Predicting E-Commerce Product Recommendations

DULARI NARAYANAR

INTRODUCTION TO NLP AND TEXT MINING CAPSTONE PROJECT (PASS)

Contents

[**Background and Objective** 3](#_Toc85227382)

[Problem Description and Objective 3](#_Toc85227383)

[Dataset and Format 3](#_Toc85227384)

[**Phase 1: Data Retrieval and Understanding** 4](#_Toc85227385)

[Details on Original Data 4](#_Toc85227386)

[Basic Data Processing 4](#_Toc85227387)

[**Phase 2: Exploratory Data Analysis** 6](#_Toc85227388)

[Distribution Of Recommended Products 6](#_Toc85227389)

[Distribution Of Rated Products 7](#_Toc85227390)

[Distribution Of Rated vs Recommended Products 7](#_Toc85227391)

[Distribution Of Product Class Name 8](#_Toc85227392)

[Distribution Of Department Name 8](#_Toc85227393)

[Distribution Of Age 9](#_Toc85227394)

[Distribution Of Positive Feedback Count 9](#_Toc85227395)

[Distribution Of Top 10 product Ids 10](#_Toc85227396)

[Distribution Of Top 10 product Ids who recommended versus did not Recommend 10](#_Toc85227397)

[Distribution Of Character Length of Customer reviews 11](#_Toc85227398)

[Distribution Of Word Count of Customer Reviews 11](#_Toc85227399)

[**Phase 3: Feature Engineering and Modeling** 13](#_Toc85227400)

[Training and Testing Dataset 13](#_Toc85227401)

[Experiments Workflow 13](#_Toc85227402)

[Experiment 1: Basic NLP Count based features, Structured Features (Age and Feedback Count) + Logistic Regression 14](#_Toc85227403)

[Experiment 2: Features from Sentiment Analysis + Logistic Regression 14](#_Toc85227404)

[Experiment 3: BOW features (1-gram) + Logistic Regression 15](#_Toc85227405)

[Experiment 4: BOW features (2-gram) + Logistic Regression 16](#_Toc85227406)

[Experiment 5: BOW features (3-gram) + Logistic Regression 16](#_Toc85227407)

[Experiment 6: BOW features (3-gram) + Feature Selection + Logistic Regression 17](#_Toc85227408)

[Experiment 7: BOW features (3-gram) + Feature Selection + Structured Features (Age and Feedback Count) + Logistic Regression 18](#_Toc85227409)

[Experiment 8: BOW features (3-gram) + Feature Selection + Structured Features (Age and Feedback Count) + TF-IDF features 19](#_Toc85227410)

[Experiment 9: BOW features (3-gram) + Feature Selection + Structured Features (Age and Feedback Count) With Random Forest 19](#_Toc85227411)

[**Phase 4: Evaluation, Insights and Recommendations** 20](#_Toc85227412)

[Performance Evaluation Summary of different Experiments 20](#_Toc85227413)

[**Key assumptions** 22](#_Toc85227414)

[**Recommendations** 22](#_Toc85227415)

# **Background and Objective**

## Problem Description and Objective

This capstone project deals with an e-commerce store which sells women’s clothing and apparel. The objective of the project is to leverage the customer review text attributes amongst other structured features and try to build a suitable predictive model and suitable predictor features to predict if the purchased product will be recommended (1 or 0) by the customer.

## Dataset and Format

The dataset used is a real-world dataset and all references to the company in the review text and body have been replaced with the word ‘retailer’.

The dataset includes 23486 rows and 10 columns (features).

Each row corresponds to a customer review, and includes the features:

* **Clothing ID:** Integer Categorical variable that refers to the specific piece being reviewed.
* **Age:** Positive Integer variable of the reviewer’s age.
* **Title:** String variable for the title of the review.
* **Review Text:** String variable for the review body.
* **Rating:** Positive Ordinal Integer variable for the product score granted by the customer from 1 Worst to 5 Best.
* **Recommended IND:** Binary variable stating where the customer recommends the product where 1 is recommended, 0 is not recommended.
* **Positive Feedback Count:** Positive Integer documenting the number of other customers who found this review positive.
* **Division Name:** Categorical name of the product high level division.
* **Department Name:** Categorical name of the product department name.
* **Class Name:** Categorical name of the product class name.

**The data set is available from many sources namely:**

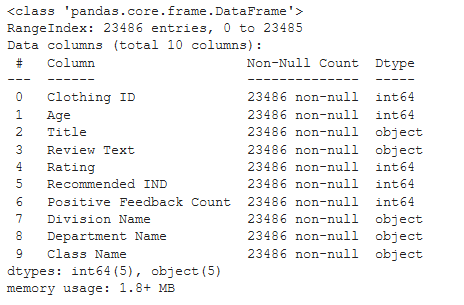
* At <https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews> .
* From [**GitHub Repo**](https://github.com/dipanjanS/text-analytics-with-python/raw/master/media/).
* Kaggle API and the following command via CLI to get it.
  + **kaggle datasets download -d nicapotato/womens-ecommerce-clothing-reviews**

**Phase 1: Data Retrieval and Understanding**

## Details on Original Data

In this phase, the women’s e-commerce clothing reviews dataset was first loaded up which is provided as a flat file. Once the data was read into a dataframe, the following attributes were observed:

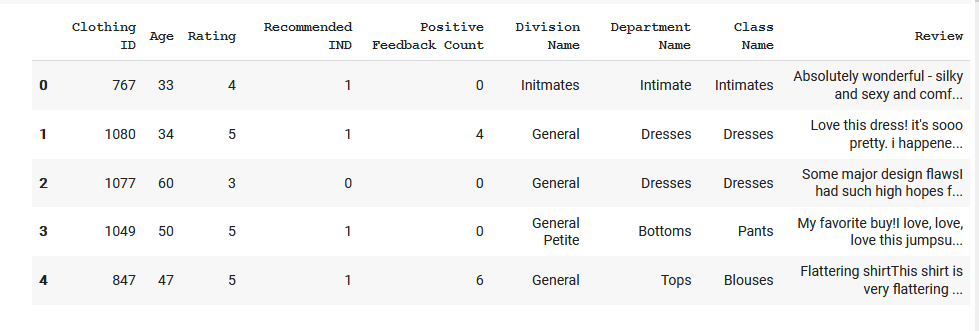
**Fig A: The Original Data after being read into a dataframe**



## Basic Data Processing

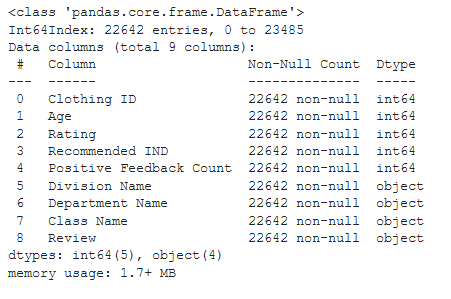
1. Since both the Title column and Review Text column both contain important keywords which can indicate the customer’s overall sentiments of the product, it makes sense to combine both columns, so that we need to only focus on one combined column. The columns **Title** and **Review Text** were combined into one text column called **Review** which will be used later for NLP tasks and the **Title** and **Review Text** columns were then removed.

**Fig B: The resulting Table after Combining Title and Review Text Columns**

****

1. All records with an empty value in the Review Column were deleted as these would not have any predictive input in the later NLP stages.

**Fig C: Final Data after Basic data processing**



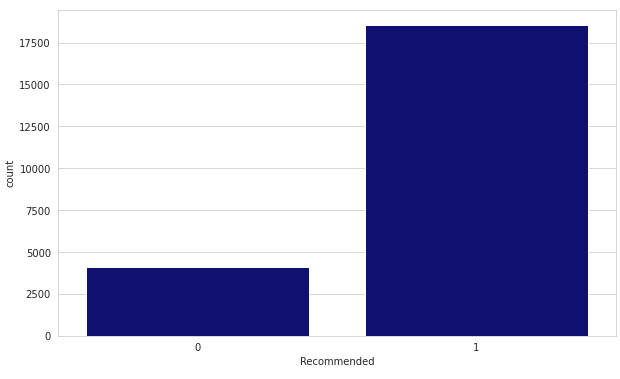
1. The extra extension **IND** can be confusing to the reader and thus **the Recommended IND** column was renamed to **Recommended.**

**Phase 2: Exploratory Data Analysis**

In this section some basic exploratory data analysis was done to get a feel for the dataset that is being analyzed. This helped in skimming through at the data before making any assumptions. By visualizing a few charts with the notebook, it was easier to identify obvious errors, understand patterns within the data, detect outliers or anomalous events and find interesting relations among the variables.

## Distribution Of Recommended Products

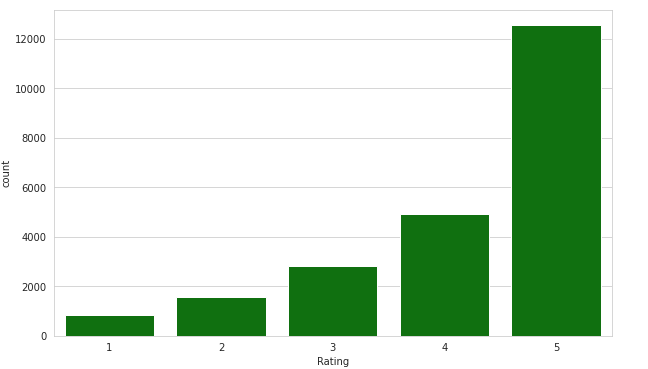
**Fig D: Number of products recommended and not recommended by customers**



The Chart suggests that there are many products recommended by the customers and very few which have not been recommended (almost 3 times more). This implies that the customers are in general very happy with the quality of the store’s products. This does give the feel, there is a higher probability that a product will be recommended than not recommended. It will be interesting to know whether these are blind recommendations or whether customers feel very strongly about the products and thus a rating distribution visualization will be good to see.

## Distribution Of Rated Products

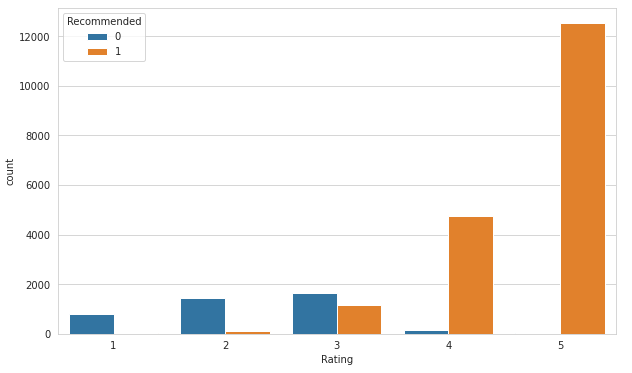
**Fig E: Distribution of Reviews Based on Ratings by Customers**



Here again we can see that the number customers who have given 5-star ratings far exceeds the other ratings, which again gives more reliability in the previous visualization and that a lot of customers are indeed happy with the products and are thus recommending. A rating versus recommendation distribution would further help in understanding whether there is a tally between the classes rating high and recommending high and vice versa.

## Distribution Of Rated vs Recommended Products

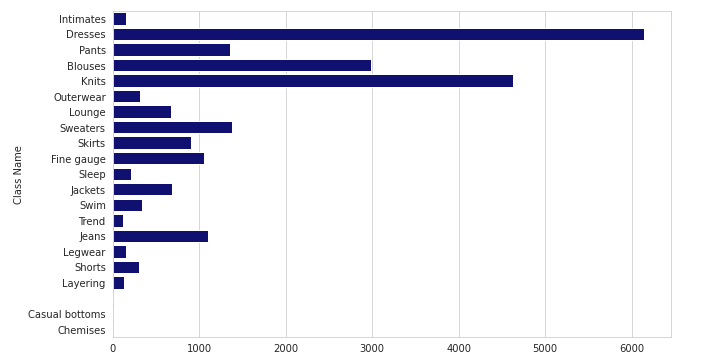
**Fig F: Distribution of Reviews Based on Ratings vs Recommendations by Customers**



It is observed that low ratings (1-2) relate to customers not recommending products and high ratings (4-5) lead to customers recommending products. Since Rating (predictor) thus relates strongly to the outcome variable (Recommendations) it causes data\target leakage to build an unbiased predictive model the **Rating Column is dropped from the analysis**.

## Distribution Of Product Class Name

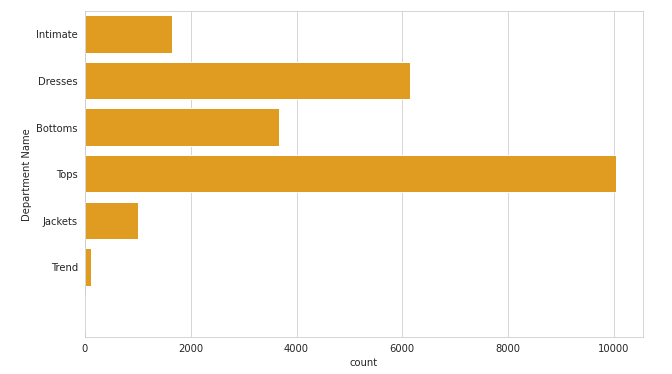
**Fig G: Distribution of Reviews Based on Product Class Name by Customers**

****

The distributions shows that most of the reviews were for Dresses, followed by Knits and the blouses. This could be since these were the products which were sold the most as well and need suggest any trend.

## Distribution Of Department Name

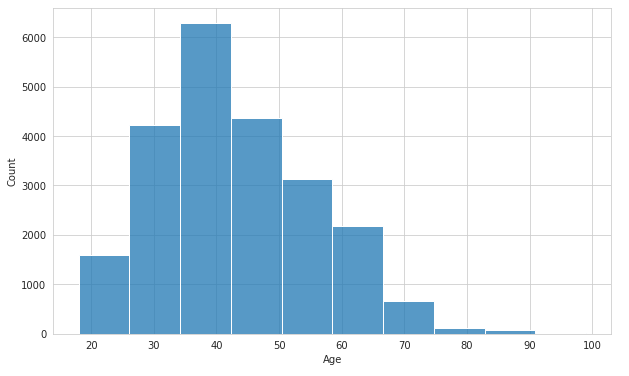
**Fig H: Distribution of Reviews Based on Department Name by Customers**

****

The distribution shows that most of the reviews were from the Tops followed by Dresses Department. This correlates with the product class name distribution as most probably the blouses and knits are sold in the Tops Department and the Dresses sold in the Dresses department. This distribution does not give any further valuable information which is not already obtained from the Product class distribution unless we are interested to analyze whether the Department staff service had an influence in the reviews.

## Distribution Of Age

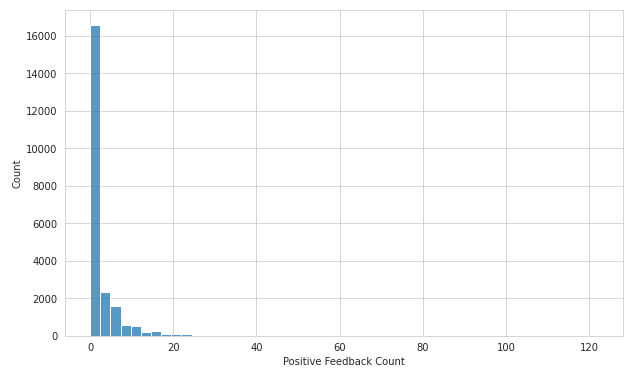
**Fig I: Distribution of Reviews Based on Age of Customers**

****

It is interesting to note that most of the reviews came from the customers in the 40s range follows by mid 40s and mid-30s. This could suggest they comprised a big bulk of the customers since they have the earning capacity. It could also stem from the fact that this age group are more vocal or interested to know about reviews and use the feature more. The age group does give a kind of reliability of the reviews as we tend to think this age group gives more honest or unbiased opinions. But then again, there is the possibility that they also may fall under the category of people who are most likely to give paid reviews.

## Distribution Of Positive Feedback Count

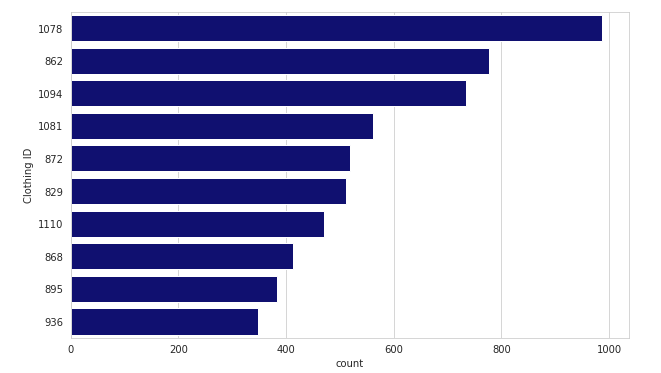
**Fig J: Distribution of Reviews Based on Positive Feedback Count**

****

It is observed that the positive feedback count for reviews is less mid-20s. Majority of the reviews had 0 Positive Feedback.

## Distribution Of Top 10 product Ids

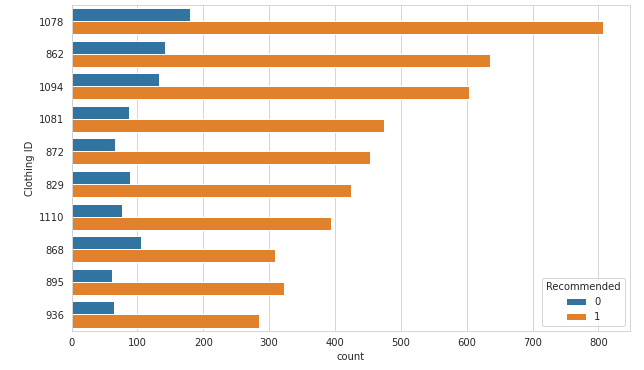
**Fig K: Distribution of Reviews Based on Top 10 Product IDs**

****

The Chart shows the Product ID 1078, followed by 862 and 1094 received the most reviews

## Distribution Of Top 10 product Ids who recommended versus did not Recommend

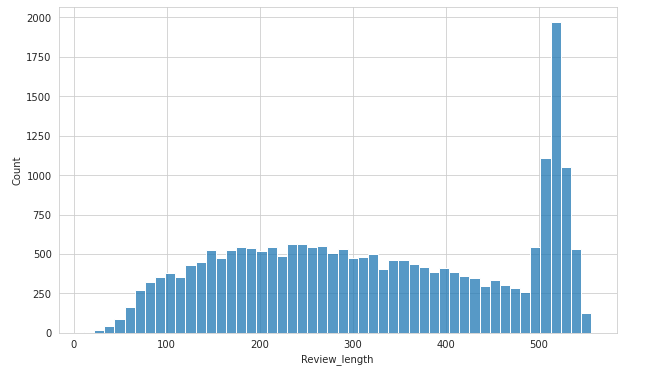
**Fig L: Distribution of Top 10 Product IDs in comparison to recommendation**

****

We can see the top 10 product ID reviewed had high recommendations as well

## Distribution Of Character Length of Customer reviews

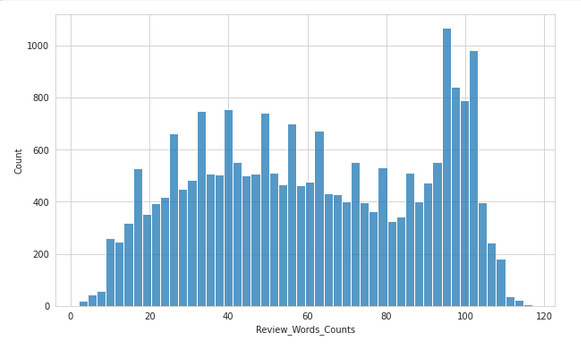
**Fig M: Distribution of Character length of Customer Reviews**

****

This textual data distribution shows that there was a good demarcation between the number of people who wrote short and long remarks. There is a significant number of people who wrote very long reviews. This Data would be interesting to research further as there will be key information which would help in understanding the sentiments of the customers.

## Distribution Of Word Count of Customer Reviews

**Fig N: Distribution of Word Count of Customer Reviews**

****

We cannot exactly see any word count patterns and it seems the distribution is pretty even except for a few high outliers. Since are 14 specific outliers, it will be interesting to dig in further if required to evaluate the reviews and see if there are any patterns or reasons for these.

**Phase 3: Feature Engineering and Modeling**

## Training and Testing Dataset

Based on the Visualization done in the previous section, ‘Review’,  ‘Age’,  ‘Positive Feedback Count’ seem to be the useful predictors for the outcome Recommended. The rest of the features are thus removed from the data set that will be used as the Training and Testing sets.

The Training and Testing Sets were then split into a 7: 3 ratio

## Experiments Workflow

In each Experiment, relevant combination of Features was extracted from the text area and machine learning was used to build predictive models to compare their performances. The combinations were as follows:

* Using structured features like Age, Positive Feedback Count along with other non-vectorized NLP count-based features to build classifiers
* Sentiment Analysis features
* Using vectorization methods like BOW based features 1,2,3-grams with \ without feature selection and build classifiers
* Combine vectorized BOW based features and structured features and build classifiers
* TF-IDF based features

For each experiment a suitable Machine Learning Model is chosen, the data is split into training and testing data. The model is trained on training data and validations are made on the test data. The performance of the model is then evaluated using confusion matrix and relevant metrics.

The models were evaluated based on these metrics

* **Accuracy** = TP+TN/TP+FP+FN+TN is a ratio of correctly predicted observation to the total observations. Accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same. Therefore, we would have to look at other parameters to evaluate the performance of our model.
* **Precision:** is TP/TP+FP. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all reviews that labeled as positive, how many are positive? Out of all the predictions made by a model for a class, how many are correct.
* **Recall:** (Sensitivity) is = TP/TP+FN. Recall is the ratio of correctly predicted positive observations to all observations in actual class. The question recall answers are: Of all the reviews, how many did we predict correctly?
* **F1-score:** = 2\*(Recall \* Precision) / (Recall + Precision). F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. In this capstone, it was observed that the class distribution was uneven and thus monitoring this score was crucial to decide which feature and model combination was the best.
* Code for listing and viewing feature importance was used for Experiment 8, but given the number of matrix features, it did not deem to be useful to use it.

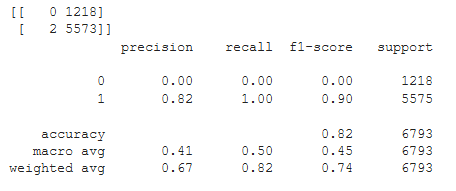
## Experiment 1: Basic NLP Count based features, Structured Features (Age and Feedback Count) + Logistic Regression

A few basic text-based NLP count features were created which were evaluated on whether being helpful in improving text classification models. The following Features were added along with Age and Feedback Count:

* **Word Count:** total number of words in the documents
* **Character Count:** total number of characters in the documents
* **Average Word Density:** average length of the words used in the documents
* **Punctuation Count:** total number of punctuation marks in the documents
* **Upper Case Count:** total number of upper count words in the documents
* **Title Word Count:** total number of proper case (title) words in the documents

A logistic regression model which is easy to train, interpret and works well on a wide variety of NLP problems was used

The Performance Metrics results



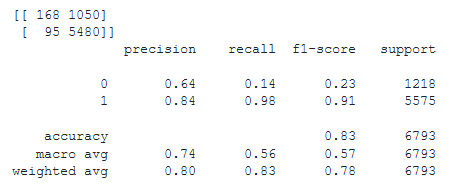
The results show that the precision, recall and F1 score for recommendation = Yes is performance of the model was good, but the model failed very badly in predicting no Recommendations. It was also observed the true positive and False Positive in the confusion matrix were not good. This means the features selected may not have been good predictors and for the next round of experiment the text counts are dropped as predictors.

## Experiment 2: Features from Sentiment Analysis + Logistic Regression

In this experiment features from unsupervised, lexicon-based sentiment analysis are used. Since reviews are subjective in nature and have strong emotions, feelings, the text documents can be used for extracting sentiment as a feature. The general expectation is that highly rated and recommended products (**label 1**) should have a **positive** sentiment and products which are not recommended (**label 0**) should have a **negative** sentiment.

Textblob which is an open-source python library for processing textual data is used. It performs different operations on textual data such as noun phrase extraction, sentiment analysis, classification, translation, etc. The sentiment function of textblob returns two properties, **polarity**, and **subjectivity**. Polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement. Subjective sentences refer to opinion, emotion or judgment whereas objective refers to factual information. Subjectivity is also a float which lies in the range of [0,1].

The Performance Metrics results



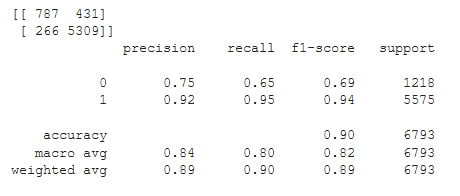
The results show that the precision, recall and F1 score for recommendation = Yes performance of the model was good and even better than Experiment 1. The model performed slightly better in predicting in no Recommendations than Experiment 2. The true positive numbers improved. But the false positive values are worrying still. This means the Sentiment analysis features was a slightly better at predicting. Overall precision only improved by 0.01. We would need to find more features to improve the model.

## Experiment 3: BOW features (1-gram) + Logistic Regression

In this Experiment the BOW (Bag of Words) features were used. The bag of words model represents each text document as a numeric vector where each dimension is a specific word from the corpus and the value could be its frequency in the document, occurrence (denoted by 1 or 0) or even weighted values.

The Pre-processed Clean review column words were used for this experiment.

The Performance Metrics results

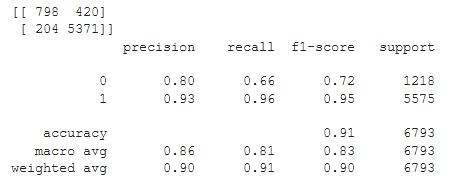


The results show that the precision, and F1 score for recommendation = Yes performance of the model improved and even better than Experiment 2, while recall dropped slightly. The number of True positives has increased, and the number of false positives has dropped. The model performed much better in predicting in no Recommendations than Experiment 2. This means the Bag of Words features were a more accurate predictors than Sentiment analysis. Overall accuracy improved also to 0.90.

## Experiment 4: BOW features (2-gram) + Logistic Regression

Experiment 3 was repeated again but with 2 grams.

The Performance Metrics results

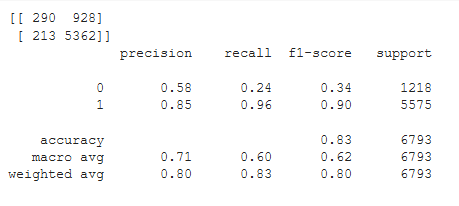


The results show that the precision, recall and F1 score for recommendation = Yes performance of the model improved and even better than Experiment 2. The model performed much better in predicting in no Recommendations than Experiment 3. The number of True positives and the number of false positives remained around the same. This means the Bag of Words with 2 grams features were a more accurate predictors than Bag of Words with 1 gram. Overall accuracy improved also to 0.91.

## Experiment 5: BOW features (3-gram) + Logistic Regression

Experiment 3 was repeated again but with 3 grams.

The Performance Metrics results

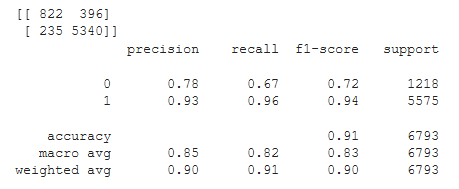
the precision, recall and F1 score for recommendation = Yes performance of the model in predicting in no Recommendations was back to the same as 

The results show that the precision, recall and F1 score for recommendation = Yes performance of the model in predicting in no Recommendations was worse than Experiment 4. Overall accuracy also dropped to 0.83 as well. The number of false positives increased as well. This means the Bag of Words with 2 grams features was the best predictors than Bag of Words with 1 gram or 3 grams.

## Experiment 6: BOW features (3-gram) + Feature Selection + Logistic Regression

The min\_df is set as 3 in CountVectorizer and other parameters were kept the same as the previous experiment Thus we have dropped all words \ n-grams which occur less than 3 times in all documents.

The Performance Metrics results

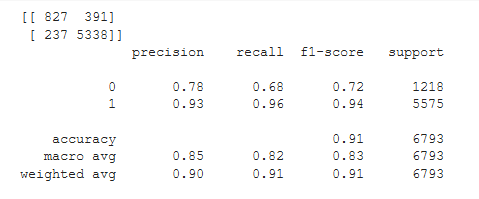


The results show that Experiment 4. Overall accuracy also went back to 0.91. The number of false negatives reduced. This means the Bag of Words with 2 grams features was the best predictors than Bag of Words with 1 gram or 3 grams, but if 3 grams has to be chosen then the minimum number of occurrences has to be increased to get a good predictable model.

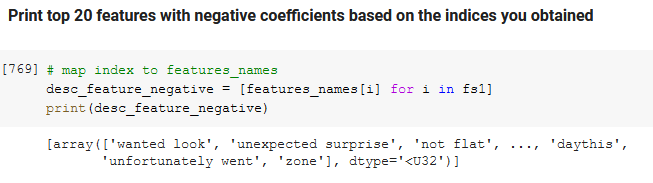
## Experiment 7: BOW features (3-gram) + Feature Selection + Structured Features (Age and Feedback Count) + Logistic Regression

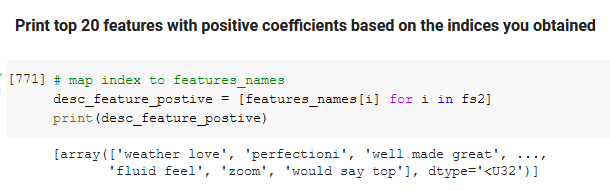
In this experiment the sparse features of BOW are combined first with the structured features (**Age, Feedback count, polarity, subjectivity) after** converting them into sparse format.

The Performance Metrics results



The results show just a slight improvement in recall of the model in predicting in no Recommendations in comparison to Experiment 6 and 4. By far this combination of features gives the best output.

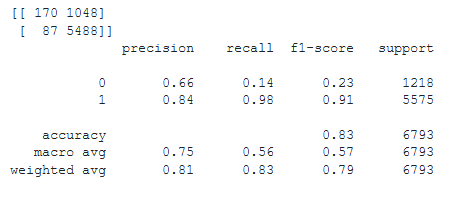




As can be seen from the top 20 negative and positive coefficients, the bag of words and n grams have served as the best predictors.

## Experiment 8: BOW features (3-gram) + Feature Selection + Structured Features (Age and Feedback Count) + TF-IDF features

The Performance Metrics results

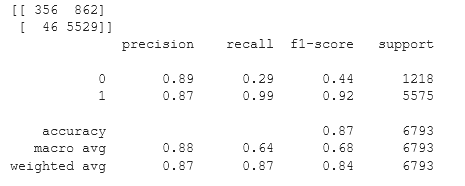


The precision, recall and F1 scores were not as impressive as Experiment 4,6 and 7. The false positive number has increased drastically from Experiment 7.

## Experiment 9: BOW features (3-gram) + Feature Selection + Structured Features (Age and Feedback Count) With Random Forest

In this experiment all the features of Experiment 7 were retained as it has shown the best outcome and Random Forest Algorithm was used. The reason for choosing to try it out as an experiment is because, it is known to help reduce overfitting in decision trees and helps to improve the accuracy and is also flexible to both classification and regression problems while working well with both categorical and continuous values. The model was then evaluated

The Performance Metrics results



Although the precision, recall and F1 score for recommendation = Yes performance of the model was reasonable but did not match up to Experiment 7, the model did not perform well in predicting in no Recommendations as can be seen from its low recall and f1-scores

**Phase 4: Evaluation, Insights and Recommendations**

## Performance Evaluation Summary of different Experiments

The model which performed the best and is recommended is Experiment 7

Details of the experiments are the results are discussed in detailed in Phase 3 section. Thus, the below table gives a summary of the results and the best 3 models. As discussed earlier since F1 scores were important for this study, the selection did take into consideration on the best f1 scores as well in comparison to accuracy.

**Table A: The Comparisons of The Models in Different Experiments**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Exp  No. | Features Used | ML Model Used | Metrics | | | | | Rating |
| Accuracy |  | Precision | Recall | F1-Score |
| 1 | Basic NLP Count based features, Structured Features (Age and Feedback Count) | Logistic Regression | 82% | 0 | 0 | 0 | 0 |  |
| 1 | 0.82 | 1.00 | 0.90 |
| 2 | Features from Sentiment Analysis | Logistic Regression | 83% | 0 | 0.64 | 0.14 | 0.23 |  |
| 1 | 0.84 | 0.98 | 0.91 |
| 3 | BOW features (1-gram) | Logistic Regression | 90% | 0 | 0.75 | 0.65 | 0.69 |  |
| 1 | 0.92 | 0.95 | 0.94 |
| 4 | BOW features (2-gram) | Logistic Regression | 91% | 0 | 0.80 | 0.66 | 0.72 | 3 |
| 1 | 0.93 | 0.96 | 0.95 |
| 5 | BOW features (3-gram) | Logistic Regression | 83% | 0 | 0.58 | 0.24 | 0.34 |  |
| 1 | 0.85 | 0.96 | 0.90 |
| 6 | BOW features (3-gram) + Feature Selection | Logistic Regression | 91% | 0 | 0.78 | 0.67 | 0.72 | 2 |
| 1 | 0.93 | 0.96 | 0.94 |
| 7 | BOW features (3-gram) + Feature Selection + Structured Features (Age and Feedback Count) | Logistic Regression | 91% | 0 | 0.78 | 0.68 | 0.72 | 1 |
| 1 | 0.93 | 0.96 | 0.94 |
| 8 | BOW features (3-gram) + Feature Selection + Structured Features (Age and Feedback Count) + TF-IDF features | Logistic Regression | 83% | 0 | 0.66 | 0.14 | 0.23 |  |
| 1 | 0.84 | 0.98 | 0.91 |
| 9 | BOW features (3-gram) + Feature Selection + Structured Features (Age and Feedback Count) | Random Forest | 87% | 0 | 0.89 | 0.29 | 0.44 |  |
| 1 | 0.87 | 0.99 | 0.92 |

**Key assumptions**

The assumption made in this case study as mentioned earlier is that the reviews are genuine and depict the real sentiments of the buyers and are not bought review in favor or against the company. If not the predictive power of the review data will be diminished.

The assumption is that the reviews have words which are all spelt correctly and there are no spelling variations.

**Recommendations**

Since the features are extracted from reviews which can be very noisy datasets, if you don’t preprocess enough, it is going to be garbage-in-garbage-out and the model is not going to be reliable one.

It would be good to use a spell checker function and lemmatize the words to transform words to the actual root. This would help in reducing the number of duplicate words and normalize the features to a certain extent.

Text normalization is important for noisy texts in reviews where abbreviations, misspellings and use of out-of-vocabulary words (oov) are prevalent. Some common approaches to text normalization to do dictionary mappings (easiest), statistical machine translation (SMT) and spelling-correction based approaches.