REPORT-CAPSONE PROJECT-2

1. PROBLEM DEFINITION

Multi-class classification on Women E-Trade Comments Problem.

E-commerce is the activity of buying or selling of products on online services or over the Internet. Electronic commerce draws on technologies such as mobile commerce, electronic funds transfer, supply chain management, internet marketing, online transaction processing, electronic data interchange (EDI), inventory management systems, and automated data collection systems.

Modern e-commerce typically uses the World Wide Web for at least one part of the transaction's life cycle although it may also use other technologies such as e-mail. The largest Internet retailer in the world as measured by revenue and market capitalization is Amazon.com, Inc. which is an American e-commerce and cloud computing company. And it became the fourth most valuable public company in the world (behind only Apple, Alphabet, and Microsoft), the largest Internet company by revenue in the world, and after Walmart, the second largest employer in the United States.

But **Client:**

Ecommerce traders whose trade mostly depends on the reviews and ratings.

a. Data Set:

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. Because this is real commercial data, it has been anonymized, and references to the company in the review text and body have been replaced with "retailer".

This dataset consists of 23486 rows and 10 features. Each row corresponds to a customer review, and includes the variables:

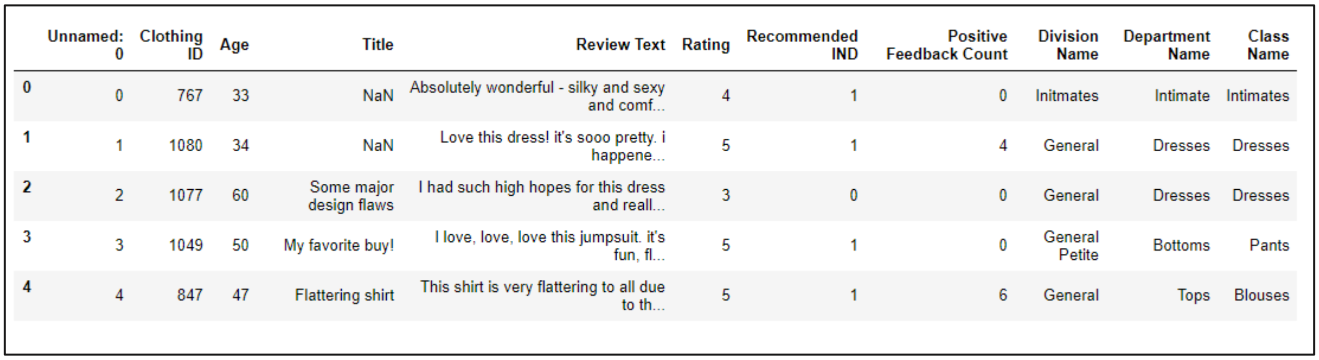
- Clothing ID: Integer Categorical variable that refers to the specific piece being reviewed.
- Age: Positive Integer variable of the reviewers age.
- Title: String variable for the title of the review.
- Review Text: String variable for the review body.
- Rating: Positive Ordinal Integer variable for 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.

https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews/home

2. DATA WRANGLING

a. Data Set Basic Formatting:

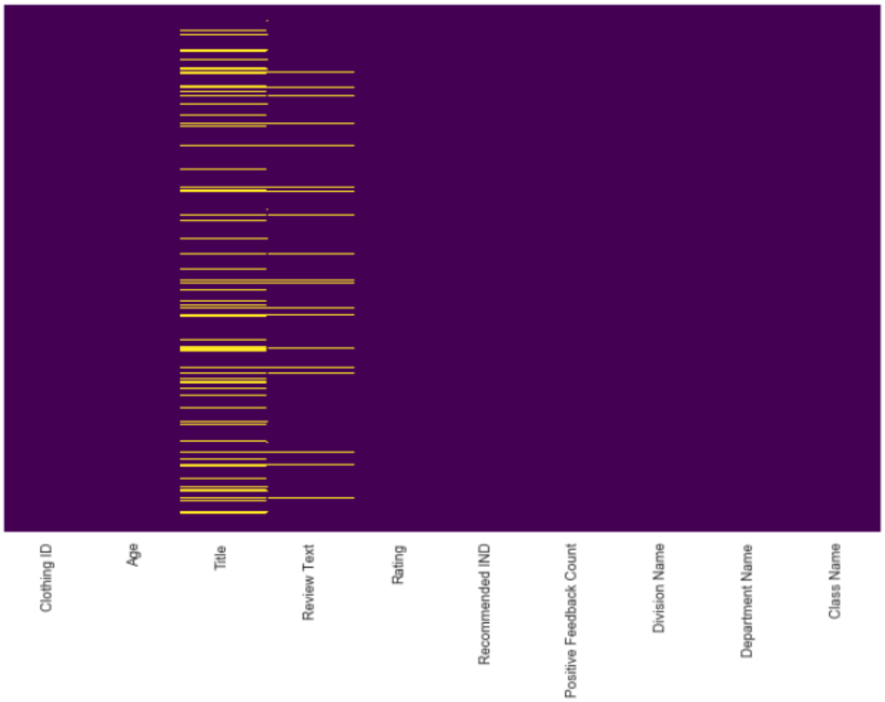
There was no basic formatting on the data set as seen below. The feature names and the whitespace between the word of the names was filled with '_'.



	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

b. Missing Values:

As seen below, mostly null values are cumulated under Title and Review features. Since the texts of Title feature will give us very helpful ideas we merged these two features and created a new one named as 'new_text' which both of them have not null values.



c. Cleaning the new_text Feature:

After creating the merged feature, in the context of corpus normalization we applied advanced text cleaning such as:

[1] Lowercase the text

[2] Keep only words

[3] Find URLs

[4] Remove links from posts

[5] Expending contractions

[6] Removing whitespace

[7] Remove apostrophe signs

[8] Remove stop words and stemming

d. Creating a new column consists of the classification of the ratings:

We have created a new column based on the ratings. Rating 4 and 5 are classified as ‘Good’, Rating 3 as ‘Neutral and Rating 1 and 2 as ‘Bad’.

e. Creating new column consists of the length of the cleaned reviews:

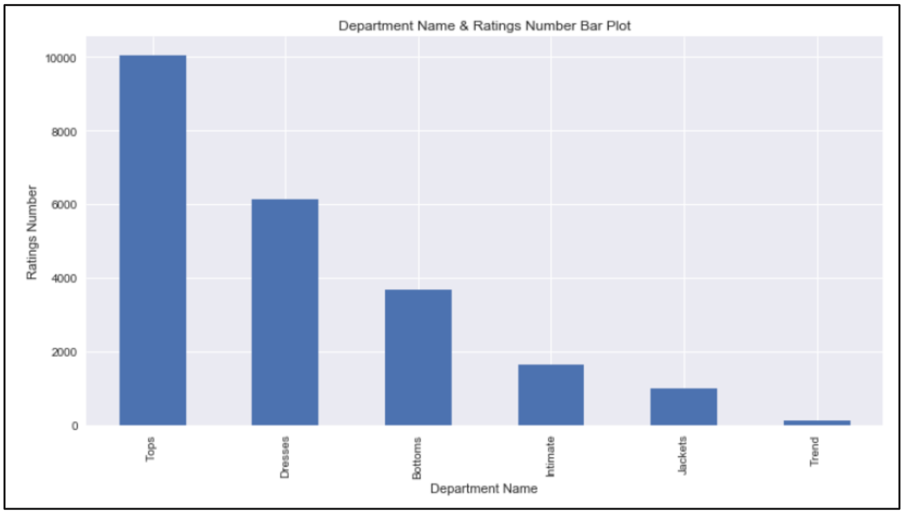
We have created two different columns, first one is the character based the length of the reviews and the second column is the word based by using the tokenizer.

f. Dropping the null values and saving the cleaned data set:

As a final step, we dropped the null values and saved the cleaned data set as 'Cleaned_Women_ECommerce.csv’

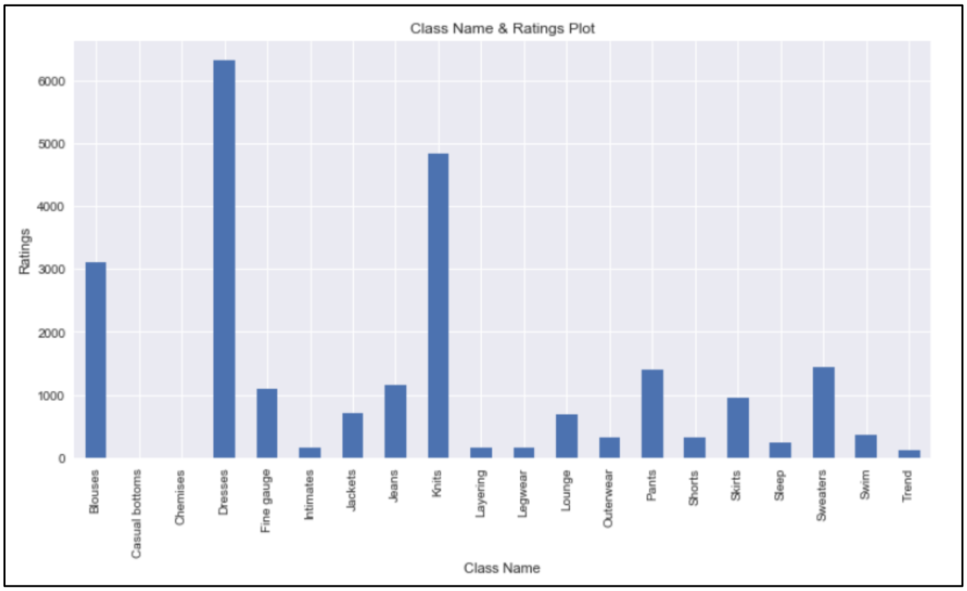
3. EXPLORATORY DATA ANALYSIS (EDA)-DATA VISUALIZATION

a. Department Name and Ratings Number



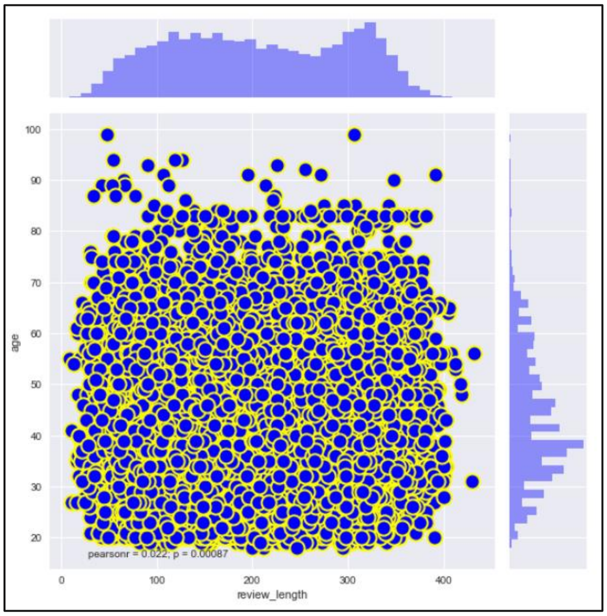
On Department basis, Tops and Dresses are sold mostly. Trend is the weakest sold as seen.

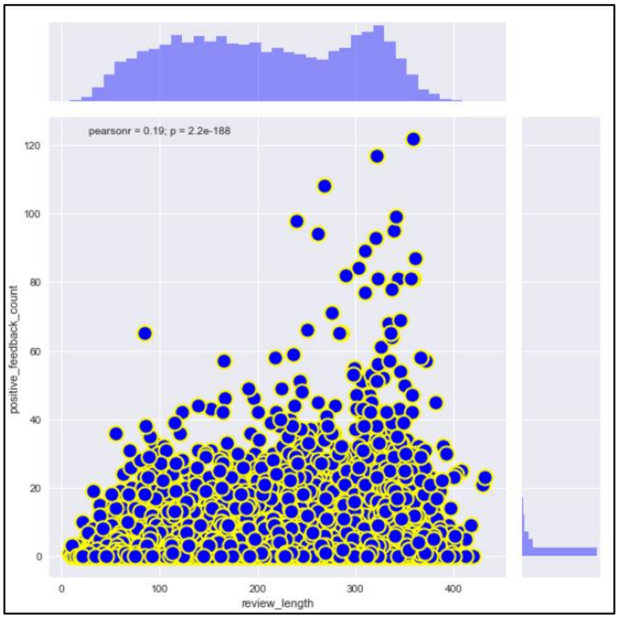
b. Rating Numbers Based on the Class Name



Most ratings were given to Dresses, Knits and Blouses.

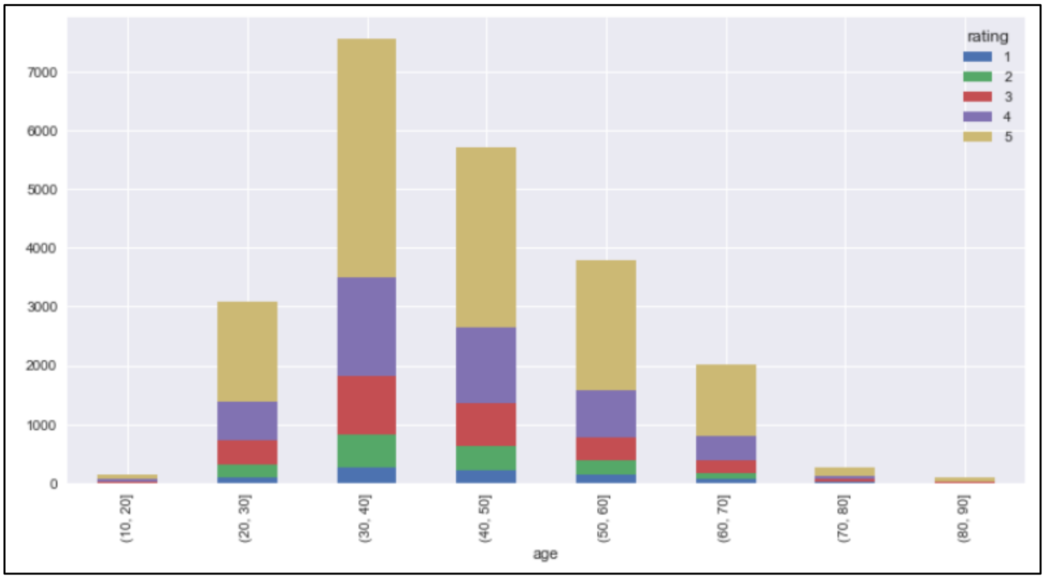
c. Review Length & Age/Positive Feedback Count





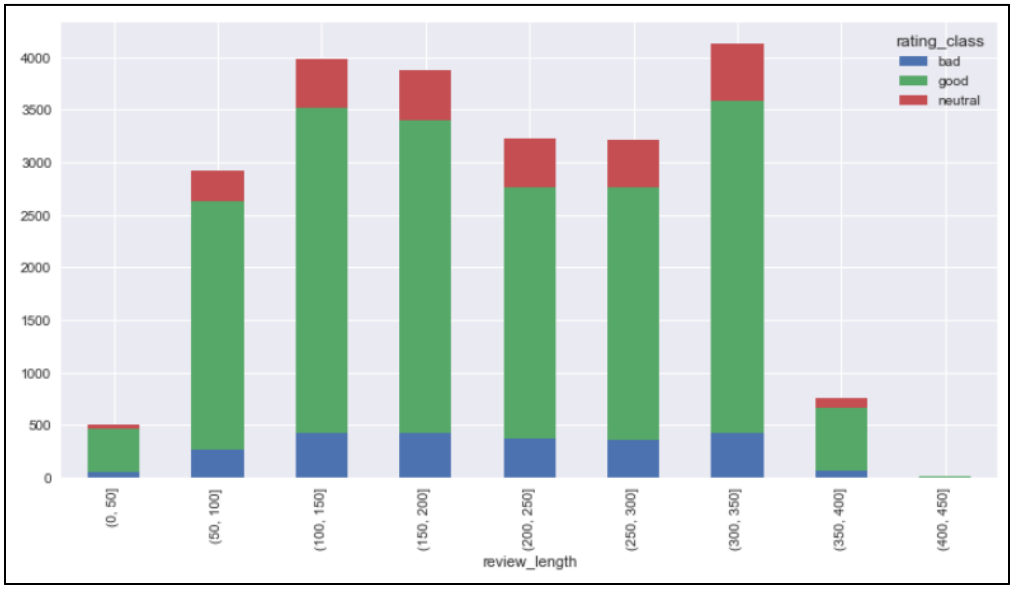
Mostly the customers between 20 and 65 ages left reviews and the review length is between 50 to 350 characters.

d. Age and Ratings



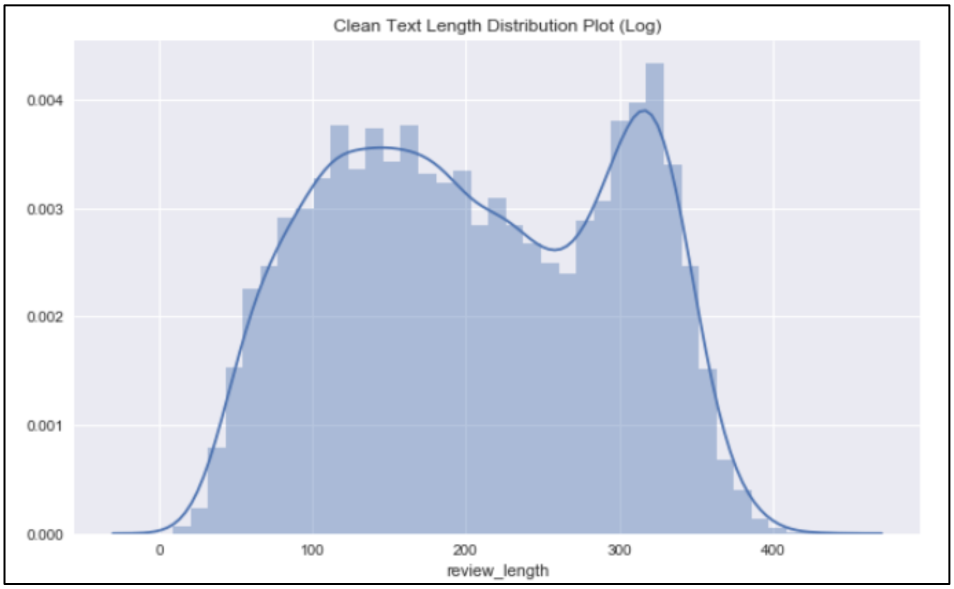
The most satisfied age group is between 30 and 50. Customers mostly left positive reviews.

e. Review Length and Rating Class:

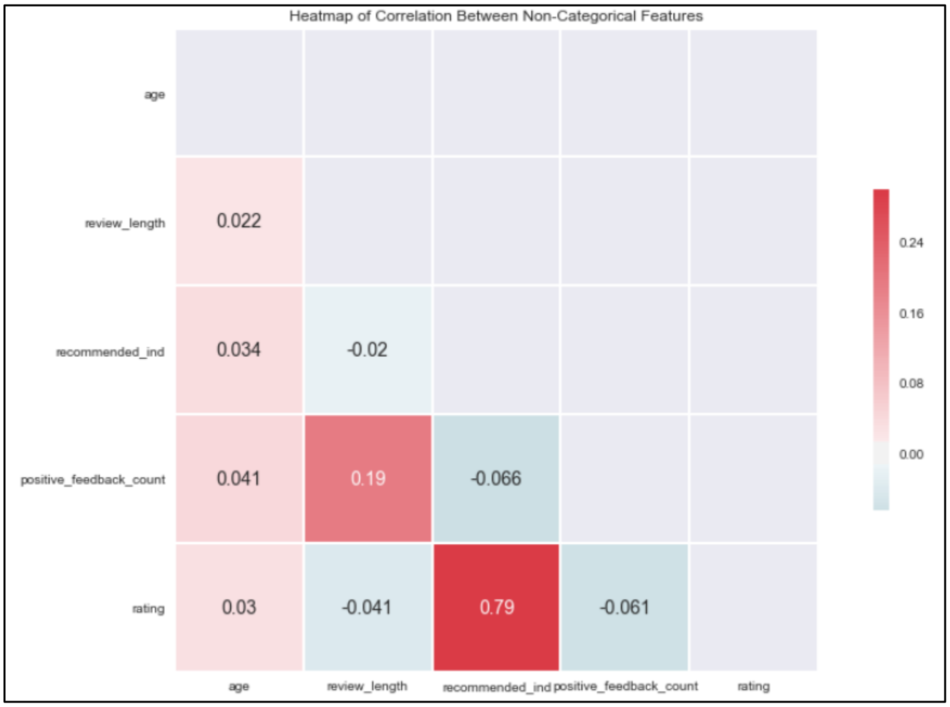


Good reviews have the longest review lengths since Good reviews has the overwhelming majority in our data set.

f. Distribution of Review Length:

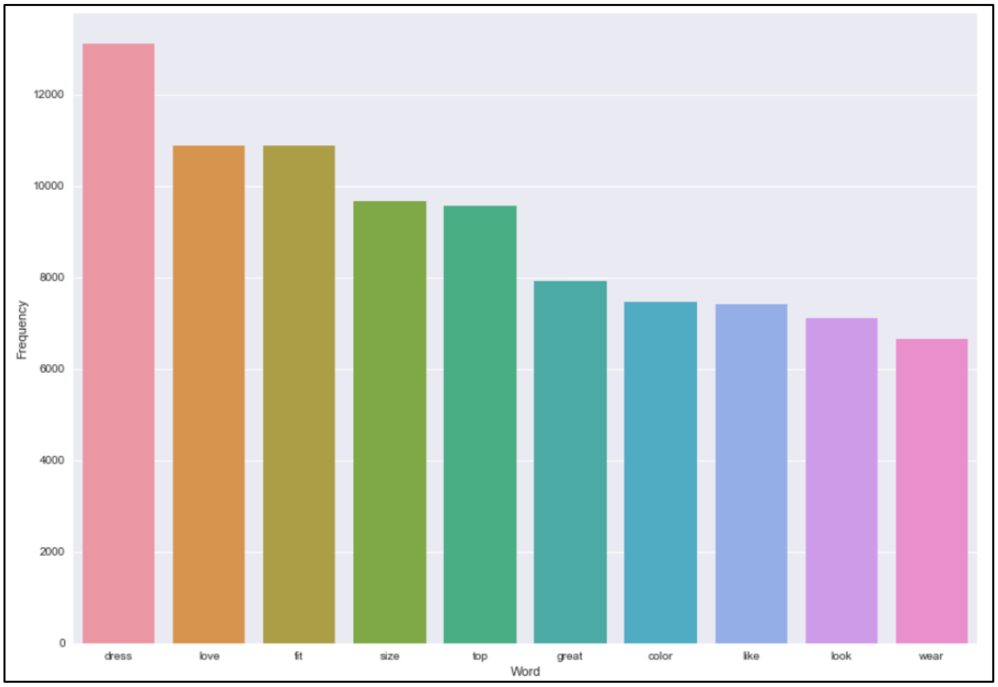


g. Numerical Features Correlation Heatmap:



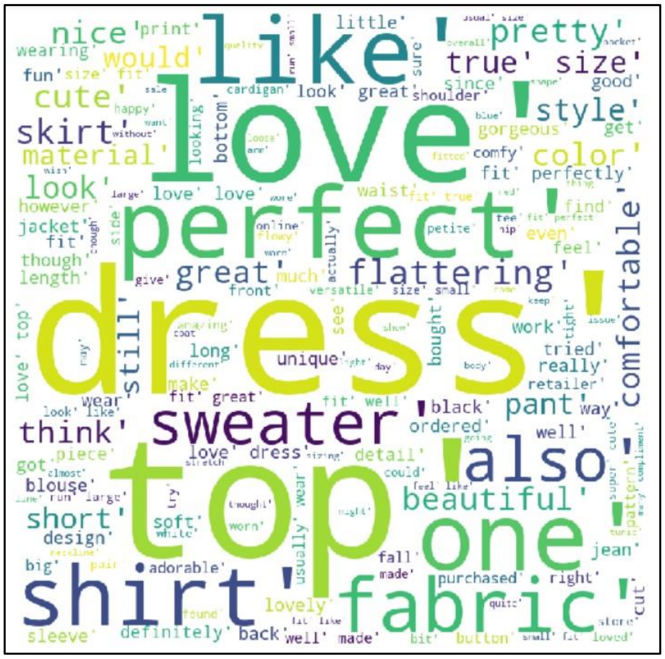
There is a strong correlation between rating and the recommendation. And also a positive correlation between review length and the positive feedback count.

h. Most Common Words:

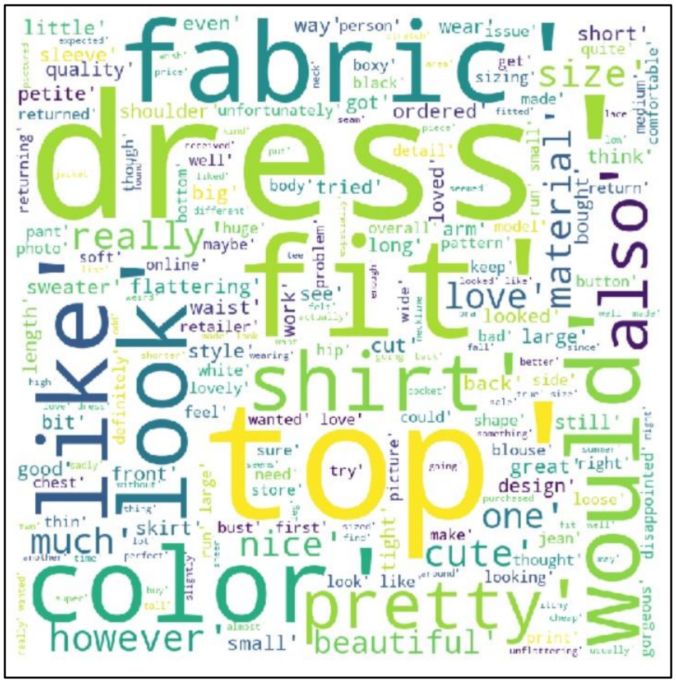


The most common words used are dress, love, fit, size, top etc.

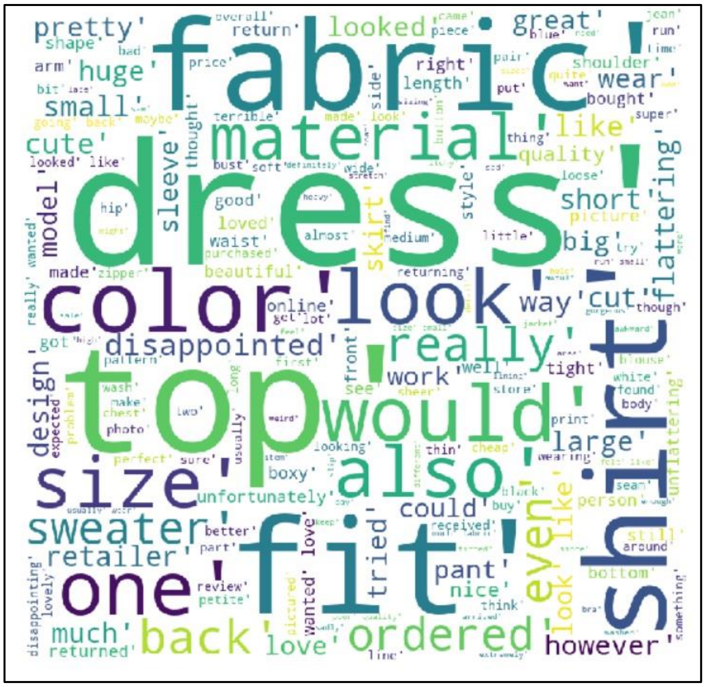
i. Word Cloud Good:



j. Word Cloud Neutral:



k. Word Cloud Bad:

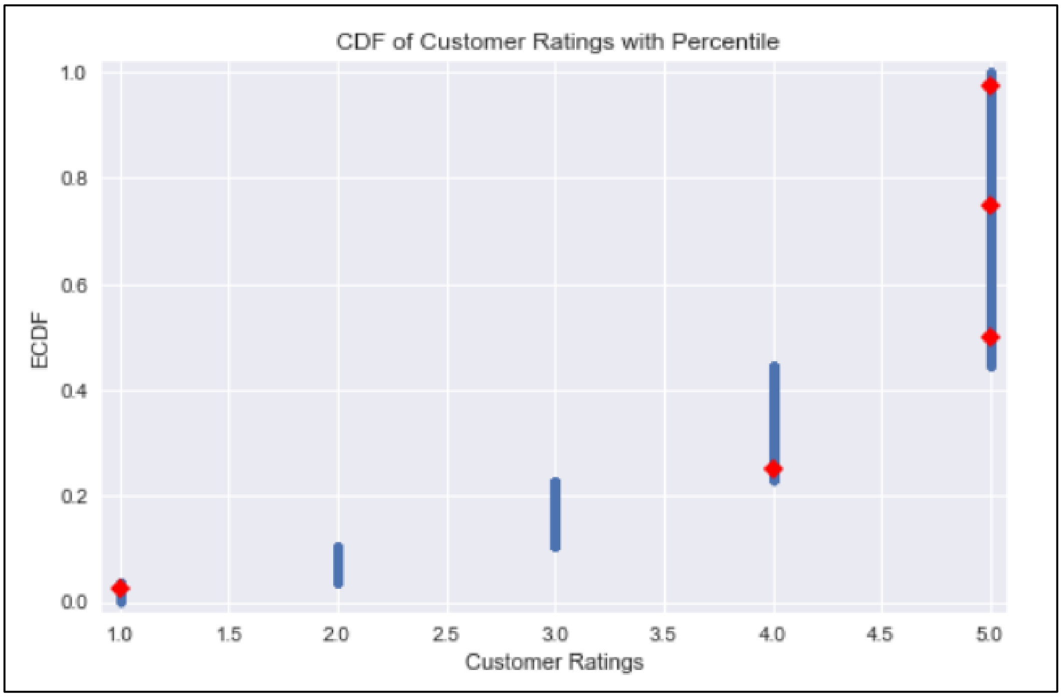


As seen above, word clouds do not have too much characteristic words to distinguish classes. For instance in the bad word could we just started to meet some complaining words such as ‘disappointed, returned, unfortunately, unflattering etc.’ Thus in the second part of the project we expanded the stop words by adding the most common 70 words and the least common 70 words to the stop words list to make the dictionary of the each class more homogeneous.

4. MACHINE LEARNING MODELS WITH 3 RATING CLASSES.

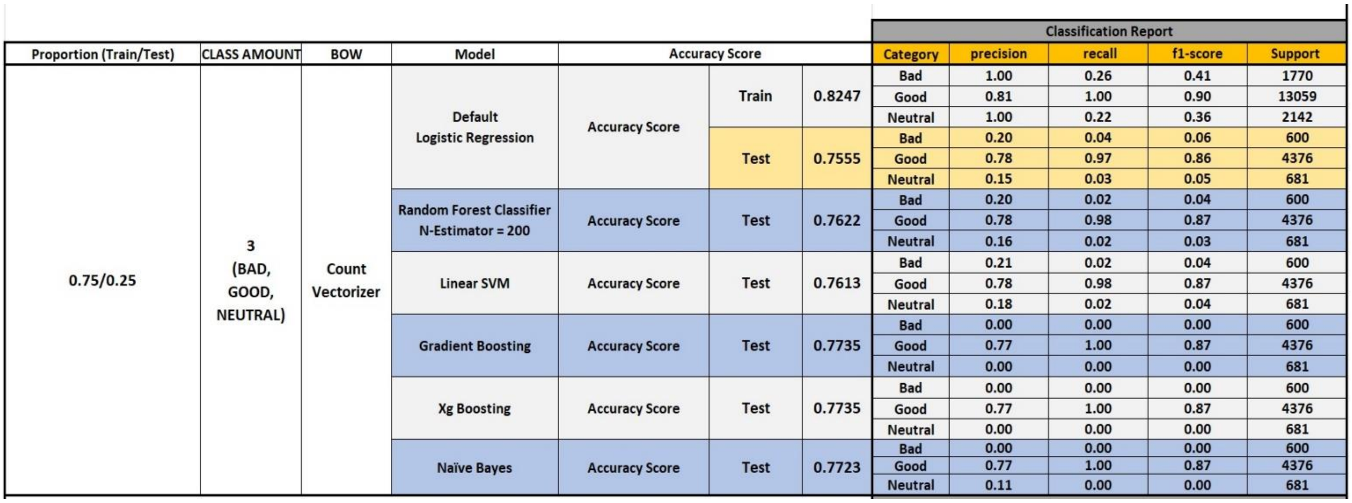
This is a supervised multi-class classification problem. We are trying to predict the ratings based on the reviews left by females who bought clothes via E-Commerce systems. We used Python's Scikit Learn libraries to solve our problem. In this context, we implemented Logistic Regression, Linear SVM and Random Forest, Gradient Boosting, XGBOOST, Naive Bayes, Catboost and TPOT algorithms.

Since the ratings of the reviews was not distributed normally as seen below, we decided to decrease rating classes from 5 to 3 by merging Rating 1 and 2 as 'Bad', Rating 3 as Neutral and Rating 4 and 5 as Good and applied the algorithms. And finally we decreased the rating classes to 2 to check whether our algorithms will do better with binomial classification problem or not. To do so, we merged Rating 1 and 2 as 'Bad' as we did before, but this time Rating 3 joined to 4 and 5 as 'Not Bad' and applied the same models afterwards. We have already applied Count Vectorizing and TF-IDF separately to figure out which one has the better performance. Thus we managed to apply all possible combinations to get the best precision scores for the classes.



Additionally, we used Grid Search method with 5-fold cross validation technique to get rid of overfitting problem. As an evaluation metric we used accuracy.

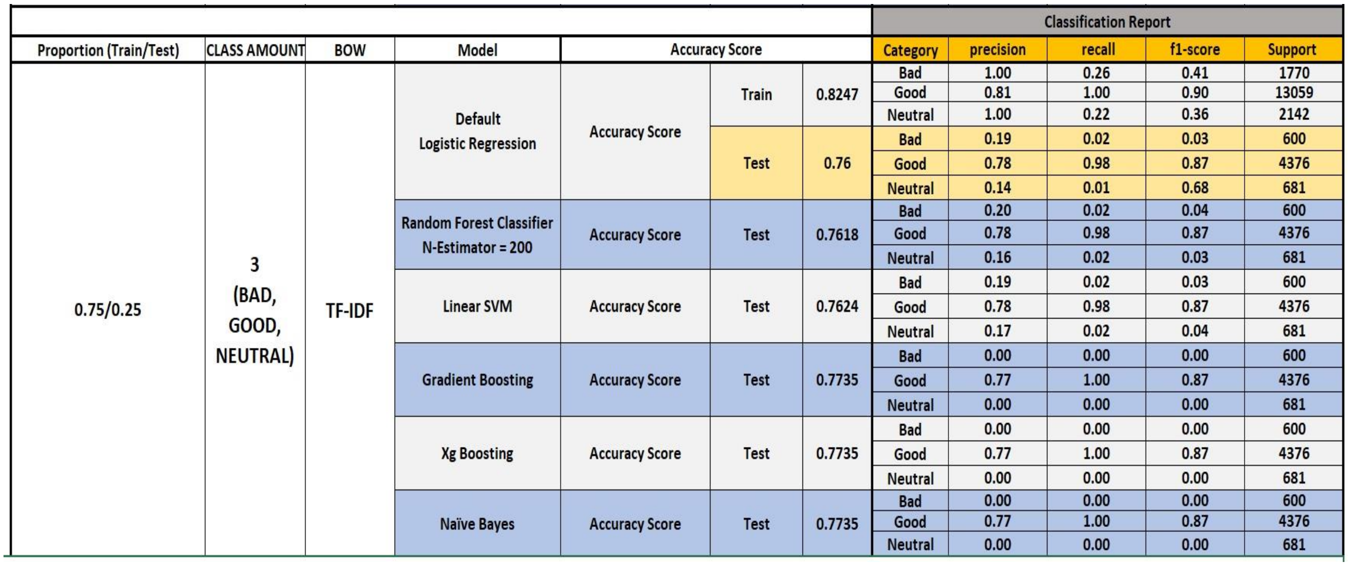
a. 3 Rating Classes, Count-Vectorizing and the Algorithms aforementioned:



							Classification Report				
Proportion (Train/Test)	CLASS AMOUNT	BOW	Model	Accuracy Score			Category	precision	recall	f1-score	Support
0.75/0.25	3 (BAD, GOOD, NEUTRAL)	Count Vectorizer	Default Logistic Regression	Accuracy Score	Train	0.8247	Bad	1.00	0.26	0.41	1770
							Good	0.81	1.00	0.90	13059
							Neutral	1.00	0.22	0.36	2142
					Test	0.7555	Bad	0.20	0.04	0.06	600
							Good	0.78	0.97	0.86	4376
							Neutral	0.15	0.03	0.05	681
			Random Forest Classifier N-Estimator = 200	Accuracy Score	Test	0.7622	Bad	0.20	0.02	0.04	600
							Good	0.78	0.98	0.87	4376
							Neutral	0.16	0.02	0.03	681
			Linear SVM	Accuracy Score	Test	0.7613	Bad	0.21	0.02	0.04	600
							Good	0.78	0.98	0.87	4376
							Neutral	0.18	0.02	0.04	681
			Gradient Boosting	Accuracy Score	Test	0.7735	Bad	0.00	0.00	0.00	600
							Good	0.77	1.00	0.87	4376
							Neutral	0.00	0.00	0.00	681
			Xg Boosting	Accuracy Score	Test	0.7735	Bad	0.00	0.00	0.00	600
							Good	0.77	1.00	0.87	4376
							Neutral	0.00	0.00	0.00	681
			Naïve Bayes	Accuracy Score	Test	0.7723	Bad	0.00	0.00	0.00	600
							Good	0.77	1.00	0.87	4376
							Neutral	0.11	0.00	0.00	681

As regards of Logistic Regression Train Set classification report, Precisions of each classes and Recall of Good class is almost perfect, but we cannot say the same thing for the test set which means that there is an overfitting problem in our model. Random Forest and Linear SVM also gave the similar results while having pretty good precision score for the Good class but slightly less for the Bad and Neutral. Boosting algorithms were the worst ones in this approach since they were not able to predict the minority class explicitly.

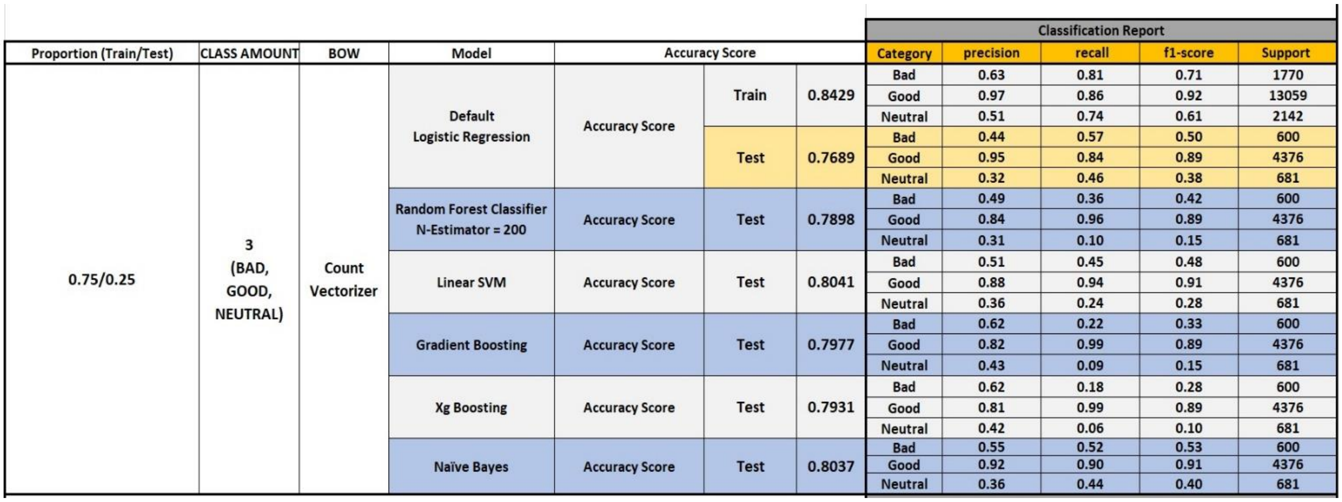
b. 3 Rating Classes, TF-IDF and the Same Algorithms:



							Classification Report				
Proportion (Train/Test)	CLASS AMOUNT	BOW	Model	Accuracy Score			Category	precision	recall	f1-score	Support
0.75/0.25	3 (BAD, GOOD, NEUTRAL)	TF-IDF	Default Logistic Regression	Accuracy Score	Train	0.8247	Bad	1.00	0.26	0.41	1770
							Good	0.81	1.00	0.90	13059
							Neutral	1.00	0.22	0.36	2142
					Test	0.76	Bad	0.19	0.02	0.03	600
							Good	0.78	0.98	0.87	4376
							Neutral	0.14	0.01	0.68	681
			Random Forest Classifier N-Estimator = 200	Accuracy Score	Test	0.7618	Bad	0.20	0.02	0.04	600
							Good	0.78	0.98	0.87	4376
							Neutral	0.16	0.02	0.03	681
			Linear SVM	Accuracy Score	Test	0.7624	Bad	0.19	0.02	0.03	600
							Good	0.78	0.98	0.87	4376
							Neutral	0.17	0.02	0.04	681
			Gradient Boosting	Accuracy Score	Test	0.7735	Bad	0.00	0.00	0.00	600
							Good	0.77	1.00	0.87	4376
							Neutral	0.00	0.00	0.00	681
			Xg Boosting	Accuracy Score	Test	0.7735	Bad	0.00	0.00	0.00	600
							Good	0.77	1.00	0.87	4376
							Neutral	0.00	0.00	0.00	681
			Naïve Bayes	Accuracy Score	Test	0.7735	Bad	0.00	0.00	0.00	600
							Good	0.77	1.00	0.87	4376
							Neutral	0.00	0.00	0.00	681

With the TF-IDF we have got almost the same results.

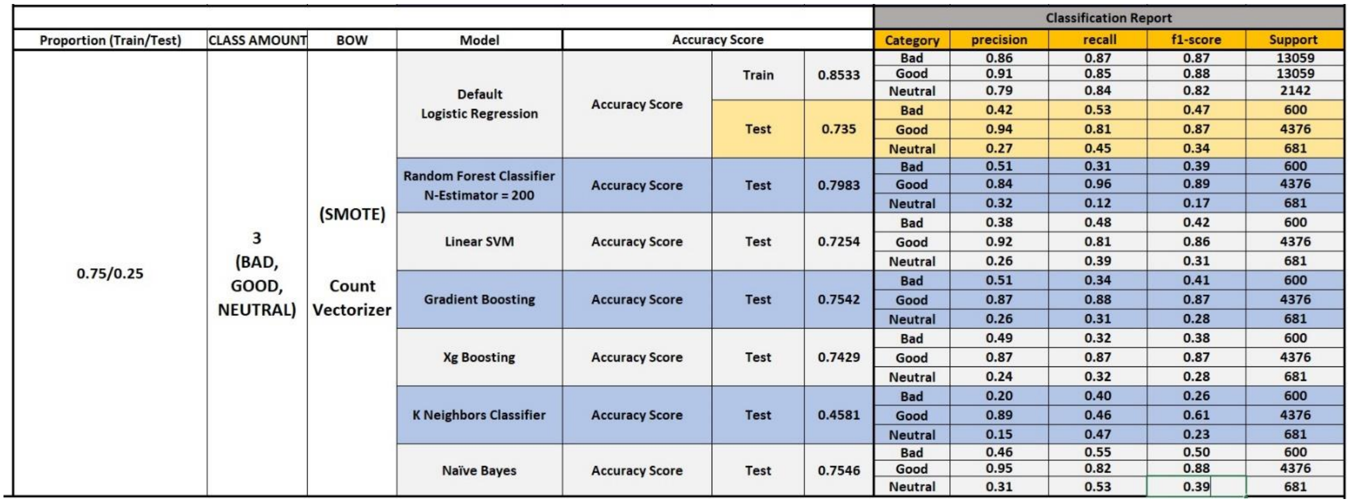
c. Expanded Stop Words, 3 Rating Classes, Count Vectorizer and the Same Algorithms:



							Classification Report				
Proportion (Train/Test)	CLASS AMOUNT	BOW	Model	Accuracy Score			Category	precision	recall	f1-score	Support
0.75/0.25	3 (BAD, GOOD, NEUTRAL)	Count Vectorizer	Default Logistic Regression	Accuracy Score	Train	0.8429	Bad	0.63	0.81	0.71	1770
							Good	0.97	0.86	0.92	13059
							Neutral	0.51	0.74	0.61	2142
					Test	0.7689	Bad	0.44	0.57	0.50	600
							Good	0.95	0.84	0.89	4376
							Neutral	0.32	0.46	0.38	681
			Random Forest Classifier N-Estimator = 200	Accuracy Score	Test	0.7898	Bad	0.49	0.36	0.42	600
							Good	0.84	0.96	0.89	4376
							Neutral	0.31	0.10	0.15	681
			Linear SVM	Accuracy Score	Test	0.8041	Bad	0.51	0.45	0.48	600
							Good	0.88	0.94	0.91	4376
							Neutral	0.36	0.24	0.28	681
			Gradient Boosting	Accuracy Score	Test	0.7977	Bad	0.62	0.22	0.33	600
							Good	0.82	0.99	0.89	4376
							Neutral	0.43	0.09	0.15	681
			Xg Boosting	Accuracy Score	Test	0.7931	Bad	0.62	0.18	0.28	600
							Good	0.81	0.99	0.89	4376
							Neutral	0.42	0.06	0.10	681
			Naïve Bayes	Accuracy Score	Test	0.8037	Bad	0.55	0.52	0.53	600
							Good	0.92	0.90	0.91	4376
							Neutral	0.36	0.44	0.40	681

After expanding the stop words list, the precision of ‘Neutral and Bad’ classes increased up to almost three times from 0.19 to ~0.6 Since there were not a slight difference between Count Vectorizing and TF-IDF, we decided to go with the Count Vectorizer as a Bag of Words method. In this case we got the best accuracy for each class by applying Linear SVM. But for the minority classes Gradient Boosting gave the best scores with 0.62 for Bad, 0.82 for Good and 0.43 for Neutral.

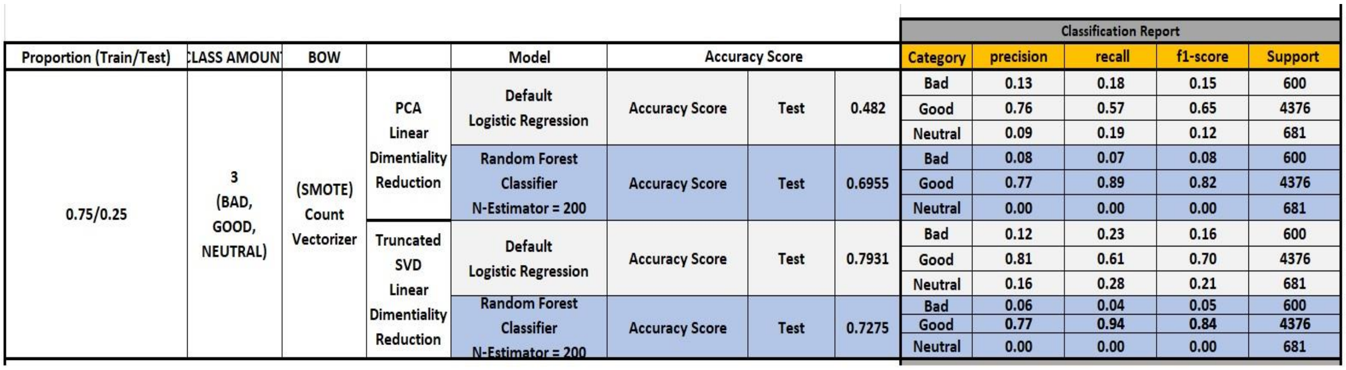
d. Expanded Stop Words, 3 Rating Classes, Count Vectorizer and the Same Algorithms and SMOTE:



							Classification Report				
Proportion (Train/Test)	CLASS AMOUNT	BOW	Model	Accuracy Score			Category	precision	recall	f1-score	Support
0.75/0.25	3 (BAD, GOOD, NEUTRAL)	(SMOTE) Count Vectorizer	Default Logistic Regression	Accuracy Score	Train	0.8533	Bad	0.86	0.87	0.87	13059
							Good	0.91	0.85	0.88	13059
							Neutral	0.79	0.84	0.82	2142
					Test	0.735	Bad	0.42	0.53	0.47	600
							Good	0.94	0.81	0.87	4376
							Neutral	0.27	0.45	0.34	681
			Random Forest Classifier N-Estimator = 200	Accuracy Score	Test	0.7983	Bad	0.51	0.31	0.39	600
							Good	0.84	0.96	0.89	4376
							Neutral	0.32	0.12	0.17	681
			Linear SVM	Accuracy Score	Test	0.7254	Bad	0.38	0.48	0.42	600
							Good	0.92	0.81	0.86	4376
							Neutral	0.26	0.39	0.31	681
			Gradient Boosting	Accuracy Score	Test	0.7542	Bad	0.51	0.34	0.41	600
							Good	0.87	0.88	0.87	4376
							Neutral	0.26	0.31	0.28	681
			Xg Boosting	Accuracy Score	Test	0.7429	Bad	0.49	0.32	0.38	600
							Good	0.87	0.87	0.87	4376
							Neutral	0.24	0.32	0.28	681
			K Neighbors Classifier	Accuracy Score	Test	0.4581	Bad	0.20	0.40	0.26	600
							Good	0.89	0.46	0.61	4376
							Neutral	0.15	0.47	0.23	681
			Naïve Bayes	Accuracy Score	Test	0.7546	Bad	0.46	0.55	0.50	600
							Good	0.95	0.82	0.88	4376
							Neutral	0.31	0.53	0.39	681

Since this data set is an imbalanced data set, we decided to apply resampling by using Syntactic Minority Oversampling Technique(SMOTE) and apply the same models. However there was not a tangible difference between the results as seen above.

e. Expanded Stop Words, 3 Rating Classes, Count Vectorizer, Logistic Regression &Random Forest, SMOTE and Linear Dimensionality Reduction (PCA and Truncated SVD):



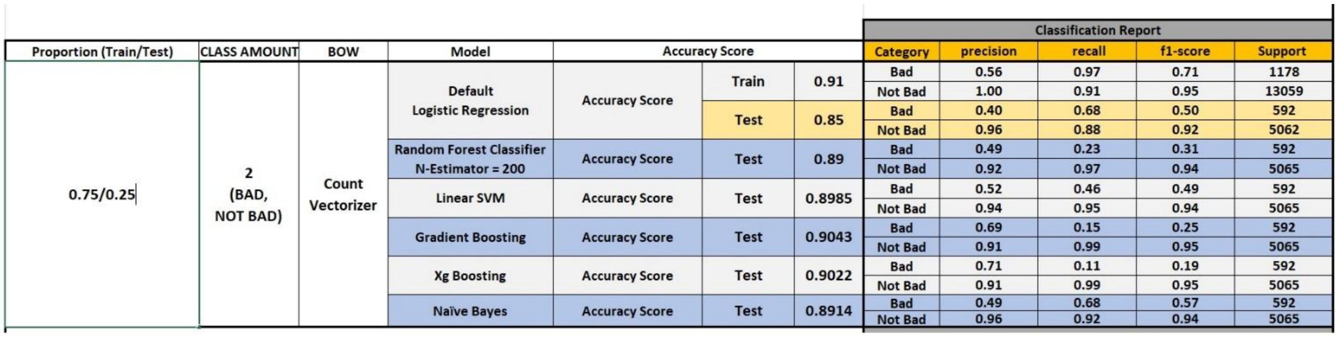
								Classification Report				
Proportion (Train/Test)	CLASS AMOUN	BOW		Model	Accuracy Score			Category	precision	recall	f1-score	Support
0.75/0.25	3 (BAD, GOOD, NEUTRAL)	(SMOTE) Count Vectorizer	PCA Linear Dimentiality Reduction	Default Logistic Regression	Accuracy Score	Test	0.482	Bad	0.13	0.18	0.15	600
								Good	0.76	0.57	0.65	4376
								Neutral	0.09	0.19	0.12	681
				Random Forest Classifier N-Estimator = 200	Accuracy Score	Test	0.6955	Bad	0.08	0.07	0.08	600
								Good	0.77	0.89	0.82	4376
								Neutral	0.00	0.00	0.00	681
			Truncated SVD Linear Dimentiality Reduction	Default Logistic Regression	Accuracy Score	Test	0.7931	Bad	0.12	0.23	0.16	600
								Good	0.81	0.61	0.70	4376
								Neutral	0.16	0.28	0.21	681
				Random Forest Classifier N-Estimator = 200	Accuracy Score	Test	0.7275	Bad	0.06	0.04	0.05	600
								Good	0.77	0.94	0.84	4376
								Neutral	0.00	0.00	0.00	681

Since after applying the Count Vectorizing, our model had dictionary length feature size and most of the features have only '0' values, we decided to apply Linear Dimensionality Reduction using with two different approaches, PCA and Truncated SVD, but the results were not good as expected as seen above. These techniques decreased the precision of the minority classes back to the very beginning values.

5. MACHINE LEARNING MODELS WITH 2 RATING CLASSES.

We were cautious to see what will happen once we decrease the rating classes form 3 to 2 by merging Rating 1 and 2 as 'Bad' and Rating 3,4,5 as 'Not Bad'. After tweaking with the rating classes we applied almost the same algorithms with expanded stop words and Count Vectorizer Bag of Words method and got the results as written below.

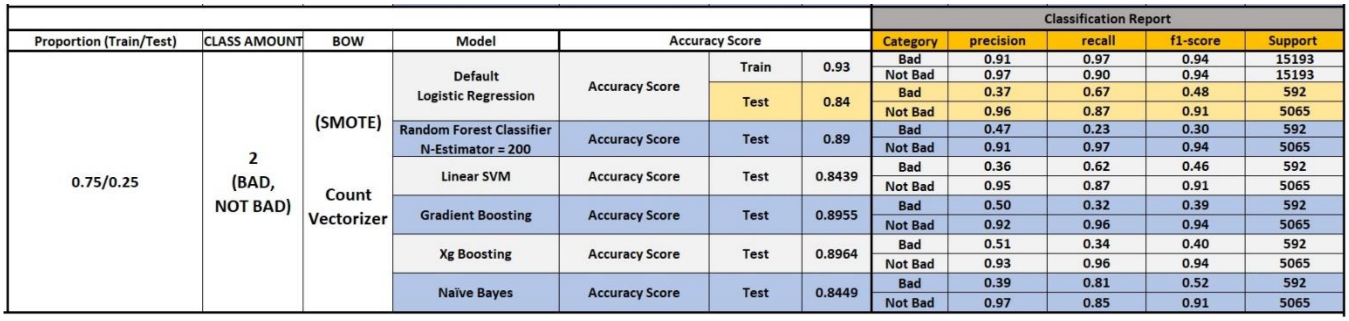
a. Expanded Stop Words, 2 Rating Classes, Count Vectorizer and All Algorithms:



							Classification Report				
Proportion (Train/Test)	CLASS AMOUNT	BOW	Model	Accuracy Score			Category	precision	recall	f1-score	Support
0.75/0.25	2 (BAD, NOT BAD)	Count Vectorizer	Default Logistic Regression	Accuracy Score	Train	0.91	Bad	0.56	0.97	0.71	1178
							Not Bad	1.00	0.91	0.95	13059
					Test	0.85	Bad	0.40	0.68	0.50	592
							Not Bad	0.96	0.88	0.92	5062
			Random Forest Classifier N-Estimator = 200	Accuracy Score	Test	0.89	Bad	0.49	0.23	0.31	592
							Not Bad	0.92	0.97	0.94	5065
			Linear SVM	Accuracy Score	Test	0.8985	Bad	0.52	0.46	0.49	592
							Not Bad	0.94	0.95	0.94	5065
			Gradient Boosting	Accuracy Score	Test	0.9043	Bad	0.69	0.15	0.25	592
							Not Bad	0.91	0.99	0.95	5065
			Xg Boosting	Accuracy Score	Test	0.9022	Bad	0.71	0.11	0.19	592
							Not Bad	0.91	0.99	0.95	5065
			Naïve Bayes	Accuracy Score	Test	0.8914	Bad	0.49	0.68	0.57	592
							Not Bad	0.96	0.92	0.94	5065

With the 2 classes, precision of Bad, 'minority class' increased up to 0.71 with XgBoosting. However, we could not say the same thing for Recall of the 'Bad' class. Despite Gradient Boosting got the best accuracy score Boosting algorithms are not as good as the others to predict the Recall.

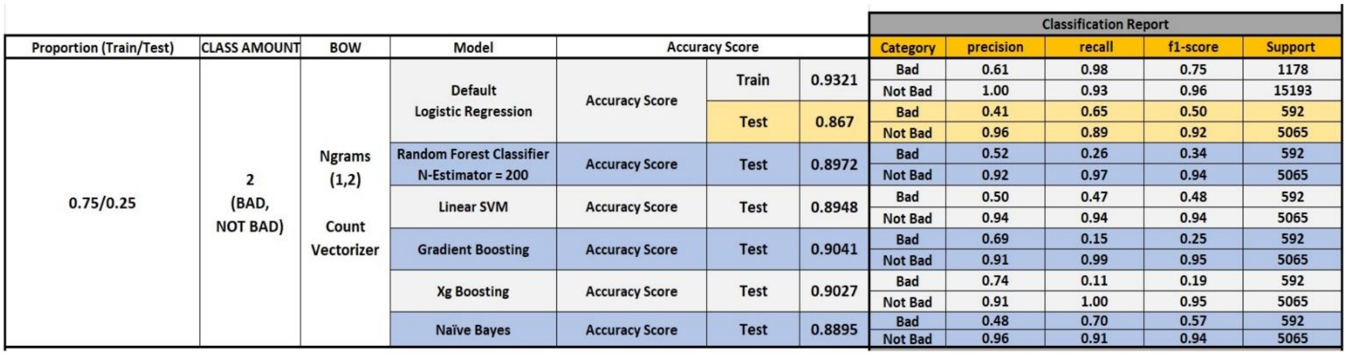
b. Expanded Stop Words, 2 Rating Classes, Count Vectorizer and All Algorithms and SMOTE:



							Classification Report				
Proportion (Train/Test)	CLASS AMOUNT	BOW	Model	Accuracy Score			Category	precision	recall	f1-score	Support
0.75/0.25	2 (BAD, NOT BAD)	(SMOTE) Count Vectorizer	Default Logistic Regression	Accuracy Score	Train	0.93	Bad	0.91	0.97	0.94	15193
							Not Bad	0.97	0.90	0.94	15193
					Test	0.84	Bad	0.37	0.67	0.48	592
							Not Bad	0.96	0.87	0.91	5065
			Random Forest Classifier N-Estimator = 200	Accuracy Score	Test	0.89	Bad	0.47	0.23	0.30	592
							Not Bad	0.91	0.97	0.94	5065
			Linear SVM	Accuracy Score	Test	0.8439	Bad	0.36	0.62	0.46	592
							Not Bad	0.95	0.87	0.91	5065
			Gradient Boosting	Accuracy Score	Test	0.8955	Bad	0.50	0.32	0.39	592
							Not Bad	0.92	0.96	0.94	5065
			Xg Boosting	Accuracy Score	Test	0.8964	Bad	0.51	0.34	0.40	592
							Not Bad	0.93	0.96	0.94	5065
			Naïve Bayes	Accuracy Score	Test	0.8449	Bad	0.39	0.81	0.52	592
							Not Bad	0.97	0.85	0.91	5065

After applying SMOTE to only minority class, the accuracy, precision and the recall of the classes decreased quite a bit but not as worse as 3 classes. Xg Boosting got the best accuracy and precision of the minority class, however strangely enough Naïve Bayes was the best predictor from the minority recall point of view with 0.81 score.

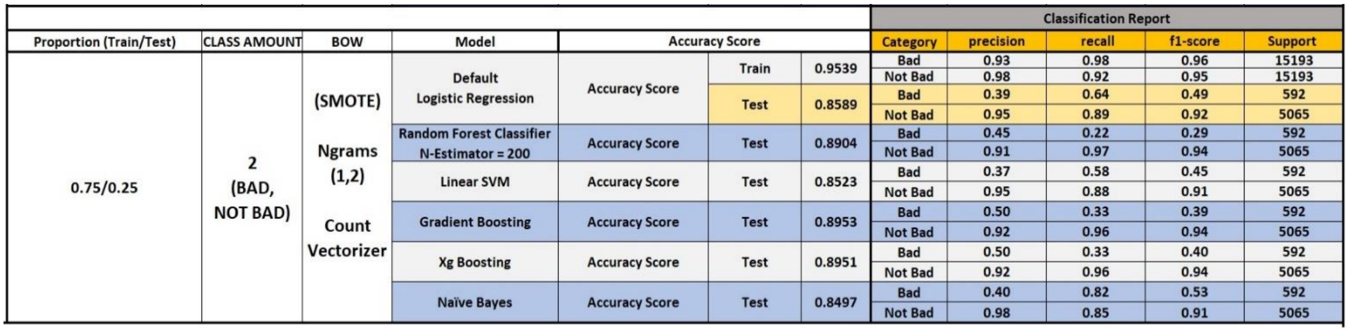
c. Expanded Stop Words, 2 Rating Classes, Count Vectorizer and All Algorithms and N_Grams(1,2):



							Classification Report				
Proportion (Train/Test)	CLASS AMOUNT	BOW	Model	Accuracy Score			Category	precision	recall	f1-score	Support
0.75/0.25	2 (BAD, NOT BAD)	Ngrams (1,2) Count Vectorizer	Default Logistic Regression	Accuracy Score	Train	0.9321	Bad	0.61	0.98	0.75	1178
							Not Bad	1.00	0.93	0.96	15193
					Test	0.867	Bad	0.41	0.65	0.50	592
							Not Bad	0.96	0.89	0.92	5065
			Random Forest Classifier N-Estimator = 200	Accuracy Score	Test	0.8972	Bad	0.52	0.26	0.34	592
							Not Bad	0.92	0.97	0.94	5065
			Linear SVM	Accuracy Score	Test	0.8948	Bad	0.50	0.47	0.48	592
							Not Bad	0.94	0.94	0.94	5065
			Gradient Boosting	Accuracy Score	Test	0.9041	Bad	0.69	0.15	0.25	592
							Not Bad	0.91	0.99	0.95	5065
			Xg Boosting	Accuracy Score	Test	0.9027	Bad	0.74	0.11	0.19	592
							Not Bad	0.91	1.00	0.95	5065
			Naïve Bayes	Accuracy Score	Test	0.8895	Bad	0.48	0.70	0.57	592
							Not Bad	0.96	0.91	0.94	5065

Once we applied N-grams (1,2) without SMOTE, Boosting algorithms got the best precision scores for the minority class with 0.74 (Xg Boosting) and 0.69 (Gradient Boosting). However as we saw in the previous model, they were not as good as the other algorithms to predict the recall of the minority class. Naïve Bayes again gave the best minority recall score.

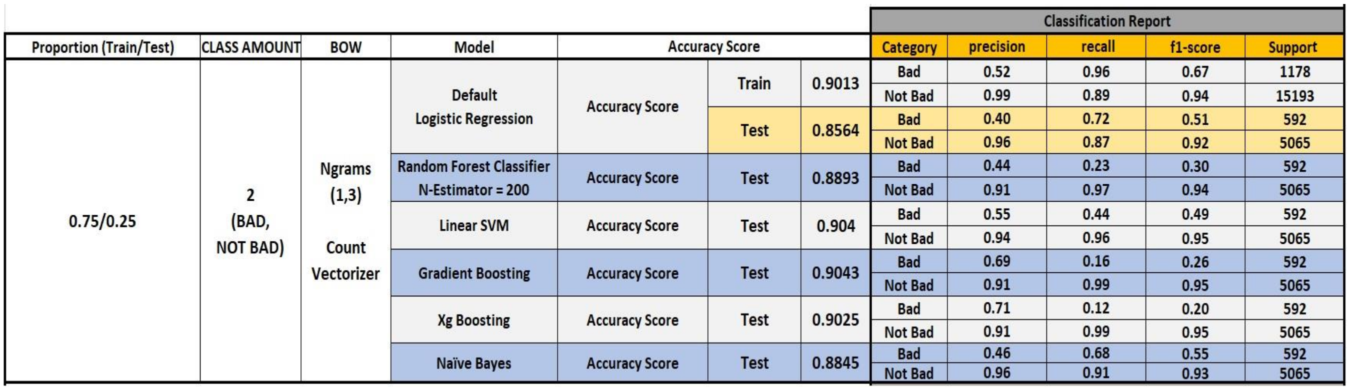
d. Expanded Stop Words, 2 Rating Classes, Count Vectorizer and All Algorithms, N_Grams(1,2) and SMOTE:



							Classification Report				
Proportion (Train/Test)	CLASS AMOUNT	BOW	Model	Accuracy Score			Category	precision	recall	f1-score	Support
0.75/0.25	2 (BAD, NOT BAD)	(SMOTE) Ngrams (1,2) Count Vectorizer	Default Logistic Regression	Accuracy Score	Train	0.9539	Bad	0.93	0.98	0.96	15193
							Not Bad	0.98	0.92	0.95	15193
					Test	0.8589	Bad	0.39	0.64	0.49	592
							Not Bad	0.95	0.89	0.92	5065
			Random Forest Classifier N-Estimator = 200	Accuracy Score	Test	0.8904	Bad	0.45	0.22	0.29	592
							Not Bad	0.91	0.97	0.94	5065
			Linear SVM	Accuracy Score	Test	0.8523	Bad	0.37	0.58	0.45	592
							Not Bad	0.95	0.88	0.91	5065
			Gradient Boosting	Accuracy Score	Test	0.8953	Bad	0.50	0.33	0.39	592
							Not Bad	0.92	0.96	0.94	5065
			Xg Boosting	Accuracy Score	Test	0.8951	Bad	0.50	0.33	0.40	592
							Not Bad	0.92	0.96	0.94	5065
			Naïve Bayes	Accuracy Score	Test	0.8497	Bad	0.40	0.82	0.53	592
							Not Bad	0.98	0.85	0.91	5065

With N_Grams(1,2) after applying the SMOTE, despite the accuracy of the algorithms and precision decreased explicitly, Naïve Bayes was the best predictor for the minority class recall again.

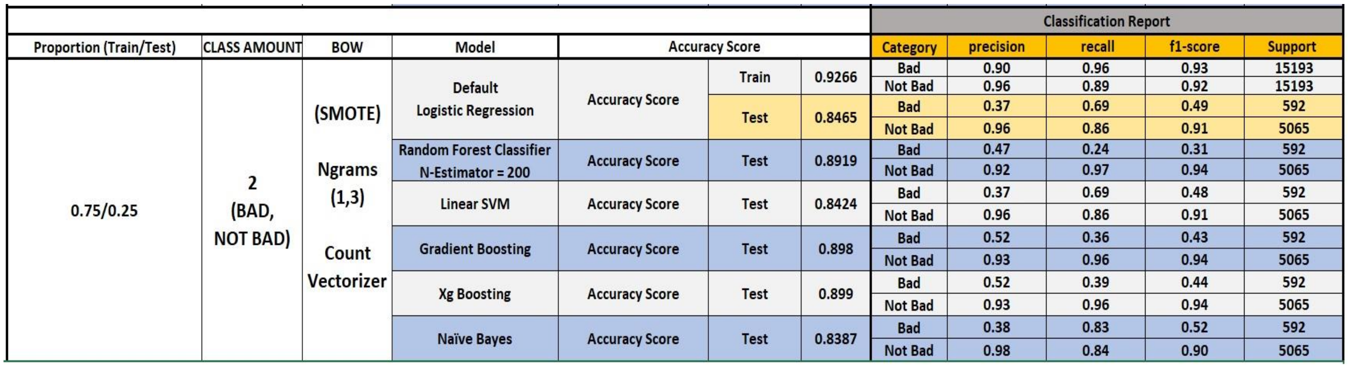
e. Expanded Stop Words, 2 Rating Classes, Count Vectorizer, All Algorithms and N_Grams(1,3):



							Classification Report				
Proportion (Train/Test)	CLASS AMOUNT	BOW	Model	Accuracy Score			Category	precision	recall	f1-score	Support
0.75/0.25	2 (BAD, NOT BAD)	Ngrams (1,3) Count Vectorizer	Default Logistic Regression	Accuracy Score	Train	0.9013	Bad	0.52	0.96	0.67	1178
							Not Bad	0.99	0.89	0.94	15193
					Test	0.8564	Bad	0.40	0.72	0.51	592
							Not Bad	0.96	0.87	0.92	5065
			Random Forest Classifier N-Estimator = 200	Accuracy Score	Test	0.8893	Bad	0.44	0.23	0.30	592
							Not Bad	0.91	0.97	0.94	5065
			Linear SVM	Accuracy Score	Test	0.904	Bad	0.55	0.44	0.49	592
							Not Bad	0.94	0.96	0.95	5065
			Gradient Boosting	Accuracy Score	Test	0.9043	Bad	0.69	0.16	0.26	592
							Not Bad	0.91	0.99	0.95	5065
			Xg Boosting	Accuracy Score	Test	0.9025	Bad	0.71	0.12	0.20	592
							Not Bad	0.91	0.99	0.95	5065
			Naïve Bayes	Accuracy Score	Test	0.8845	Bad	0.46	0.68	0.55	592
							Not Bad	0.96	0.91	0.93	5065

N_Grams (1,3) did not give better results than the N_Grams (1,2) as seen above. The best precision of the minority class was 0.71 with Xg Boosting and best recall of the minority class was 0.72 with Default Logistic Regression.

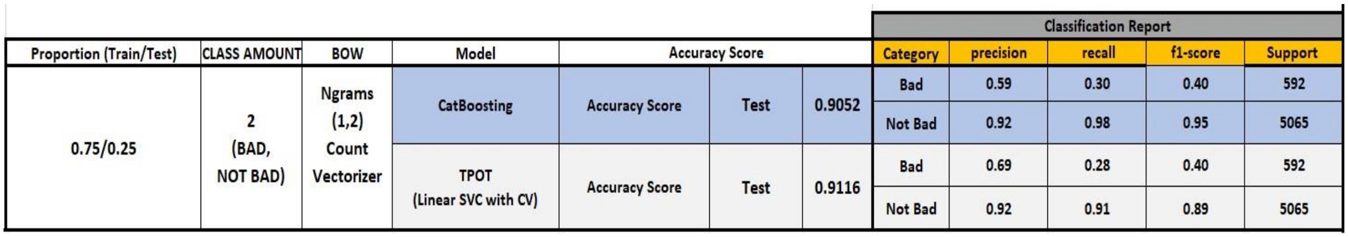
f. Expanded Stop Words, 2 Rating Classes, Count Vectorizer, All Algorithms, N_Grams(1,3) and SMOTE:



							Classification Report				
Proportion (Train/Test)	CLASS AMOUNT	BOW	Model	Accuracy Score			Category	precision	recall	f1-score	Support
0.75/0.25	2 (BAD, NOT BAD)	(SMOTE) Ngrams (1,3) Count Vectorizer	Default Logistic Regression	Accuracy Score	Train	0.9266	Bad	0.90	0.96	0.93	15193
							Not Bad	0.96	0.89	0.92	15193
					Test	0.8465	Bad	0.37	0.69	0.49	592
							Not Bad	0.96	0.86	0.91	5065
			Random Forest Classifier N-Estimator = 200	Accuracy Score	Test	0.8919	Bad	0.47	0.24	0.31	592
							Not Bad	0.92	0.97	0.94	5065
			Linear SVM	Accuracy Score	Test	0.8424	Bad	0.37	0.69	0.48	592
							Not Bad	0.96	0.86	0.91	5065
			Gradient Boosting	Accuracy Score	Test	0.898	Bad	0.52	0.36	0.43	592
							Not Bad	0.93	0.96	0.94	5065
			Xg Boosting	Accuracy Score	Test	0.899	Bad	0.52	0.39	0.44	592
							Not Bad	0.93	0.96	0.94	5065
			Naïve Bayes	Accuracy Score	Test	0.8387	Bad	0.38	0.83	0.52	592
							Not Bad	0.98	0.84	0.90	5065

With N_Grams (1,3) after applying the SMOTE, despite the accuracy of the algorithms and precision decreased explicitly, Naïve Bayes was the best predictor for the minority class recall again.

g. Expanded Stop Words, 2 Rating Classes, Count Vectorizer, Genetic Algorithms:



							Classification Report				
Proportion (Train/Test)	CLASS AMOUNT	BOW	Model	Accuracy Score			Category	precision	recall	f1-score	Support
0.75/0.25	2 (BAD, NOT BAD)	Ngrams (1,2) Count Vectorizer	CatBoosting	Accuracy Score	Test	0.9052	Bad	0.59	0.30	0.40	592
							Not Bad	0.92	0.98	0.95	5065
			TPOT (Linear SVC with CV)	Accuracy Score	Test	0.9116	Bad	0.69	0.28	0.40	592
							Not Bad	0.92	0.91	0.89	5065

By applying the Genetic Algorithms seen above, we got the best accuracy of the model with 0.9116 (TPOT(Linear SVC with CV)). Despite the precision of the minority class was not high as much as the other boosting algorithms, from the recall point of view, TPOT got the best score. On the other hand Cat Boosting gave the best majority class recall with 0.98 and mostly overperformed the other boosting algorithms such as Xg Boosting and Gradient Boosting.

5. CONCLUSION:

In this study, we tried to predict the rating scores based on the reviews left by the female customers. Before going through the model result, it is explicitly shown that data set preparation and feature engineering are as much important as the model creation. While data wrangling we merged two string features by dropping only null values which was occurred at the same rows for both features to increase the effect of the reviews and expanding the stop words by adding the most common 70 and least common 70 words to the stop words list make slightly difference of the precision and the recall scores of the minority classes.

Finally decreasing the rating numbers from 3 to 2 made a great impact on the accuracy, precision, recall of the minority class. Resampling technique and linear dimensionality reduction did not make a positive improvement of the model accuracy but decreased the results.

Boosting Algorithms gave the best precision scores, but they were not so good on the recall. Naïve Bayes mostly gave better balanced scores from precision and recall point of view. And finally, Genetic Algorithms won the race.

6. FUTURE STUDY:

As a future study, we will concentrate on the topics mentioned below:

- We will implement Deep Learning models to get better results.
- We will use word2vec technique in NLP part.
- We will implement Dask library for parallel processing to decrease run time.
- After decreasing run time, we will focus on hyperparameter tuning more.