Emine Akbulut,

akbulutemine7@gmail.com 9757-500-1667

Health and Personal Care Product Amazon Review Sentiment Analysis

September 15, 2020

Contents

1. INTRODUCTION

1.1. General :

Sentiment analysis that is one of the subtopics of Natural Language Processing (NLP) is increasingly becoming popular. It is text mining that identifies and pulls up subjective information in the source material and helps a business organization to better understand how people in society reacts to their brand, product or service by scrutinizing online conversations.

There are many application areas of sentiment analysis that ranges from e-commerce, marketing, to politics and other types of research to deal with unstructured text data. Companies in e-commerce sector, for example, perform sentiment analyses in order to gather customer feedback about their products and services. Additionally, prospective customers tend to review the feedback of the existing customers before purchasing a product or service of a given company. As indicated here, there are two actors in e-commerce, which are 1) the online retailer who aims to maximize its sales of products or services 2) consumers who aim to buy the best product or service over other alternatives.

1.2 Problem:

Health and personal care products are increasingly important and even vital for people's health after the COVID-19 outbreak. What customers think about how effective these products in making us clean and healthy has not been investigated, however. In this project, my client is Amazon, and I will provide a software tool to Amazon in a way I will identify positive and negative words included in customers' reviews for health and personal care products. For this purpose, I will use the customer reviews on these products in the period of 2004-2014.

Based on these reviews in the said time period, I will develop a sentiment analysis model through natural language processing to determine customers' sentiment toward these products. By developing such a model, I will provide my client, Amazon, better understanding of customers' opinion and thoughts about its health and personal care products. **Thus, the central purpose of this project is to collect and analyze customer feedback about Amazon's health and personal care products, and provide Amazon the detailed analyses of customer impressions on these products so that Amazon can benefit from these analyses in order to make improvements on its products and maximize e-commerce sales.**

1.3 Data Set Description:

The data includes features in relation to customer reviews and ratings on health and personal care products. The data has 55,076 rows and 9 features. In each row, I have a customer review

	reviewerID	asin	reviewerName	helpful	reviewText	overall	summary	unixReviewTime	reviewTime
0	A1N655X9X7C6QY	B00000J47L	C. Cook "LIVE.....LOVE.....AND.....	[0, 0]	Right out of the box and ready to go !Everythi...	5.0	Out of the box ready to go.....	1357948800	01 12, 2013
1	AEL6CQNQXONBX	B00000J47L	Cute Chihuahua	[1, 1]	In this day and age it is hard to understand w...	5.0	SAVE LOTS OF MONEY AND THE ENVIRONMENT AT THE ...	1227571200	11 25, 2008
2	A2032LF6FWWK8E	B00000J47L	Jay Riemenschneider	[6, 10]	At 2500ah, these rechargeable NiMh batteries p...	5.0	Excellent power, quality NiMh batteries to go ...	1141689600	03 7, 2006

and the following variables: Each row in the data corresponds to a customer review, and includes the following variables:

reviewerID : ID of the reviewer, e.g. A2SUAM1J3GNN3B - type: object

asin : ID of the product , e.g. 0000013714 – type: object

reviewerName : name of the reviewer – type: object

helpful : helpfulness of the review, e.g. 2/3 – type: object

reviewText : text of the review – type: object

overall : Rating (1,2,3,4,5)– type: float64

summary : summary of the review – type: object

unixReviewTime : time of the review (unix time) – type: int64

reviewTime : time of the review (raw) – type: object

The data was located in Standford Analysis Project webpage. The original data was stored in a JSON format there. To analyze the data, I have to change the format of the data. For this purpose, I import JSON and decode JSON file with using query so I can convert JSON file to csv file format.

Data Source:

http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Health_and_Personal_Care_10.json.gz

2. DATA WRANGLING

2.1 Inspecting the Data Set:

```
# Basic Information on dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 55076 entries, 0 to 55075
Data columns (total 9 columns):
reviewerID        55076 non-null object
asin              55076 non-null object
reviewerName      54787 non-null object
helpful           55076 non-null object
reviewText        55076 non-null object
overall           55076 non-null float64
summary           55076 non-null object
unixReviewTime    55076 non-null int64
reviewTime        55076 non-null object
dtypes: float64(1), int64(1), object(7)
memory usage: 4.2+ MB
```

The data includes 55,076 rows(observations) and 9 columns(feature variables) and its memory usage is 4.2+ MB. In the dataset, we have 7 object, 1 float64 and 1 int64 data types.

289 **'reviewerName'** is missing in the data. Because customers don't reveal their identity, the results of the analysis may not be reliable. I would prefer to drop the missing values from the data because I have sufficient amount of observations to produce a prediction as a result of my sentiment analysis.

'reviewText' and **'summary'** variables are concatenated because both variables provide the similar information about a given product in text format. **'helpful'** feature was also dropped because I do not need to use that column for my model need that column for my analysis.

'overall' (ratings) are categorized as good and bad to perform the sentiment analysis. For this purpose, I dropped the observations where 'overall' columns' values are equal to 3 because this particular rating group doesn't provide a specific opinion about as to whether a given product is good or bad. I created a new column that is named as 'rate_class' based on the 'overall' column and converted its' values as 'good' and 'bad'. Then, I dropped the 'overall' column.

In the dataset, **'reviewerID'** and **'reviwerName'** variables were both used for identification of customers. I dropped **'reviewerName'** variable since the names of customers were not in a standardized form. There are many different styles of the customer name variable to represent them in it.

'unixReviewTime' variable was dropped since the **'reviewTime'** variable already represents what this variable capture in a more understandable format. Also, I converted **'reviewTime'** to datetime data type and a new **'year'** column was generated to perform the analysis on the other variables in the future work. After that, 'reviewTime' column was also dropped.

I renamed the columns to enhance readability of the coding:

reviewerID : **"customer"**

asin : **"product"**

reviewText: This variable will be concatenated with the "summary" variable and renamed as **"review_text"**

overall: **"rating_class"**

reviewTime: **"year"**

2.2 Descriptive Statistics:

In my dataset, I have 4,625 reviews which includes bad ratings whereas 41,879reviews that includes have good ratings.

I have 2,184 unique customers and 1,260 products in this dataset. On average, each customer writes 25 reviews for products and on average each product has 43 reviews in the website.

2.3 Preprocessing the Text:

Because the text is in unstructured form, there are various types of noise in the data and it is not readily in analyzable format without doing any pre-processing. Text preprocessing is a process of cleaning, making the data noise-free and make it ready for analysis. In this section, I apply the following text preprocessing, respectively.

Removing HTML tags

The HTML tags were removed by a function I wrote. These tags typically do not add a significant value towards analyzing the text.

Removing accented characters

I wrote a function in order to convert and standardize accented characters into ASCII characters.

Expanding Contractions

Another function has been written to convert each contraction to its expanded and original form for the purpose of text standardization.

Removing Special Characters

Simple regular expressions(regexes) have been used to remove special characters and symbols. They are usually non-alphanumeric characters or occasional numeric characters.

Lemmatization

I removed word affixes so I can get to the base form of a word.

Removing stopwords

I wrote a function to remove stopwords. They do not have any significance in the text.

Building a Text Normalizer

Given the functions written above and additional techniques for text correction (such as lowercase the text, and remove the extra newlines, white spaces, apostrophes), I developed a text normalizer to preprocess the new_text document.

After applying text normalizer to ‘the review_text’ document, I applied tokenizer to generate tokens for the clean text. After that, I had 4,572,979 words in total with a vocabulary size of 53,506. Maximum review length is 2760 while minimum review length is 2 word.

Finally, after finishing up all data wrangling and preprocessing phases, I save the dataframe to csv file as ‘cleaned_dataset’.

A clean dataset will enable a model to learn meaningful features and not overfit on irrelevant noise. After completing these steps and checking if there are additional errors, I can get started using the clean and labelled data in order to train models in the modeling section.

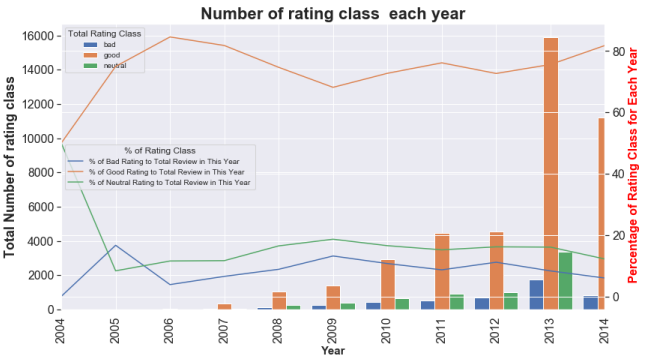
3. DATA STORYTELLING

3.1 Target Variable ("rating_class")

There are 54787 reviews and ratings that are wroted for each health and personal care product which they are bought in the Amazon between 2004-2014. This above graph demonstrates most of costumer satisfy with product.
I categorized those 5 rating categories into 3 categories such as good, bad ,and neutral for doing a better sentiment analysis on their reviews. As the graph shows, %76.44 of reviews (41879) are classified as good, %15.12 of reviews(8283) are classified as neutral. %8.44 of reviews(4625) are classified as bad..

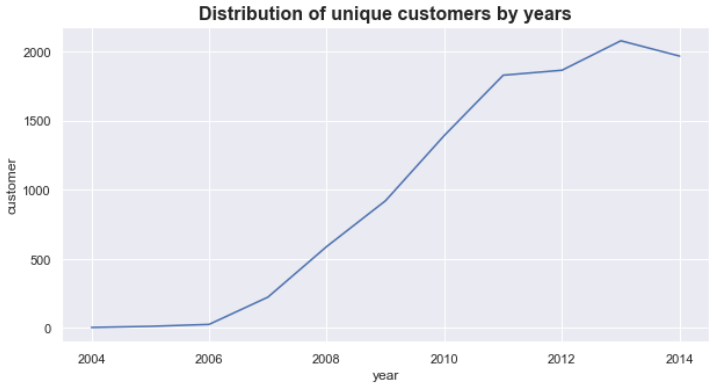
3.2 Features

3.2.1 "Year" Feature



The above figure is the graphical illustration of the percentage distribution of each rating class over time. As indicated in the figure, the percentage of good ratings to total reviews is significantly higher than the percentage of neutral and bad ratings. In 2013, the difference between good reviews and bad or neutral reviews becomes much larger. In terms of fluctuating lines in the figure, the percentages of neutral and good ratings to total reviews are equal in 2004, but while the percentage of good ratings significantly increases after 2004 up until 2007, the percentage of neutral ratings sharply declined one year after 2004. From 2005 to 2014, The level of percentage of neutral ratings remains stable without any sharp increase or decrease. But the orange line representing the percentage of good ratings to total reviews fluctuates with some small increases and decreases.

3.2.2 “Customer” Feature



The figure on the left reports the number of customers making purchase of health and personal care products per year. It demonstrates that the number of customers sharply increased after 2006, and that increase lasted until 2011. After 2011, the rate of increase remains stable, but a slight decline is reported in the year of 2014.

rating_class	review_text	number_of_customers
bad	4625	1510
neutral	8283	2183
good	41879	1900

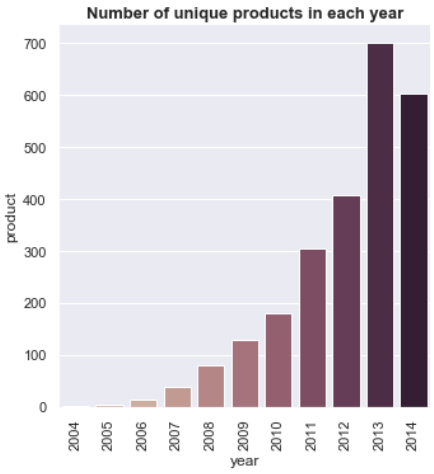
The table on the left reports the number of customers per rating class. Given this table, it is possible to say that there are more customers writing neutral ratings, but the highest number of reviews has been written. by customers writing good reviews. The numnber of good reviews written by 1900 customers is much higher than the number of neutral (8283) and the number of bad (4625) reviews.

In the table on the right, we see the first 10 customers who write most reviews and buy product have been reported with their customer id. It demonstrates that customers whose id are A3NHUQ33CFH3VM and A1UQBFCERIP7VJ are the customers who wrote most reviews and bought products.

customer	review_text	product
A3NHUQ33CFH3VM	235	235
A1UQBFCERIP7VJ	235	235
A2OCDK0BOW6UCY	198	198
ALNFHVS3SC4FV	196	196
A2P739KOM4U5JB	182	182
A3094EPI56GKZ6	181	181
A34BZM6S9L7QI4	170	170
AEL6CQNQXONBX	170	170
ALQ4USPEQ9L5N	158	158
A2ULQOGN59LDNK	155	155

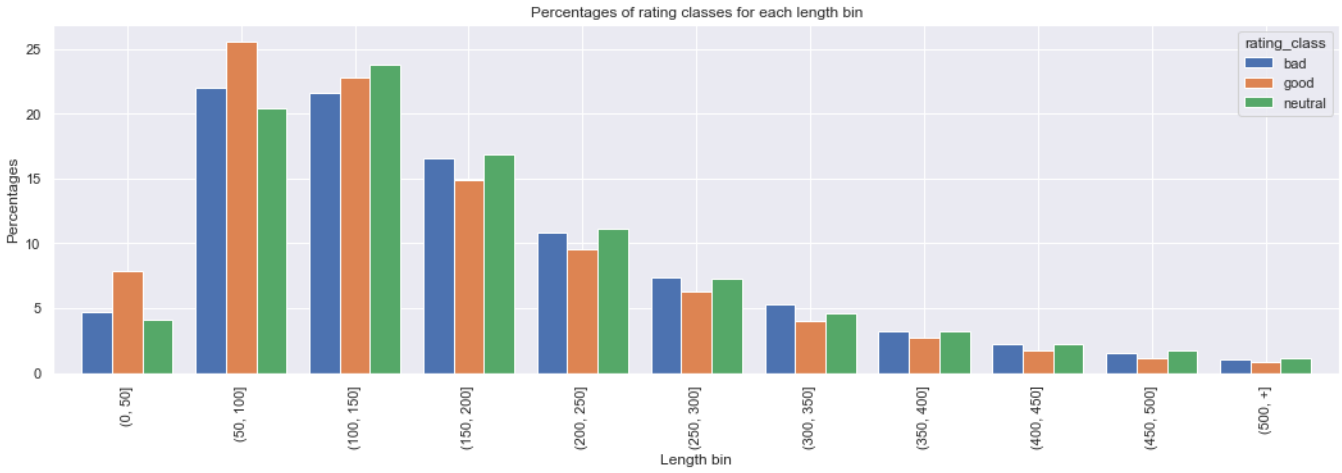
The figure on the left illustrates the number of reviews written by the first 10 customers who write most reviews and make purchases. According to the figure, the customer whose id is A1UQBFCERIP7VJ has written the highest number of reviews. The customer with A2P739KOM4U5JB id has written the highest number of bad reviews, and the customer with A2P739KOM4U5JB has written the highest number of neutral reviews.

3.2.3 “Product” Feature



The figure on the right, represents the distribution of number of products over time. As shown above, the number of products increases over time. The rate of increase is especially larger when it comes to the year of 2013, which is why the number of total reviews significantly increased in this year. In similar, the number of products decreases in 2014 so is the number of total reviews.

In the figure on the left, we see the distribution of each rating class for the 10 products with most reviews. According to the figure, the product with B0037KMI0U id has the highest number of good reviews; the product with B003X5FYJQ id has the highest number of neutral reviews, and the product with B003X5FYJQ has the highest number of bad reviews.



3.2.4 “Review Length” Feature

As shown in the above table, the highest percentage of good reviews lies in the word range of 50-100. The lowest percentage of good reviews falls in the range of 500+ words. In terms of bad reviews, the highest percentage falls in the range of 50-100 words, and the lowest percentage lies in the range of 500+ words. So, in both good and bad ratings, the highest and lowest percentiles are in the same word ranges.

3.2.5 “Text Review” Feature

Bad Rating Words:

Bad Rating Words: Words	Avg.	Words	Avg.	Words	Avg.	Words	Avg.
disappointment	0.461538	yuck	0.304965	fail	0.259259	misleading	0.235849
nope	0.413223	poor	0.301230	joke	0.257862	homeopathic	0.235294
ineffective	0.402878	useless	0.290816	poise	0.257426	gag	0.233533
disappointing	0.402542	repeatedly	0.287037	disgusting	0.255682	grainy	0.233333
sorry	0.369767	horrible	0.286573	tasteless	0.255639	ouch	0.232394
garbage	0.361842	nowhere	0.276786	sensa	0.253012	trash	0.230483
disappointed	0.350877	nauseous	0.275862	waste	0.250000	slimy	0.229730
overprice	0.340741	thinkthin	0.273885	unhealthy	0.250000	fake	0.227979
sadly	0.336406	unfortunately	0.270191	terrible	0.248077		
poorly	0.323864	shame	0.266667	sip	0.243697		
gauze	0.322368	sync	0.266187	junk	0.238532		
spit	0.317568	ridiculous	0.264901	worse	0.238532		
mediocre	0.311927	awful	0.262115	fructose	0.238411		
bland	0.311594	discontinue	0.261194	frustrating	0.238372		

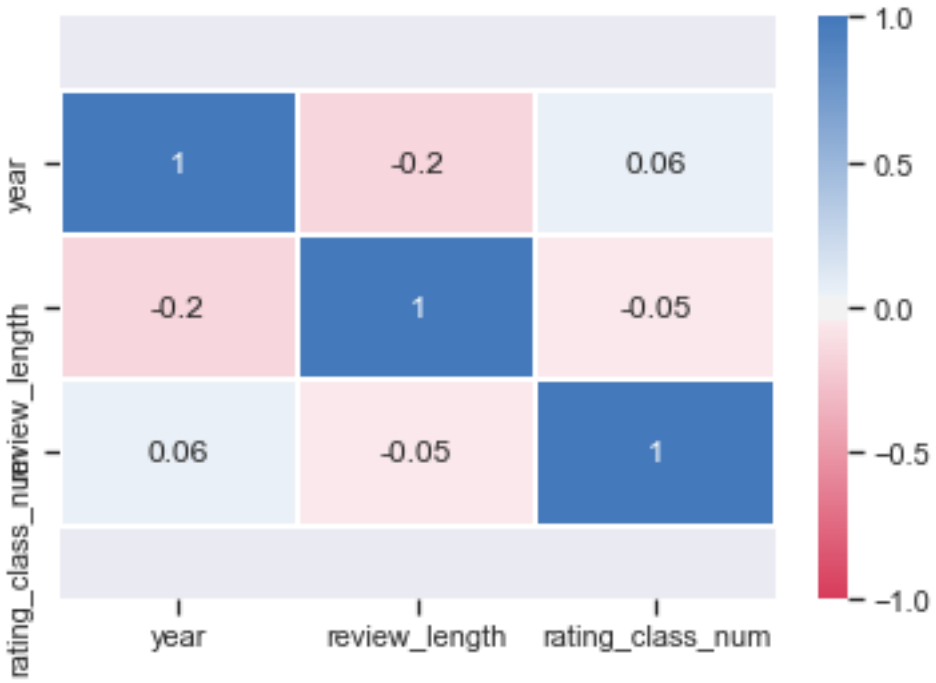
The most common 70 words in bad reviews are shown in the table above. Each of these words indicate the customers' impression on the products. For example, "ineffective" word suggests that the customer thinks that the product is not effective in cleaning or protecting the customer's health. According to the table, it is possible to say that the word "disappointment" has been most used by the customers in their bad reviews, suggesting that those customers found that the products they purchased fail to clean or protect the customers' health.

Good Rating Words:

Good Rating Words: Words	Avg.	Words	Avg.	Words	Avg.	Words	Avg.
superb	0.962963	insurance	0.92	natto	0.906542	wonderful	0.896032
delighted	0.947368	amazed	0.918455	mr	0.906404	strengthen	0.894231
terrific	0.945946	inflammatory	0.914754	olive	0.905941	fantastic	0.894085
pleasantly	0.944732	awesome	0.914465	keeper	0.903846	platinum	0.893805
alkaline	0.941667	eneloop	0.91411	hygienist	0.903846	formulas	0.893617
expiration	0.938907	shellfish	0.914013	virgin	0.903509	cleanse	0.893258
highly	0.936758	affordable	0.913636	subscribe	0.902813	sweater	0.891667
quadrant	0.931915	ubiquinol	0.913462	curcumin	0.901860	winner	0.891304
excellent	0.931778	recommende	0.913386	vital	0.900901		
mk	0.927419	aging	0.91195	powerball	0.9		
pleased	0.925808	barber	0.91129	saver	0.898649		
amazingly	0.924658	purity	0.910569	goodsense	0.898305		
recommended	0.92446	thistle	0.907143	coq	0.897849		
camera	0.92053	lime	0.90678	peace	0.897436		

In addition to the bad reviews, I also look at the most common 70 words written in good reviews. These words indicate the customers' appreciation of the products they have purchased. According to the table, the "superb", "delighted" and "terrific" are three most common words used by the customers in their reviews, suggesting that those customers have a very positive impression on the products, and greatly appreciate them.

3.2.6. Correlation Between Features



Features with numeric data type are 'year','review_length', and 'rating_class_num'. The correlation matrix above includes those features .The above heatmap illustrates the relationships between the variables. While the red color indicates a negative correlation between the two variables, blue color indicates positive correlation. The magnitudes of correlations have also been shown on the heatmap.

There is no strong correlation between two numeric variable.

4. NATURAL LANGUAGE PROCESSING

4.1 Feature Engineering and Selection

Numerical values are included in machine learning models. My dataset, however, include a list of sentences. Therefore, I need to represent the data in a way I can make the data understand these sentences as a list of numbers. It is also important for my algorithm to extract patterns from the data.

For this purpose, I have used CounterVectorizer, TF-IDF, Hashing Vectorizer, and included the words that are most commonly used in the text into the stop word list and used SMOTE techniques into our classification models.

4.2 Data Preprocessing

Separate response variable and features:

I have separated features(clean text) and response variable(rating class). Features are named as X, and response variable is named as y.

Split the dataset into the Training set and Test set:

After that, I have divided the data into the training and test sets. 20% of dataset will be my testing set and 80% of the dataset will be the training set. To make sure to have consistent results, I also set the random state of the split.

4.3 Choosing the Right Evaluation Metric

Because I have a data imbalance in my case, I need to perform the evaluation of the classifier performance by using sufficient metrics to take the class distribution into account and to give a closer look to the minority class. Given that, I have used f1 score that is harmonic average of precision and recall it as evaluation metric.

It is important to understand what types of errors in our model. In order to visualize the types of potential errors in the model, I use Confusion Matrix. It compares the predictions of my model with the true label. So, I use this matrix along with evaluation matrix, f1 score.

4.4 Modeling

This problem is a supervised binary classification problem. My analysis aims to predict the sentiment toward the product based on the reviews written by customer who bought health and personal care products in Amazon. I used Python's Scikit Learn libraries to solve the problem. In this case, I performed Logistic Regression, Random Forest and Naive Bayes algorithms .

Because the ratings on the reviews were not normally distributed, I decreased rating categories from 5 to 2 by merging Rating 1 and 2 and rename as 'Bad' and merging Rating 4 and 5 and rename as 'Good'. I dropped Rating 3 from the dataset due to the reason I mentioned above.

For feature selection purposes, I applied threshold for word occurrence by using min_df/max_df. For feature engineering purposes, I applied CountVectorizer, TF-IDF, and Hashing Vectorizer to the text data so I can convert a collection of text documents into numerical feature vectors.

Before I start modeling, I looked at the dummy classifier f1 score. This classifier is useful to set a

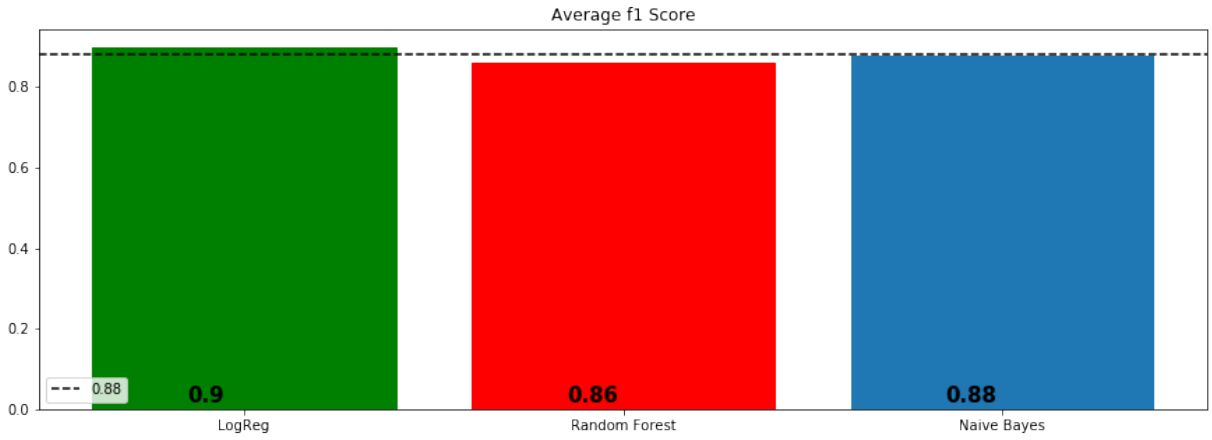
simple baseline to compare it with other real classifiers. It is equal to 0.82. It means that random selection will produce this score.

4.4.1 Count Vectorizer

Vectorization is a general process of converting a collection of text documents into numerical feature vectors. This particular strategy (tokenization, counting and normalization) is called the **Bag of Words** or "Bag of n-grams" representation. Word occurrences describe documents while disregarding the relative position information of the words in the document. **"CountVectorizer"** performs both tokenization and occurrence counting in a single class.

			precision	recall	f1-score	support
model	accuracy	class				
LogReg	0.888937	bad	0.460584	0.682162	0.549891	925.0
		good	0.962930	0.911772	0.936653	8376.0
		average	0.912971	0.888937	0.898189	9301.0
Random Forest	0.903129	bad	0.961538	0.027027	0.052576	925.0
		good	0.902965	0.999881	0.948955	8376.0
		average	0.908790	0.903129	0.859808	9301.0
Naive Bayes	0.861628	bad	0.385152	0.656216	0.485406	925.0
		good	0.958835	0.884312	0.920067	8376.0
		average	0.901781	0.861628	0.876839	9301.0

Logistic Regression is the winner with 0.898189 score.

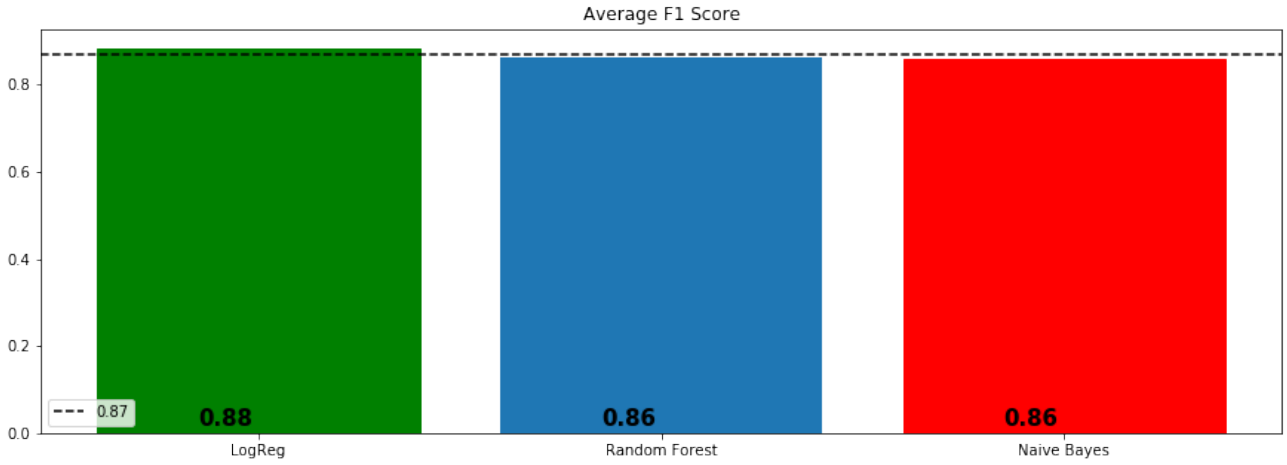


4.4.2 TF-IDF Vectorizer

In order for our model to focus more on meaningful words, I would use a TF-IDF score (Term Frequency, Inverse Document Frequency) on top of our Bag of Words model. TF-IDF weighs words based on how rare they are in the data, and it discounts the value of the words that are too frequent. TF-IDF works by diminishing the value of these common words by assigning lower weights to them while putting more emphasis on the words which appear in a subset of a specific document.

			precision	recall	f1-score	support
model	accuracy	class				
LogReg	0.864208	bad	0.409042	0.821622	0.546173	925.0
		good	0.977832	0.868911	0.920159	8376.0
		average	0.921264	0.864208	0.882966	9301.0
Random Forest	0.903881	bad	1.000000	0.033514	0.064854	925.0
		good	0.903560	1.000000	0.949337	8376.0
		average	0.913151	0.903881	0.861374	9301.0
Naive Bayes	0.903021	bad	0.870968	0.029189	0.056485	925.0
		good	0.903128	0.999522	0.948884	8376.0
		average	0.899930	0.903021	0.860133	9301.0

Logistic Regression is the winner with 0.882966 score.

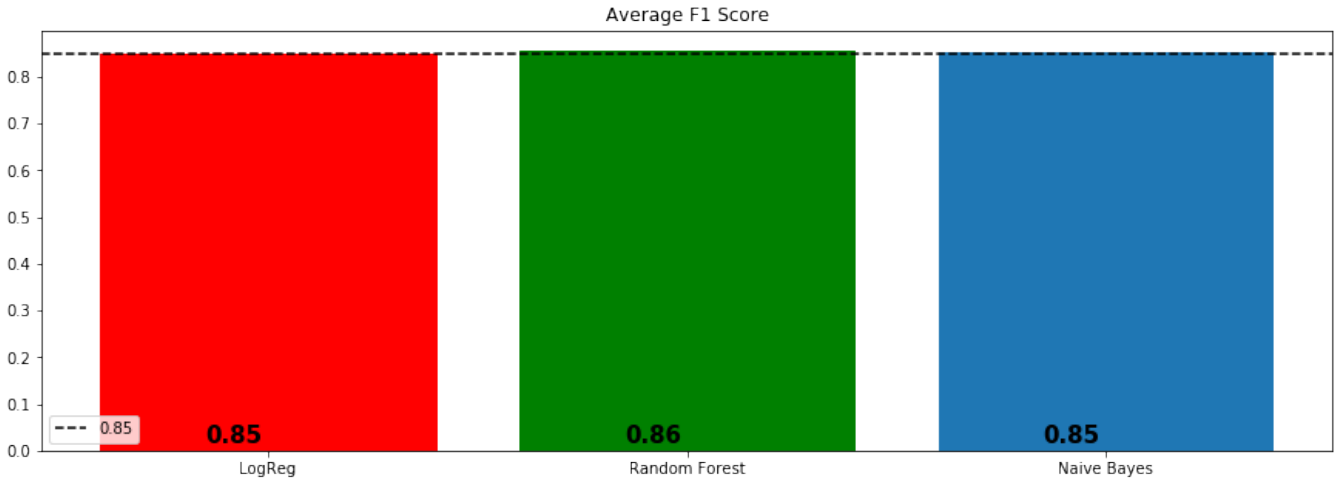


4.4.3 Hash Vectorizer

Hash Vectorizer is designed to be as memory efficient as possible. Rather than storing the tokens as strings, the vectorizer applies the hashing trick to encode them as numerical indexes. The disadvantage of this method is that once they are vectorized, the features' names can't be retrieved.

			precision	recall	f1-score	support
model	**accuracy**	**class**				
LogReg	0.821095	bad	0.333933	0.803243	0.471746	925.0
		good	0.974279	0.823066	0.892312	8376.0
		average	0.910596	0.821095	0.850486	9301.0
Random Forest	0.901731	bad	1.000000	0.011892	0.023504	925.0
		good	0.901615	1.000000	0.948262	8376.0
		average	0.911399	0.901731	0.856293	9301.0
Naive Bayes	0.900548	bad	0.000000	0.000000	0.000000	925.0
		good	0.900548	1.000000	0.947672	8376.0
		average	0.810987	0.900548	0.853425	9301.0

Random Forest is the winner with 0.856293 score.

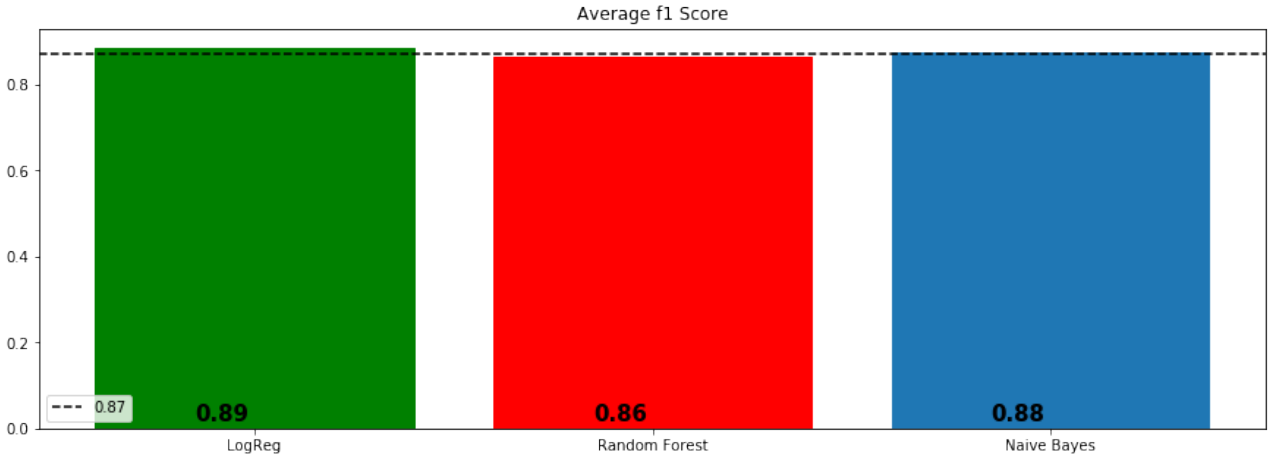


4.4.4 Adding Most Common and Lest Common Words to Stopwords List (Count Vectorizer)

Because there were not too many distinguisher words in different categories, the most and least common 70 words added to the list of stopwords and models were performed to see any changes in evaluation metrics.

			precision	recall	f1-score	support
model	accuracy	class				
LogReg	0.873669	bad	0.412096	0.633514	0.499361	925.0
		good	0.956974	0.900191	0.927715	8376.0
		average	0.902785	0.873669	0.885114	9301.0
Random Forest	0.903559	bad	0.750000	0.045405	0.085627	925.0
		good	0.904489	0.998329	0.949095	8376.0
		average	0.889125	0.903559	0.863222	9301.0
Naive Bayes	0.863348	bad	0.380854	0.597838	0.465292	925.0
		good	0.952605	0.892670	0.921664	8376.0
		average	0.895744	0.863348	0.876277	9301.0

Logistic Regression is the winner with 0.885114 score.



The performance of the models have not been impacted by adding most and least common words.

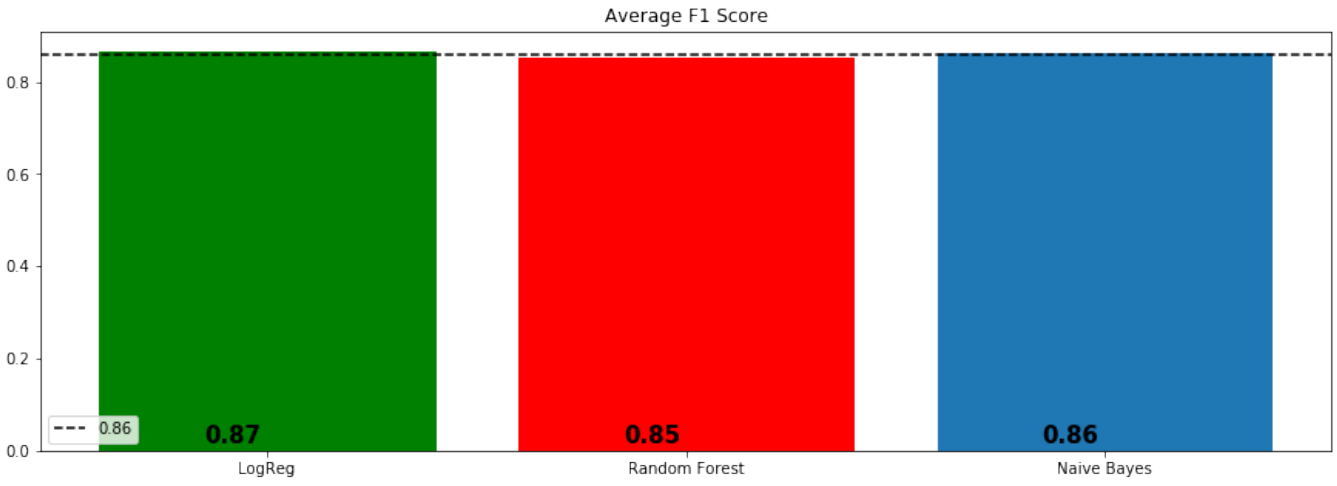
4.4.5 Synthetic Minority Oversampling Technique (SMOTE)

Because there is an imbalance in the values within the target variable, I will perform the SMOTE method in the dealing with this skewed value issue so I can see whether I may be able to improve our accuracy score.

After SMOTE mechanism to mitigate target class imbalance and identifying best hyper-parameters, f1 score of my model did not show an enhancement and decreased to 0.866074.

			precision	recall	f1-score	support
model	accuracy	class				
LogReg	0.851736	bad	0.345368	0.548108	0.423736	925.0
		good	0.946636	0.885267	0.914924	8376.0
		average	0.886839	0.851736	0.866074	9301.0
Random Forest	0.863886	bad	0.234009	0.162162	0.191571	925.0
		good	0.910508	0.941380	0.925687	8376.0
		average	0.843229	0.863886	0.852678	9301.0
Naive Bayes	0.848188	bad	0.327915	0.501622	0.396581	925.0
		good	0.941542	0.886461	0.913172	8376.0
		average	0.880516	0.848188	0.861796	9301.0

Logistic Regression is the winner with 0.866074 score.



5. CONCLUSION

In this project, I aimed to predict the rating scores given the reviews written by the customers. It is clearly demonstrated that data set preparation and feature engineering are as equally important as the model creation. I performed;

- Count Vector, TF-IDF, Hashing Vector
- Classification Models
- Adding most and least common words to CountVect,
- SMOTE,

I will use Logistic Regression with Count Vectorizing in deploying section (f1 score is 0.896189). Including most and least common words in the stopword list didn't have a significant effect on the performance of the models' performance. Resampling technique and linear dimensionality reduction did not positively improve the model accuracy. However, it decreased the model performance.

5.1 Recommendations

I would recommend the client to use the model as it is and provide me to get better scores to implement Logistic Regression with with count vectorizer. By the way, the size of the data and prediction time are also important to be considered. For example, neural network algorithms may not be able to outperform with the low size data sets.

I would recommend the client to provide a system where a user can write their reviews without mistakes. For example, that type of system should not accept the review if it does not meet the word/ grammar correctness criterion.

I would recommend the client to encourage users to share their experience with products by providing incentives to them.

I would recommend the client to use the reviews as a feedback mechanism, take actions towards them, and let the customers know about whether the problems reported in the reviews have been fixed.

Controversial Cases

The controversial case such as "I was expecting better - negative meaning" or "it was better than my expectation - positive meaning " can be handled in the modelling section by using deep learning technique (Keras with Word2Vec).

5.2 Future Study

As a future study, I might focus on the following topics:

* Implementation of Dask library for parallel processing to reduce run time.
* After reducing run time, focusing on hyperparameter tuning more.
* Implementation of Deep Learning with different neural network types and different layer combinations to get better results.
* Implementation of Principal Component Analysis maybe to improve result.