A Study on Sentiment Analysis: Methods and Tools

Group Number 1 / Delivery 7

June 2019

Table 1: Group Members

Group Members			
Group Member Name	***Number***	***Department***	***Semester***
Şevvalnur Kahraman	150116822	Computer Engineering	4
Nurcihane Köroğlu	150115062	Computer Engineering	4
Emine Feyza Memiş	150114077	Computer Engineering	4
Gizem Aybüke Çeri	150315060	Industrial Engineering	4
Yiğit Hallaç	150714036	Electronics Engineering	4
Amine Boumusou	199518004	Computer Engineering	4

1 Abstract

Sentiment analysis is the process of analyzing online pieces of writing to determine the emotional tone they carry. It is the study of people's opinions, appraisals, attitudes, and emotions toward entities, individuals, issues, events, topics and their attributes. In simple words, it is used to find the author's attitude towards something.

Since the businesses always want to find public or consumers' opinions about their brands, sentiment analysis became very popular nowadays. This contextual mining helps companies to understand if their customer is satisfied by their products/services or not. With the increasing need for research and development of the deep learning concept, sentiment analysis became a very useful way to see the customers' real opinions represented on social media or web. Because of these two fields' explosive growth, it is extremely difficult to find and monitor the opinion sites, and distill the information contained in them in order to conduct an analysis due to the huge amount of opinionated text. Owing to the improvement of technology, there are some tools helping to categorize pieces of writing as positive, neutral, or negative.

With this project, we aimed to analyse the sentiments of some reviews taken from the imdb.com, amazon.com and yelp.com websites/fields which include positive or negative labelled comments about movies, products and restaurants. By using sentiment analysis, we will try to understand the users' underlying intentions and reactions by monitoring online conversations. The project will start with the reorganization of the dataset. Then, the features and attributes will be selected. After that, tokens will be labeled as positive or negative by conducting the analysis and the top tokens will be found. In order to udnerstand the behaviour of our data, clustering will be used. With the sense of finding the most accurate results, different algorithms will be used and compared by their accuracy and AUC values.

2 Introduction

In the past few years, machine learning has revolutionized the way we do business. A disruptive breakthrough that differentiates machine learning from other approaches to automation is a step away from the rules-based programming. ML algorithms allowed engineers to leverage data without explicitly programming machines to follow specific paths of problem-solving. Instead, machines

themselves arrive at the right answers based on the data they have. This capability made business executives reconsider the ways they use data to make decisions.[1]

The aim of the project is to collect sentiments (reviews, comments) of product such as movies and restaurants from some datasets and define and select features by using machine learning algorithms such as Decision Tree, Naïve Bayes, K-nn in order to get more reliable results. We firstly started by finding datasets which we prepared and we format in order to select features, and then we proceed training and testing.

3 Related Work

Sentiment Analysis is a process of automatic extraction of features by mode of notions of others about specific product, services or experience. They thought that analyzing customer opinion is very important to rate the product. To automate rate the opinions in the form of unstructured data is been a challenging problem. Thus, this paper discusses about Sentiment analysis methods and tools used. Kaushik et. al. [2] , introduced methods and tools in sentiment analysis. This paper has same purpose with our project. They were classified sentences as negative or positive as in our project.

Joscha et. al, in their paper [3] devised and compared various techniques like Bag of words models, n-grams for using semantic information to improve the performance of sentiment analysis. The earlier approaches did not consider the semantic associations between sentences or documents parts.

Amir Hossein Yazdavar et al. in this paper [4] provided novel understanding of sentiment analysis problem containing numerated data in drug reviews. They analyzed sentences which contained quantitative terms to classify them into opinionated or non-opinionated and also to identify the polarity expressed by using fuzzy set theory. The development of fuzzy knowledge base was done by interviewing several doctors from various medical centers.

Dhiraj Murthy in his paper [5] he identified what roles do tweets play in political elections. He pointed out that even though there were various researches and studies done to find out the political engagement of Twitter, no work was done to find out if these tweets were Predictive or Reactive. In his paper, he concluded that the tweets are more reactive than predictive. He found out that electoral success in not at all related to the success on Twitter and that various social media platforms were used to increase the popularity of a candidate by generating a buzz around them.

In this section, we will examine articles with similar topics as our project. In fact, there are many articles with similar approaches to this project. But we will explain a few of them.

4 Approach

The dataset contains sentences labelled with positive or negative sentiment, extracted from reviews of products, movies, and restaurants with their format like "sentence score".

Score is either 1 (for positive) or 0 (for negative) and each sentences come from three different websites/fields that are imdb.com, amazon.com and yelp.com. For each website, there exist 500 positive and 500 negative sentences. Those were selected randomly for larger datasets of reviews. We attempted to select sentences that have a clearly positive or negative connotation, the goal was for no neutral sentences to be selected.

If you want to see the full datasets you can look for the followings: **imdb:** Maas et. al., 2011 'Learning word vectors for sentiment analysis'
amazon: McAuley et. al., 2013 'Hidden factors and hidden topics: Understanding rating dimensions with review text'
yelp: Yelp dataset challenge http://www.yelp.com/dataset_challenge

Table 2: Features/Attributes

Features/Attributes				
Number	***Features***	***Description***	***Type***	***Target***
1	Sentence	It contains sentences come from three different websites/fields	String	No
2	Class	It contains score is either 1 (for positive) or 0 (for negative)	Numeric	Yes

Table 3: Class Attribute Values

Class Attribute Values			
Number	***Class Value***	***Number of Instances***	***Description***
1	0	1500	It contain negative sentences
2	1	1500	It contains positive sentences

5 Experiment Setup

5.1 Most used words

Table 4: Most used words in data set

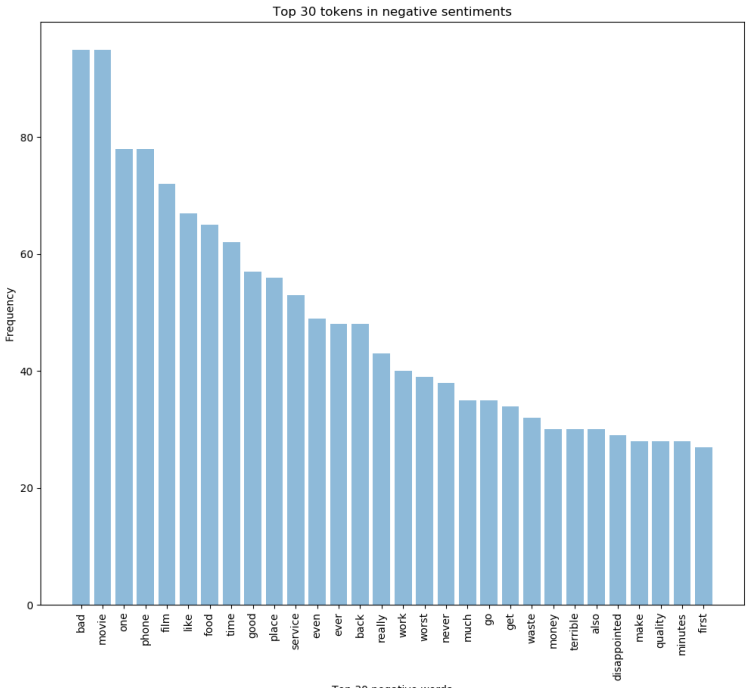
Most Used Words	
Word	***Count***
good	221
great	192
movie	174
phone	159
film	150
one	136
like	123
food	122
place	111
time	106
really	102
service	101
bad	96
would	84
well	83
best	77
even	74
ever	72
also	71
back	68
quality	64
love	61
go	61
made	57
product	56
get	55
work	54
could	53
works	52
better	51
nice	51
much	50
recommend	49
never	49
use	49
excellent	48
sound	47
headset	47
think	46
first	45
way	44
battery	44
see	42
make	42
pretty	41
acting	40
still	39

5.2 Explanation Before Finding Top Tokens

We used *python* language in our project. We use *pandas* library for reading, representing data as a data frame. We use *numpy* library for array operations. Also for drawing graphics of our final results we use matplotlib library.
First of all, we converted all uppercase to lowercase. Before finding tokens in sentiments we made recovery for each sentences. For example : *i'm to i am*
After making recovery we defined stop words in English by using *nltk* library and we deleted them from sentences.
Negative sentences represented with 0 and positive sentences with 1 in our data set. We calculated term frequencies of words as positive or negative after deleting stop words.
We stored that words in arrays and plotted graphics with the help of this arrays.

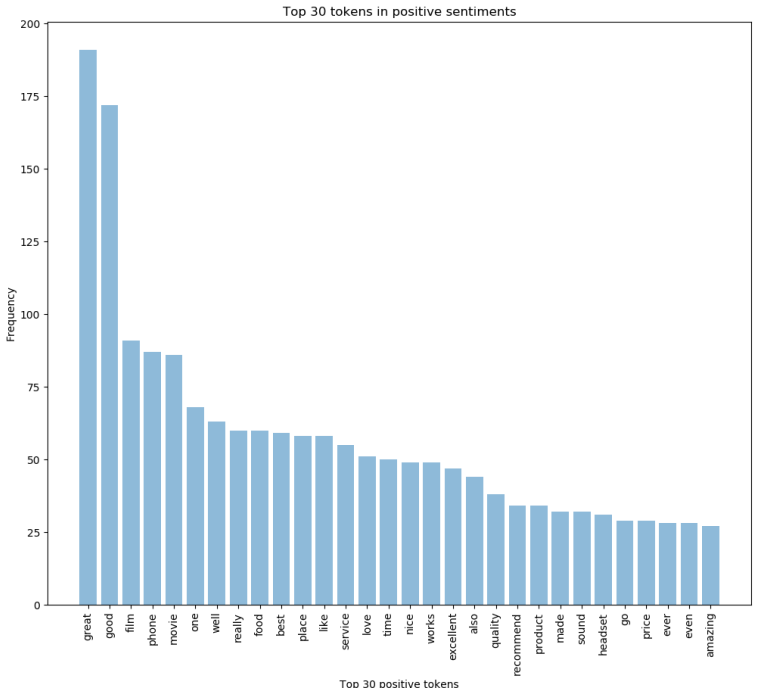
5.3 Top 30 Tokens in Negative Sentiments

For this part of project we found top 30 words in negative sentiments as frequency.
We found this tokens in negative sentences as we mentioned in the first section of this document.
We can see from the graphic the most used word is *bad* in negative sentences.



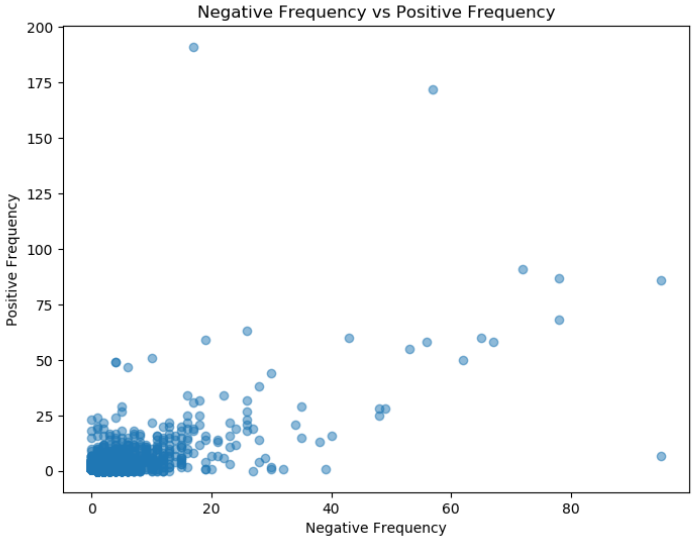
5.4 Top 30 Tokens in Positive Sentiments

For this part of project we found top 30 words in positive sentiments as frequency
We found this tokens in positive sentences as we mentioned in the first section of this document.
We can see from the graphic the most used word is *great* in negative sentences.



5.5 Negative Frequency vs Positive Frequency

For this part we compare negative and positive word's term frequencies that we found before in the graph.
We made this comparison with using scatter plot.
We can see that most of the distribution is in the region near zero. The reason for this is that most words have been used for 0-20 times. Each word may not be as distinctive as *bad* or *good*.



5.6 Feature Selection

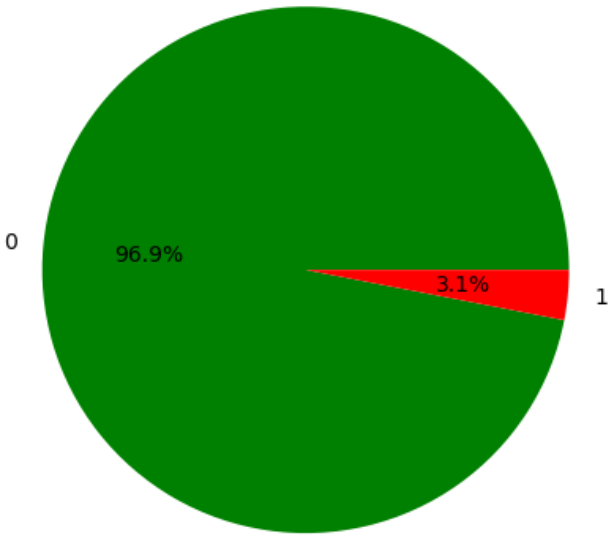
Table 5: Features and Their Values

Features and Their Values							
#	***Feature Names***	***Description***	***Cluster1 (Negative)***	***Cluster2 (Positive)***	***Total***	***Type***	***Overall Average***
1	good	word	57	172	229	nominal	0.076333
2	great	word	17	191	208	nominal	0.069333
3	movie	word	95	86	181	nominal	0.060333
4	phone	word	78	87	165	nominal	0.055000
5	film	word	72	91	163	nominal	0.054333
6	one	word	78	68	146	nominal	0.048667
7	food	word	65	60	125	nominal	0.041667
8	like	word	67	58	125	nominal	0.041667
9	place	word	56	58	114	nominal	0.038000
10	time	word	62	50	112	nominal	0.037333
11	service	word	53	55	108	nominal	0.036000
12	really	word	43	60	103	nominal	0.034333
13	bad	word	95	7	102	nominal	0.034000
14	well	word	26	63	89	nominal	0.029667
15	best	word	19	59	78	nominal	0.026000
16	even	word	49	28	77	nominal	0.025667
17	ever	word	48	28	76	nominal	0.025333
18	also	word	30	44	74	nominal	0.024667
19	back	word	48	25	73	nominal	0.024333
20	quality	word	28	38	66	nominal	0.022000
21	go	word	35	29	64	nominal	0.021333
22	love	word	10	51	61	nominal	0.020333
23	made	word	26	32	58	nominal	0.019333
24	work	word	40	16	56	nominal	0.018667
25	product	word	22	34	56	nominal	0.018667
26	get	word	34	21	55	nominal	0.018333
27	nice	word	4	49	53	nominal	0.017667
28	better	word	26	27	53	nominal	0.017667
29	[illegible]	[illegible]	6	47	53	[illegible]	0.017667

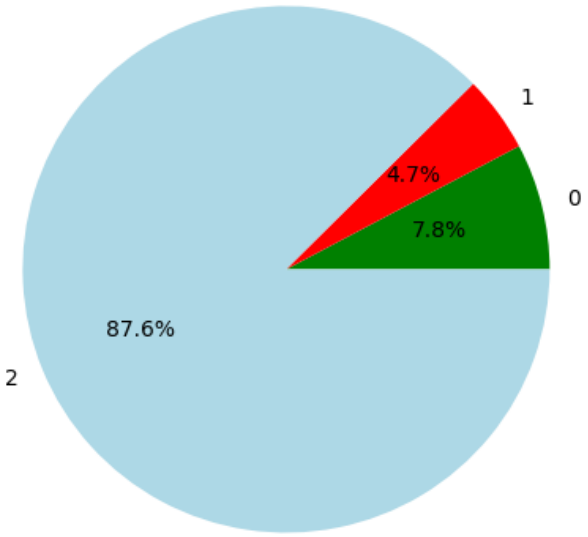
Table 6: Features and Their Values

Features and Their Values			
	k=2	***k=3***	***k=5***
Cluster 0	2664	213	2377
Cluster 1	84	129	53
Cluster 2	-	2406	114
Cluster 3	-	-	51
Cluster 4	-	-	153
Overall Average	1374.0	916.0	549.6
STD	0.17214326065001764	0.5623139798013362	1.0479497680905905
SSE	1396	6431	3786
NMI	0.0008379505803413498	0.03459079818237664	0.0032774096303950363
Silhouette Value	0.001614085096514876	0.004749202598581923	0.004377112827600871
RI	0.49994621630371944	0.5041961880871497	0.5001788374137408

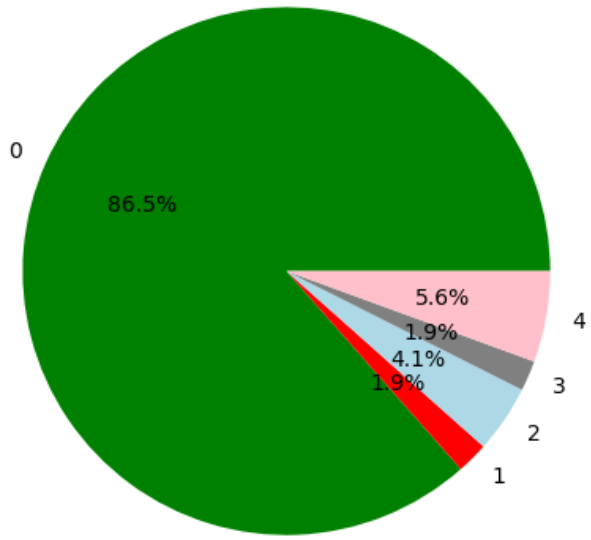
5.7 Pie Chart for k is 2



5.8 Pie Chart for k is 3



5.9 Pie Chart for k is 5



6 Experimental Results

6.1 Feature Selection

Table 7: First 15 Word For Mutual Information Gain

Word	***Mutual Information Gain Value***
good	0.050022
great	0.047929
movie	0.039749
film	0.036147
phone	0.035585
one	0.031655
like	0.027275
food	0.026628
time	0.025886
place	0.025647
service	0.024317
really	0.023111
bad	0.021199
well	0.019981
even	0.017811

Table 8: Last 15 Word For Mutual Information Gain

Word	***Mutual Information Gain Value***
11	0.000231
1199	0.000231
15lb	0.000231
18	0.000231
18th	0.000231
1928	0.000231
1947	0.000231
1948	0.000231
1949	0.000231
1971	0.000231
1973	0.000231
1979	0.000231
1980	0.000231
1986	0.000231
1995	0.000231

This 15 words are the least common words in our dataset.

Table 9: First 15 chi-square:

Word	***Chi-square value***
great	0.047929
bad	0.021199
excellent	0.011919
good	0.050022
works	0.012124
love	0.014241
waste	0.007679
nice	0.012371
worst	0.009088
terrible	0.007444
poor	0.006045
delicious	0.005344
awesome	0.004643
money	0.007198
horrible	0.004410

This 15 words are the most common words in our dataset.

Table 10: Last 15 chi-square:

Word	***Chi-square value***
minute	0.000924
given	0.001848
verizon	0.002079
evening	0.000462
morgan	0.000462
production	0.001155
wine	0.000924
eggs	0.000462
internet	0.001617
replace	0.000924
heard	0.000924
comments	0.000462
ends	0.000462
stage	0.000462
central	0.000462

This 15 words are the least common words in our dataset.

Two selection method have different formulas. But they both find common words. They're not the same words, but they're all are the most common words in our dataset.

6.2 Algorithms

We did 10 tests for each algorithm. Because we wanted our results to be more reliable. We split our dataset into 10 part and we did training and testing on each of them

Table 11: Algorithm name: Decision Tree:

K-fold	***Accuracy***	***AUC***
0	0.727	0.732
1	0.740	0.738
2	0.777	0.786
3	0.770	0.773
4	0.783	0.784
5	0.757	0.757
6	0.743	0.743
7	0.770	0.773
8	0.750	0.747
9	0.720	0.743

Table 12: Algorithm name: Naive Bayes

K-fold	***Accuracy***	***AUC***
0	0.803	0.804
1	0.843	0.843
2	0.793	0.802
3	0.837	0.835
4	0.853	0.851
5	0.767	0.767
6	0.810	0.811
7	0.820	0.820
8	0.813	0.812
9	0.733	0.769

Table 13: Algorithm name: k-NN neighbor number is 1

K-fold	***Accuracy***	***AUC***
0	0.633	0.576
1	0.620	0.570
2	0.510	0.574
3	0.567	0.603
4	0.673	0.688
5	0.623	0.619
6	0.653	0.643
7	0.737	0.733
8	0.630	0.666
9	0.667	0.569

Table 14: Algorithm name: k-NN neighbor number is 2

K-fold	***Accuracy***	***AUC***
0	0.617	0.550
1	0.587	0.530
2	0.443	0.521
3	0.530	0.578
4	0.593	0.616
5	0.583	0.578
6	0.610	0.597
7	0.693	0.711
8	0.550	0.601
9	0.687	0.571

Table 15: Decision Tree with remove words occurring less than 5 times

K-fold	***Accuracy***	***AUC***
0	0.680	0.679
1	0.740	0.739
2	0.727	0.737
3	0.763	0.768
4	0.743	0.744
5	0.727	0.726
6	0.753	0.752
7	0.740	0.744
8	0.773	0.772
9	0.710	0.725

Table 16: Naive Bayes with remove words occurring less than 5 times

K-fold	***Accuracy***	***AUC***
0	0.737	0.741
1	0.777	0.781
2	0.723	0.731
3	0.797	0.795
4	0.803	0.801
5	0.733	0.734
6	0.770	0.771
7	0.793	0.793
8	0.807	0.809
9	0.733	0.763

Table 17: k-NN neighbor number is 1 with remove words occurring less than 5 times

K-fold	***Accuracy***	***AUC***
0	0.620	0.641
1	0.600	0.618
2	0.700	0.682
3	0.680	0.668
4	0.673	0.666
5	0.597	0.598
6	0.637	0.640
7	0.680	0.657
8	0.740	0.724
9	0.617	0.654

Table 18: k-NN neighbor number is 2 with remove words occurring less than 5 times

K-fold	***Accuracy***	***AUC***
0	0.650	0.617
1	0.567	0.573
2	0.687	0.687
3	0.637	0.638
4	0.663	0.661
5	0.577	0.576
6	0.600	0.601
7	0.673	0.694
8	0.703	0.727
9	0.730	0.680

Table 19: Naive Bayes first 1000 from chi-square

K-fold	***Accuracy***	***AUC***
0	0.273	0.303
1	0.330	0.346
2	0.360	0.403
3	0.347	0.371
4	0.353	0.364
5	0.337	0.338
6	0.363	0.371
7	0.297	0.320
8	0.323	0.347
9	0.290	0.354

Table 20: Naive Bayes first 3000 from chi-square

K-fold	***Accuracy***	***AUC***
0	0.320	0.342
1	0.317	0.329
2	0.360	0.401
3	0.357	0.379
4	0.387	0.396
5	0.360	0.361
6	0.367	0.373
7	0.313	0.330
8	0.350	0.367
9	0.303	0.363

Table 21: Naive Bayes first 5000 from chi-square

K-fold	***Accuracy***	***AUC***
0	0.623	0.637
1	0.680	0.683
2	0.613	0.626
3	0.653	0.660
4	0.623	0.623
5	0.657	0.657
6	0.650	0.652
7	0.623	0.629
8	0.610	0.621
9	0.550	0.600

Table 22: Naive Bayes with last 4000 from chi-square

K-fold	***Accuracy***	***AUC***
0	0.810	0.816
1	0.857	0.856
2	0.830	0.836
3	0.850	0.851
4	0.883	0.883
5	0.823	0.824
6	0.827	0.828
7	0.823	0.828
8	0.807	0.811
9	0.760	0.792

Table 23: Naive Bayes with last 2000 from chi-square

K-fold	***Accuracy***	***AUC***
0	0.837	0.843
1	0.867	0.865
2	0.843	0.849
3	0.850	0.852
4	0.877	0.877
5	0.803	0.804
6	0.843	0.844
7	0.817	0.821
8	0.837	0.846
9	0.790	0.818

Table 24: Naive Bayes with last 1000 from chi-square

K-fold	***Accuracy***	***AUC***
0	0.843	0.851
1	0.867	0.866
2	0.853	0.862
3	0.857	0.861
4	0.860	0.862
5	0.803	0.804
6	0.853	0.855
7	0.817	0.824
8	0.840	0.848
9	0.807	0.833

6.3 Mean Accuracy

Table 25: Mean Accuracy

Algorithm	***Accuracy***	***AUC***
Decision Tree	0.754	0.757
Naive Bayes	0.807	0.811
k-NN neighbor number is 1	0.631	0.624
k-NN neighbor number is 2	0.589	0.585
Decision Tree with remove words occurring less than 5 times	0.736	0.739
Naive Bayes with remove words occurring less than 5 times	0.767	0.772
k-NN neighbor number is 1 with remove words occurring less than 5 times	0.654	0.655
k-NN neighbor number is 2 with remove words occurring less than 5 times	0.649	0.645

We can see that k-NN algorithm with neighbour number is 2 has the lowest accuracy and AUC value. Naive Bayes algorithm has the highest accuracy and AUC value.
When k-NN algorithm neighbour number increases, accuracy and AUC value is decreases.
k-nn Algorithm gives higher accuracy and AUC values when we remove the words occurring less than 5 times. But the Decision Tree and Naive Bayes gives lower accuracy and AUC values when we remove the words occurring less than 5 times.

7 Conclusions

We have collected 3 datasets, with 1000 sentence for each, thus 3000 sentences in total, half of them are negative where as the other half is positive.

We have done 10 tests for all the algorithms that we have used (Decision Tree, Naïve Bayes, K-nn) and we split the dataset into 10 parts and we did training and testing on each of them in order to get more reliable results. As the tests have demonstrated, it is clear that Naïve Bayes is the most reliable algorithm in our project regarding its accuracy and AUC.

References

[1] Supervised Learning Use Cases: Low-Hanging Fruit in Data Science for Businesses https://www.altexsoft.com/blog/business/supervised-learning-use-cases-low-hanging-fruit-in-data-science-for-businesses

[2] A Study on Sentiment Analysis: Methods and Tools, Abhishek Kaushik, Anchal Kaushik , Sudhanshu Naithani , Kiel University of Applied Sciences, Computer and Electrical Department, Sokratesplatz , 24149 Kiel, Germany

[3] Joscha Markle-Huß, Stefan Feuerriegel, Helmut Prendinger.2017 Improving Sentiment Analysis with Document-Level Semantic Relationships from Rhetoric Discourse Structures, Proceedings of the 50th Hawaii International Conference on System Sciences

[4] Amir Hossein Yazdavar, MonirehEbrahimi, Naomie Salim,2016,Fuzzy Based Implicit Sentiment Analysis on Quantitative Sentences, Faculty of Computing, UniversitiTechnologi Malaysia, Johor, Malaysia,Journal of Soft Computing and Decision Support Systems vol 3:4,pp.7-18.

[5] Dhiraj Murthy, Twitter and elections: are tweets, predictive, reactive, or a form of buzz, Information, Communication Society, 18:7, 816-831, DOI:10.1080/1369118X.2015.1006659

[6] Naive Bayes - https://scikit-learn.org/stable/modules/naivebayes.html

[7] Classification - https://scikit-learn.org/stable/modules/tree.html

[8] sklearn.neighbors.KNeighborsClassifier - https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNei

[9] sklearn.cluster.KMeans - https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

[10] sklearn.featureextraction.text.TfidfVectorizer- https://scikit-learn.org/stable/modules/generated/sklearn.featuree

[11] sklearn.featureextraction.text.CountVectorizer- https://scikit-learn.org/stable/modules/generated/sklearn.feature