:

- love,
- size,
- fit,
- dress,
- like,
- wear,
- great,
- im,
- just,
- fabric

While for *Title*, they are:

- great,
- love,
- dress,
- cute,

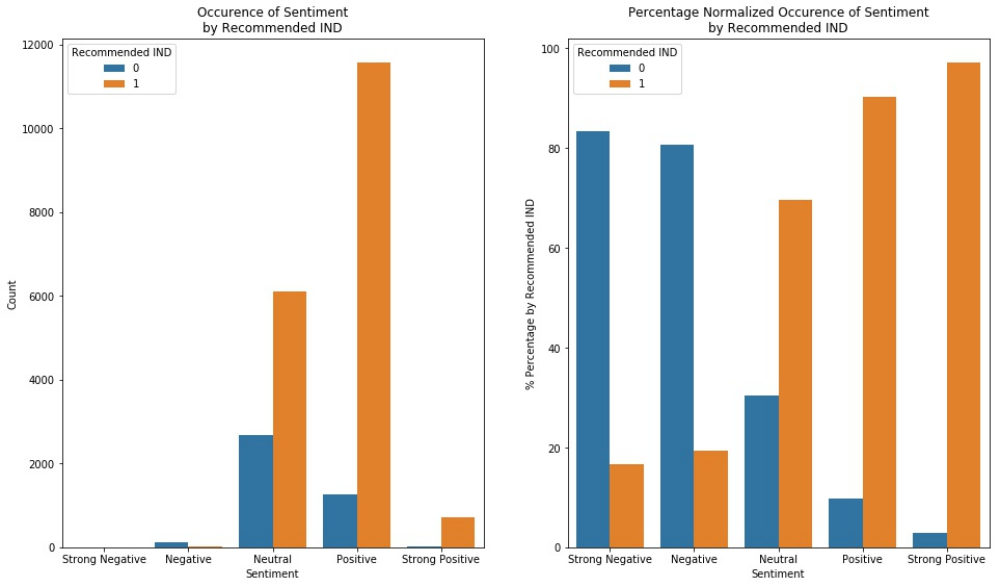

Fig. 12

- beautiful,
- perfect,
- pretty,
- fit,
- nice,
- flattering

After checking the words more used, I move to detect the sentiment of the reviewers. First, I extract polarity: it indicates the sentiment as value nearer to 1 means a positive sentiment and values nearer to -1 means a negative sentiment. This can also work as a feature for statistics study and for building a machine learning model. When I have the sentiment feature, I can plot it by Recommended IND (Fig. 12), by Rating (Fig. 13) and by Division Name considering before only the recommended cloths and after the no-recommended items (Fig. 14 and Fig. 15).

Checking these histograms I can conclude that:

- the recommended cloths have more positive sentiment in the review, while the no-recommended items more negative one, as expected;
- the distribution for the neutral, positive and strong positive sentiment increasing when the rating number increases. Instead, the negative sentiment, as expected, has more low rating occurence (1 and 2). For the strong negative sentiment, the distribution is particular: has a peak for rating 1 and 2, as expected, but also more or less 15 % of occurence for rating 5.

About the sentiment feature by Division name, the 'strong negative' sentiment presents a particular feature: for the recommended reviews only the 'General Petite' is shown, while for the non-recommended only the 'General' Division Name is shown. Instead, the distribution of division names for the other sentiments does not seem to change depending on status of recommendation.

If the reader is interested in more detail about the scripts that we used to make the exploratory data analysis and natural language processing, you can go to the following links:

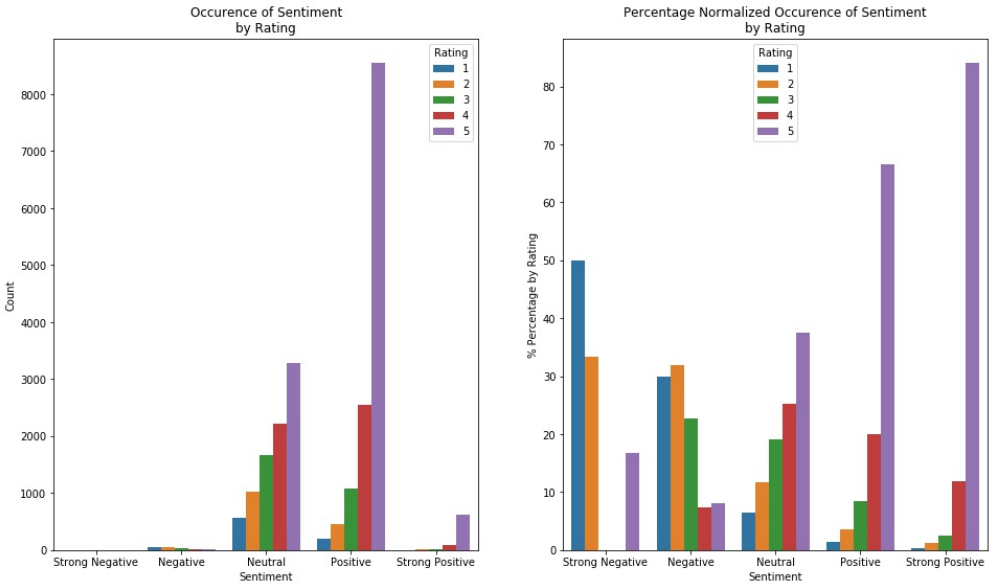
Occurence of Sentiment
by Rating
Rating
1
2
3
4
5
Count
Strong Negative
Negative
Neutral
Positive
Strong Positive
Sentiment
Percentage Normalized Occurence of Sentiment
by Rating
% Percentage by Rating

Fig. 13

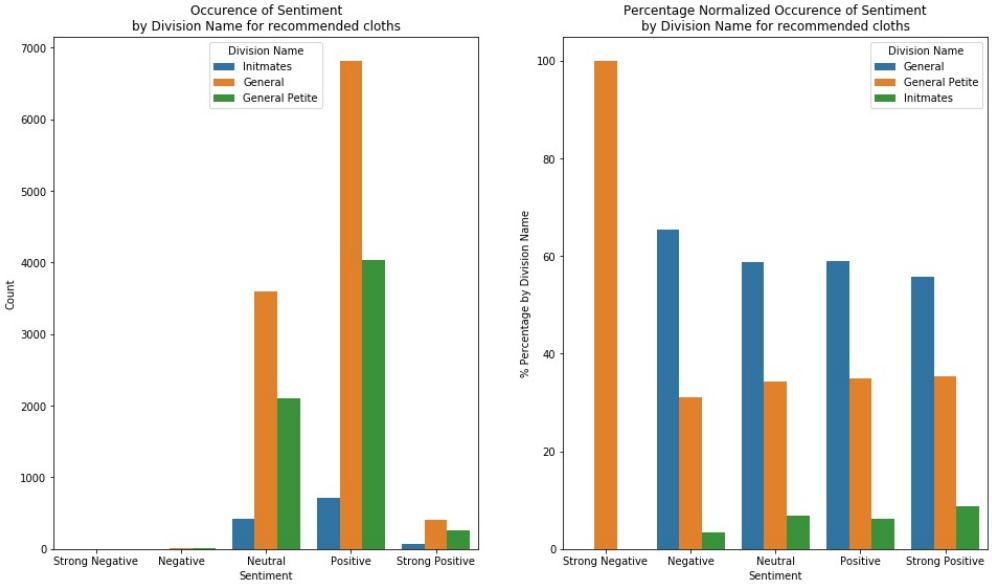
Occurence of Sentiment
by Division Name for recommended cloths
Division Name
Initmates
General
General Petite
Count
Strong Negative
Negative
Neutral
Positive
Strong Positive
Sentiment
Percentage Normalized Occurence of Sentiment
by Division Name for recommended cloths
% Percentage by Division Name

Fig. 14

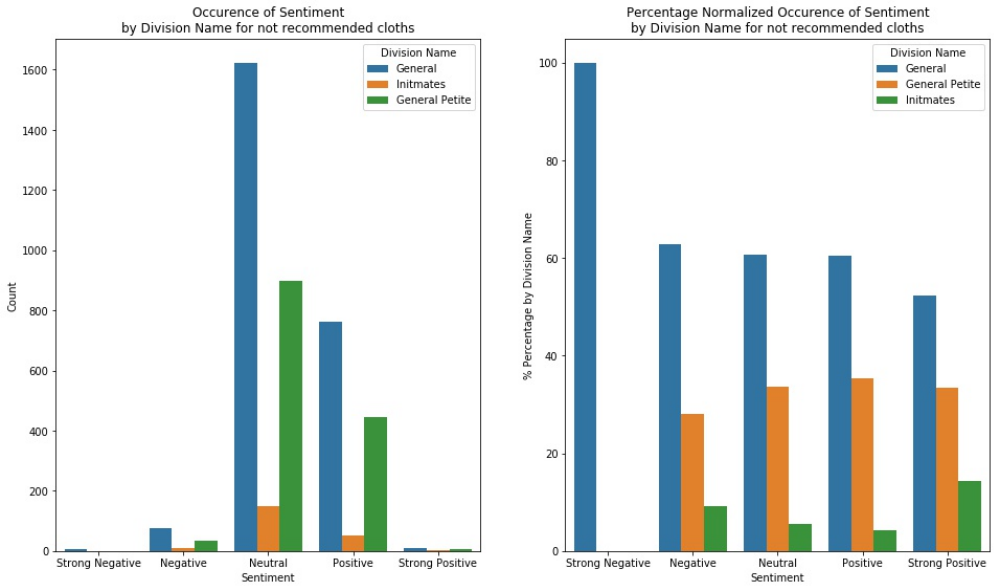

Fig. 15

https://github.com/LisaB83/Analysis-on-e-commerce-reviews/blob/LisaB83-patch-1/Capstone_Project_EDA.ipynb

https://github.com/LisaB83/Analysis-on-e-commerce-reviews/blob/LisaB83-patch-1/Capstone_Project_NLP.ipynb

4 Statistical Analysis

After analyzing the data through EDA and NLP, I search for possible correlations among different features. To find possible correlation I decided to calculate the Pearson correlation coefficient and the p-value. The Pearson correlation coefficient measures the linear relationship between two datasets. Strictly speaking, Pearson's correlation requires that each dataset be normally distributed. Like other correlation coefficients, this one varies between -1 and +1 with 0 implying no correlation. Correlations of -1 or +1 imply an exact linear relationship. Positive correlations imply that as x increases, so does y. Negative correlations imply that as x increases, y decreases. The p-value roughly indicates the probability of an uncorrelated system producing datasets that have a Pearson correlation at least as extreme as the one computed from these datasets. The p-values are not entirely reliable but are probably reasonable for datasets larger than 500 or so.

I start checking a possible correlation between age and positive feedback count, but the result shows no correlation (pearson = 0.043).

After that, I decided to calculate the count and mean of Rating, Recommended IND, Age and Positive Feedback Count grouping by Clothing ID and I plot the correlation heatmap. The results of the correlation matrix are shown in the figure 16.

We can observe that there is in fact no correlation between count and average value, which means that the popularity of the item does not lead to differential treatment when it comes to average scoring. However, there is a positive correlation of .80 between rating and recommended IND mean. It means that when the rating is high the item is recommended and viceversa. We have already seen that for the majority of the items this is right.

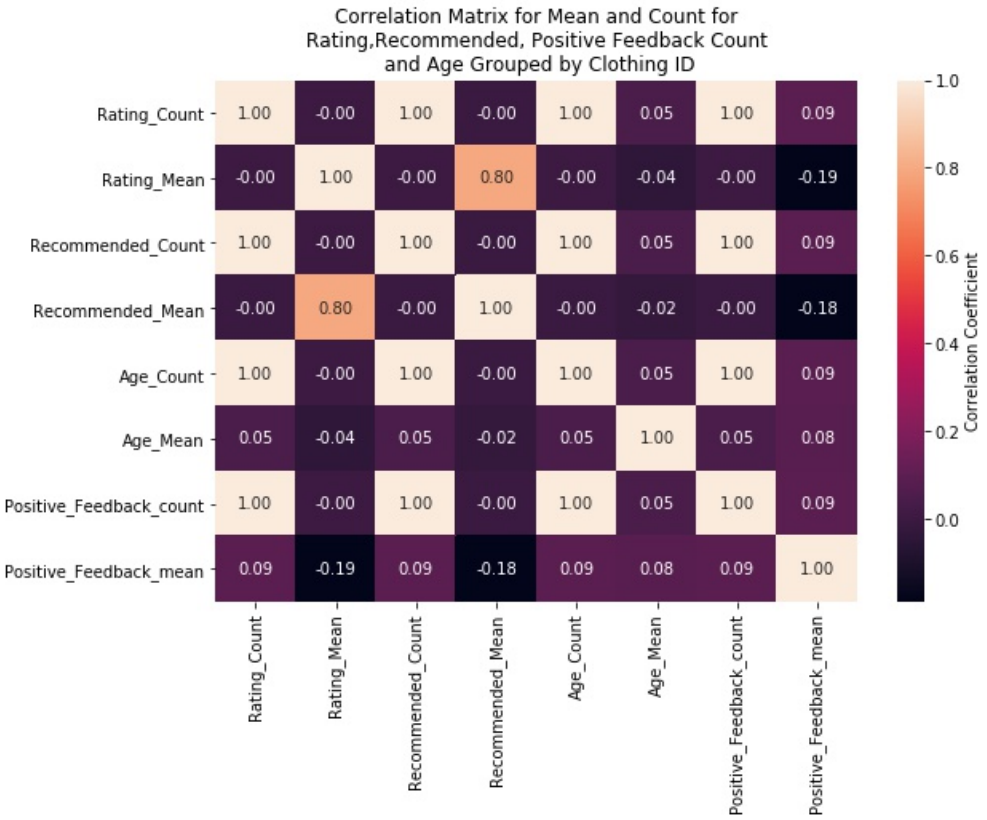
Correlation Matrix for Mean and Count for
Rating,Recommended, Positive Feedback Count
and Age Grouped by Clothing ID
Rating_Count 1.00 -0.00 1.00 -0.00 1.00 0.05 1.00 0.09
Rating_Mean -0.00 1.00 -0.00 0.80 -0.00 -0.04 -0.00 -0.19
Recommended_Count 1.00 -0.00 1.00 -0.00 1.00 0.05 1.00 0.09
Recommended_Mean -0.00 0.80 -0.00 1.00 -0.00 -0.02 -0.00 -0.18
Age_Count 1.00 -0.00 1.00 -0.00 1.00 0.05 1.00 0.09
Age_Mean 0.05 -0.04 0.05 -0.02 0.05 1.00 0.05 0.08
Positive_Feedback_count 1.00 -0.00 1.00 -0.00 1.00 0.05 1.00 0.09
Positive_Feedback_mean 0.09 -0.19 0.09 -0.18 0.09 0.08 0.09 1.00
Rating_Count
Rating_Mean
Recommended_Count
Recommended_Mean
Age_Count
Age_Mean
Positive_Feedback_count
Positive_Feedback_mean
1.0
0.8
0.6
0.4
0.2
0.0
Correlation Coefficient

Fig. 16

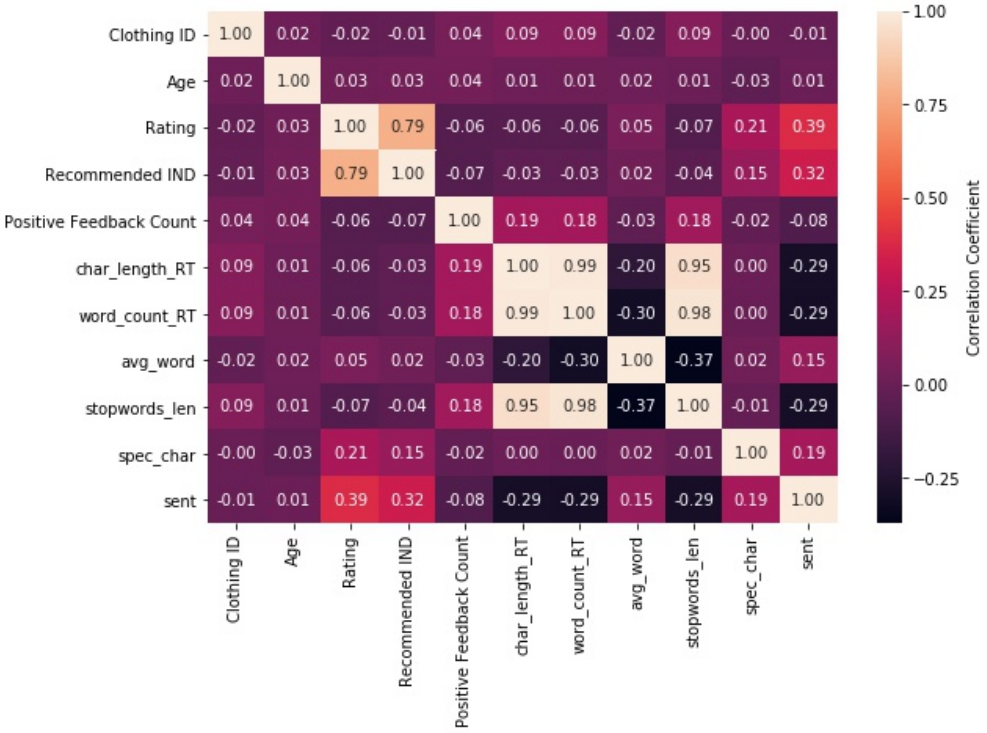

Fig.17 Correlation Matrix for the dataframe created after the work done on *Text Review*

I use the heatmap also to see the correlations for the dataframe created after the work done on Text Review (*rev_new*). From the figure 19 I can note that the polarity score (*sent*) has a slightly correlation with Rating (0.39) and also with Recommended IND (0.32). It means that, when the rating are high and the products recommended, the reviews have more positive polarity score. Also, the negative reviews are longer than the positive one.

At the end, I check the dataframe where I processed the *Title* feature. The results (Figure 18) show that there is only a slightly positive correlation between rating and the exclamation points: more '!' means higher rating.

The script that makes the data statistical analysis can be found in this link:
https://github.com/LisaB83/Analysis-on-e-commerce-reviews/blob/LisaB83-patch-1/Capstone_Project_Statistics.ipynb

5 Machine Learning

In the last section, I explore the different features to predict rating score and recommended items through different models.

The first step to use machine learning technique and develop a prediction model is to pre-process the data. Scikit-learn does not accept non-numerical features, so I need to create dummy variables and also to replace all NaN values.

After that, I determine which features were the most predictive according to the random forests regressor rf. In Fig. 21 I show the feature importance as assessed by rf and the first three most important features are *Recommended IND*, as expected after the statistical analysis, sentimental score (*sent*) and average words (*avg_word*).

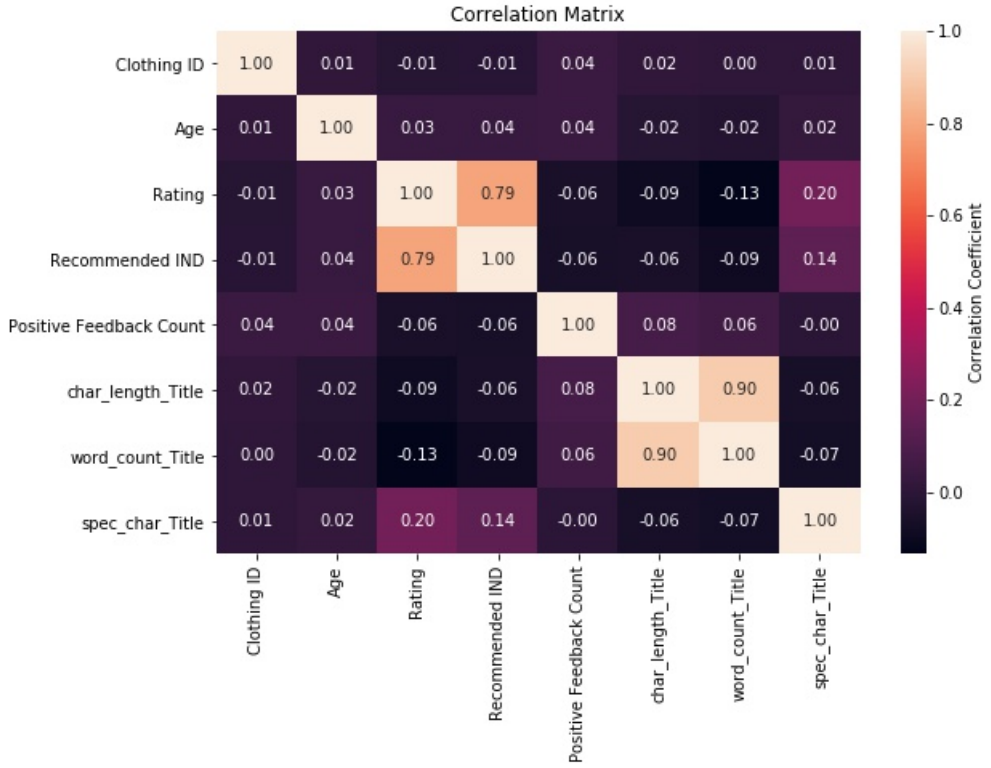

Fig.18 Correlation Matrix for the dataframe where we processed the *Title* feature

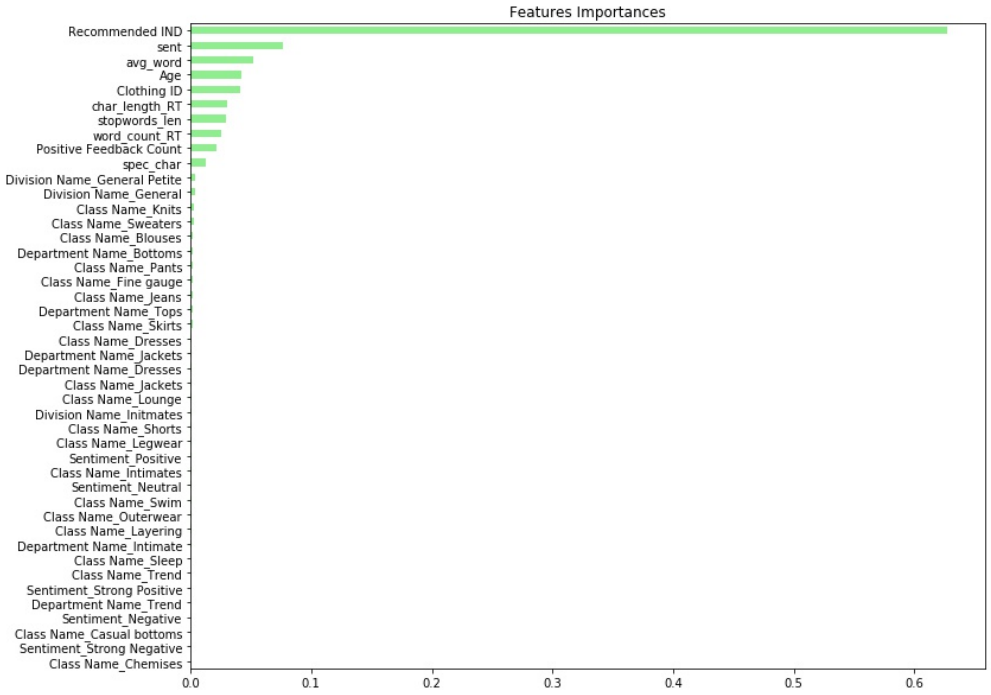

Fig. . Feature importance in the dataset.

Once the most important features are determined; I run different models to choose the best one to do rating prediction calculating the accuracy. I split the dataset into a train and test sets. The train set is used to build the predictive model, while the test set is used to evaluate the model. The train set contains 70% of the total data, while the test data contains 30% of the total data. In Table 4 the accuracy on training and test data for the different models is shown.

Table 4

Model	Accuracy on training data	Accuracy on test data
random forest classifier	0.633	0.637
logistic regression	0.627	0.633
linear SVC	0.609	0.613
bagging classifier	0.99	0.61

The Random Forest is the best model to predict rating from all the features having a better accuracy and not overfitting (as bagging classifier).

Then, I consider each feature separately and calculate the accuracy of the model to predict the rating score. Every model accuracy is around 50-60 %.

To predict the rating of the reviews from the text feature and also to derive the importance of the words in predicting product rating, I first reduce my review column as a token (which has no punctuations and stpwords). I use the Scikit-learn CountVectorizer to convert the text collection into a matrix of token counts. After that, I use the Scikit-learn TfidfTransformer to have a tfidf_matrix that has the TF-IDF values.

In Fig. 22 I show the importance of the first 25 words to predict rating.

The strongest predictor of product rating is the word *love*. *Love* is a highly emotive word that conveys a strong feeling, so I am not surprised that it can be useful in the prediction (as the word *like*). Also the words *fit* and *size*: they are both similar in nature and they are connected to the fact that the product need to fit well or not. I use Random Forest Classifier and Naive Bayes Machine Learning Algorithm to predict rating and if a cloth is recommended or not based on the words. Our models for predicting rating gives an accuracy of only 56%, while to predict recommended item our models achieve 81%.

Instead, if I use Multinomial Bayes Algorithm to predict which product has rating 5 and which 1 based on the reviews I have a great result: 96% of accuracy. I test the model, first, with the positive review: I choose a review with rating 5. After evaluating, it predict the rating as 5. Second, I test with the negative review: I choose the review with rating 1. After evaluating, it predict the rating as 1.

To finish, I use the text features to see if they help me to improve the performance of the 'total' model (predict rating based on all the other features). But after running the random forest classifier, I can see that the text features do not improve my model to predict rating.

The script that makes the feature selection and the prediction of rating and recommended items can be found in the following link:

https://github.com/LisaB83/Analysis-on-e-commerce-reviews/blob/LisaB83-patch-1/Capstone_Project_Machine_Learning.ipynb

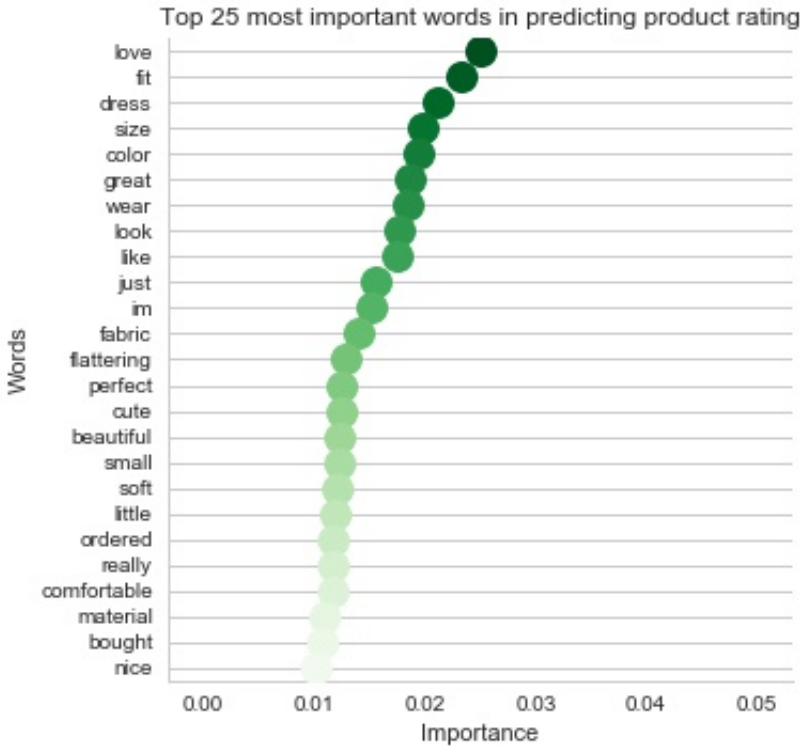

Fig. 22

6 Conclusions

During this project, I cleaned and analyzed a dataset of woman's e-commerce clothing reviews with the aim of understand the consumer's preferences, the customer sentiments and provide a rating prediction.

In this process, I have found the following features in the data:

- the age of the reviewers speads out over 80 years with a median age arounf 40;
- most of the products are recommended and with high rating;
- the density of the positive feedback count is close to 0;
- the 3 most recommended and not cloths are the same: 1078, 862 and 1094;
- the general category has most reviews, tops and dresses are the most commonly reviewed products and the most popular clothing types are dresses, knits and blouses;
- the recommended cloths have more positive sentiment in the review, while the no-recommended items more negative one;
- the distribution for the neutral, positive and strong positive sentiment increasing when the rating number increases, instead, the negative sentiment has more low rating occurence and for the strong negative sentiment, the distribution has a peak for rating 1 and 2, but also more or less 15 % of occurence for rating 5;
- the distribution of the division names for the sentiments does not seem to change depending on status of recommendation, except for the strong negative sentiment that for the recommended items shows only the 'General Petite', while for no-recommended only the 'General' division name;
- the only strong correlation is between rating and recommended IND.

For the prediction of the rating, I used a Random Forest classifier. I found an accuracy of ~60% and also adding text features didn't improve.

7 Future

The original dataset contains only woman's e-commerce clothing reviews. Future analysis could include also male's e-commerce clothing reviews to can also compare the different behaviour between male and female.