Practical Machine Learning

Project 2

Documentation

For this task, I have chosen 2 unsupervised methods, such as, K-means and DBSCAN.

About dataset

The dataset consists 3 columns, "tweet_id", having a unique value for each tweet, "sentiment" with 13 different attributes, and "content" with the respective text. The dataset is unbalanced, having the following distribution: neutral - 8638, worry - 8459, happiness – 5209, sadness- 5165,love - 3842, surprise – 2187, fun - 1776, relief – 1526, hate – 1323, empty – 827, enthusiasm – 759, boredom - 179, anger – 110, 40000 in total.

In addition, all the tweets were normalized with a function, which removes the urls, the user references, punctuation, lowercase the text, tokenize and uses stemming. It is important to mention that the dataset was downsampled to include only five classes.

The dataset was split in training and test sets to evaluate the model on unseen data.

Obtaining the baseline scores

The comparisons with random chance was implemented with a dummy classifier from sklearn, and the supervised baseline was obtained after fitting a Logistic Regression model.

The random chance was obtained with a dummy classifier, obtaining an accuracy of 0.20%.The classification report is:

	Precision	recall	F1-score
Happines	0.16	0.19	0.17
Love	0.27	0.19	0.22
Neutral	0.25	0.20	0.23
Sadness	0.12	0.19	0.15
Worry	0.17	0.3	0.19
Accuracy			0.20
Macro avg	0.20	0.20	0.19
Weighted avg	0.21	0.20	0.20

The classification report for the Logistic Regression is:

	Precision	recall	F1-score
Happines	0.41	0.43	0.42
Love	0.48	0.49	0.49
Neutral	0.49	0.55	0.52
Sadness	0.41	0.46	0.43
Worry	0.45	0.34	0.39
Accuracy			0.45
Macro avg	0.45	0.45	0.45
Weighted avg	0.45	0.45	0.45

1. K-Means

The first unsupervised method was K-means, a clustering algorithm that splits the data in a certain number of clusters. The **first feature** was to use Bag of Words representation and furthermore, performed the grid search in order to obtain the best combination of parameters.

The parameters tuned are 'n_clusters' represent the number of clusters to form and the number of centroids to generate, 'init' is the method for initialization, where 'k-means++' selects initial cluster centers in a smart way to speed up convergence and 'random' which chooses n_clusters observations at random from the data for the initial centroids, and 'algorithm' used to find the clusters.

n_clusters	init	algorithm	silhouette_score
13	k-means++	lloyd	-0.01996
13	k-means++	elkan	-0.024592
13	random	lloyd	-0.006365
13	random	elkan	-0.007515
20	k-means++	lloyd	-0.029015
20	k-means++	elkan	0.002963
20	random	lloyd	-0.037346
20	random	elkan	-0.019714
30	k-means++	lloyd	-0.031569
30	k-means++	elkan	-0.048705
30	random	lloyd	-0.032962
30	random	elkan	-0.047013
60	k-means++	lloyd	-0.056333
60	k-means++	elkan	-0.062695
60	random	lloyd	-0.056521

60	random	elkan	-0.053925
100	k-means++	lloyd	-0.060151
100	k-means++	elkan	-0.068296
100	random	lloyd	-0.05632
100	random	elkan	-0.128359
150	k-means++	lloyd	-0.067394
150	k-means++	elkan	-0.067139
150	random	lloyd	-0.123689
150	random	elkan	-0.197579
200	k-means++	lloyd	-0.088544
200	k-means++	elkan	-0.078641
200	random	lloyd	-0.192064
200	random	elkan	-0.072161
250	k-means++	lloyd	-0.069982
250	k-means++	elkan	-0.063098
250	random	lloyd	-0.160543
250	random	elkan	-0.072917

For the BOW approach, silhouette scores mostly fall into the negative range, indicating that clusters might not be well-defined or adequately separated.

The second feature was to use TF-IDF representation and a grid search was performed to obtain the best combination of parameters.

n_clusters	**init**	**algorithm**	**silhouette_score**
13	k-means++	lloyd	0.010031
13	k-means++	elkan	0.010341
13	random	lloyd	0.009911
13	random	elkan	0.010315
20	k-means++	lloyd	-0.030392
20	k-means++	elkan	0.011165
20	random	lloyd	0.012095
20	random	elkan	0.012264
30	k-means++	lloyd	0.012378
30	k-means++	elkan	0.012004
30	random	lloyd	0.013138
30	random	elkan	0.013446
60	k-means++	lloyd	-0.027551
60	k-means++	elkan	-0.033109
60	random	lloyd	0.014643
60	random	elkan	0.014134
100	k-means++	lloyd	0.017784
100	k-means++	elkan	-0.018142

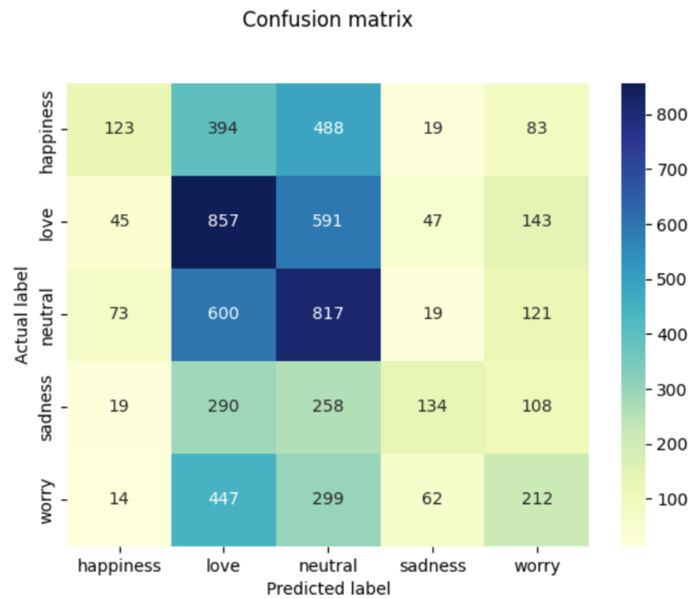
100	random	lloyd	0.016545
100	random	elkan	0.016001
150	k-means++	lloyd	-0.0222
150	k-means++	elkan	-0.026815
150	random	lloyd	0.018766
150	random	elkan	0.017306
200	k-means++	lloyd	-0.034234
200	k-means++	elkan	-0.020224
200	random	lloyd	0.018691
200	random	elkan	0.018253
250	k-means++	lloyd	-0.026315
250	k-means++	elkan	-0.025677
250	random	lloyd	0.019082
250	random	elkan	0.0187

For the TF-IDF approach, while there are still negative scores, there are instances of positive silhouette scores as well, which are notably higher than those from the BOW approach.

Using the best combination, with the TF-IDF approach, 'n_clusters' = 250, 'init' = random, the classification is:

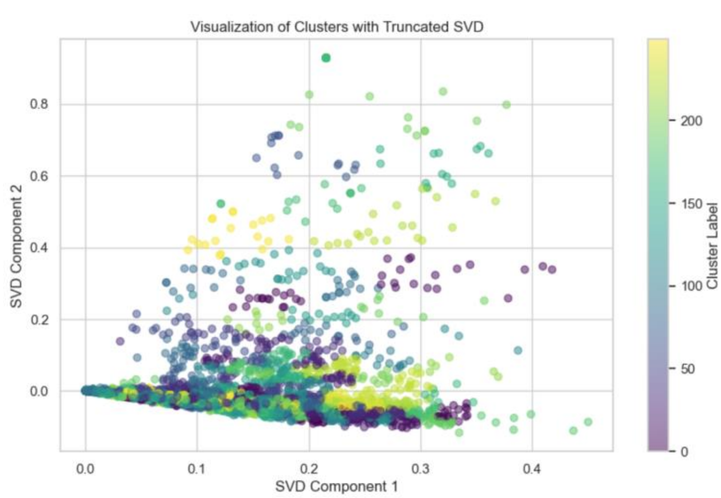
	Precision	recall	F1-score
Happines	0.45	0.11	0.18
Love	0.33	0.51	0.40
Neutral	0.33	0.50	0.40
Sadness	0.48	0.17	0.25
Worry	0.32	0.21	0.25
Accuracy			0.34
Macro avg	0.38	0.30	0.29
Weighted avg	0.37	0.34	0.32

The Confussion Matrix is:



The model seems to struggle particularly with distinguishing between 'happiness' and 'neutral', as well as 'sadness' and 'neutral'. There is a substantial number of instances where 'worry' is misclassified as 'love' or 'neutral', which could point to similar lexical features being used to express these emotions, or it might suggest that the representation is not capturing the nuances well enough.

For a better visualization, it is necessary to reduce the dimensionality of the dataset. This procces allows to transform the high-dimensional data into a lower-dimensional space. By applying techniques such as Truncated Singural Value Decomposition, we can obtain the most significant features of the data while simplifying the visualization.



There seems to be a dense region where many clusters overlap, indicating that different categories are less distinct from each other, meaning that the differentiation between some of the data points in these clusters is not clear. In addition, there are several points that are far removed from the main cluster groups, these could be outliers.

2. DBSCAN

DBSCAN groups together points that are closely packed together while marking points that lie alone in low-density regions as outliers.

There are 2 important parameters for this method, ’epsilon’ and ’min_samples’, where epsilon represent the maximum distance between 2 points for them to be considered part of the same neighborhood. It defines the size of the neighborhood around each point used to identify core points and clusters and min_samples indicates the minimum number of points required to form a dense region, which is used to classify a point as a core point.

The **first feature** was TF-IDF representation and furthermore, performed the grid search in order to obtain the best combination of parameters.

epsilon	min_samples	silhouette_score
0.1	2	-0.250195
0.1	3	-0.262983
0.1	4	-0.267581
0.1	5	-0.269146
0.1	6	-0.271068
0.15	2	-0.250009
0.15	3	-0.262983
0.15	4	-0.267581
0.15	5	-0.269146
0.15	6	-0.271068
0.2	2	-0.249682
0.2	3	-0.262642
0.2	4	-0.267176
0.2	5	-0.268742
0.2	6	-0.270824
0.4	2	-0.247174
0.4	3	-0.260755
0.4	4	-0.264963
0.4	5	-0.266764

0.4	6	-0.268414
0.45	2	-0.245902
0.45	3	-0.259757
0.45	4	-0.264463
0.45	5	-0.266646
0.45	6	-0.267869
0.47	2	-0.244996
0.47	3	-0.259818
0.47	4	-0.264414
0.47	5	-0.266585
0.47	6	-0.267822
0.8	2	-0.239072
0.8	3	-0.257661
0.8	4	-0.262243
0.8	5	-0.266067
0.8	6	-0.268047
0.85	2	-0.242077
0.85	3	-0.259899
0.85	4	-0.263912
0.85	5	-0.266827
0.85	6	-0.267427
0.9	2	-0.246254
0.9	3	-0.263986
0.9	4	-0.268741
0.9	5	-0.270138
0.9	6	-0.272005

The **second feature** was Bag of Words representation and furthermore, performed the grid search in order to obtain the best combination of parameters.

epsilon	min_samples	silhouette_score
0.1	2	-0.302347
0.1	3	-0.307351
0.1	4	-0.304865
0.1	5	-0.304895
0.1	6	-0.304636
0.15	2	-0.302347
0.15	3	-0.307351
0.15	4	-0.304865
0.15	5	-0.304895
0.15	6	-0.304636
0.2	2	-0.302347

0.2	3	-0.307351
0.2	4	-0.304865
0.2	5	-0.304895
0.2	6	-0.304636
0.4	2	-0.302347
0.4	3	-0.307351
0.4	4	-0.304865
0.4	5	-0.304895
0.4	6	-0.304636
0.45	2	-0.302347
0.45	3	-0.307351
0.45	4	-0.304865
0.45	5	-0.304895
0.45	6	-0.304636
0.47	2	-0.302347
0.47	3	-0.307351
0.47	4	-0.304865
0.47	5	-0.304895
0.47	6	-0.304636
0.8	2	-0.302347
0.8	3	-0.307351
0.8	4	-0.304865
0.8	5	-0.304895
0.8	6	-0.304636
0.85	2	-0.302347
0.85	3	-0.307351
0.85	4	-0.304865
0.85	5	-0.304895
0.85	6	-0.304636
0.9	2	-0.302347
0.9	3	-0.307351
0.9	4	-0.304865
0.9	5	-0.304895
0.9	6	-0.304636

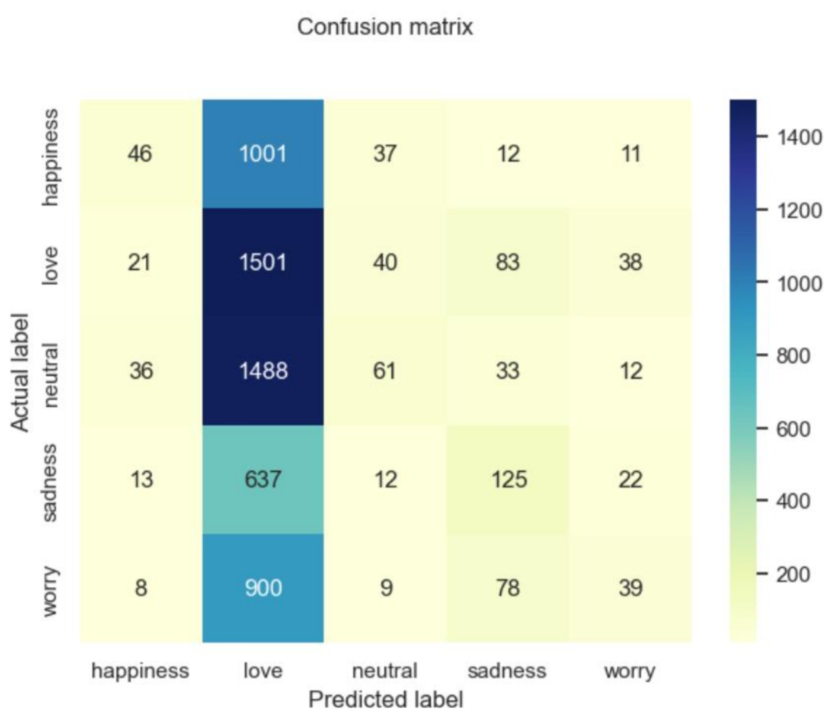
DBSCAN performs better with TF-IDF features than with BOW for the data and parameter settings tested. TF-IDF might be capturing more relevant features for clustering in this context, leading to more distinct and separable clusters, where the highest silouette score is -0.239072.

Judging by the fact that DBSCAN, does not have a predict function, it was implemented a way to compute the metrics, by finding the nearest core sample for each point in the test set and mapping the cluster labels to the most frequent true labels within those clusters.

The classification report with the best configuration, ’eps’ = 0.8; ’min_samples’ = 2, obtained after performing the grid search is:

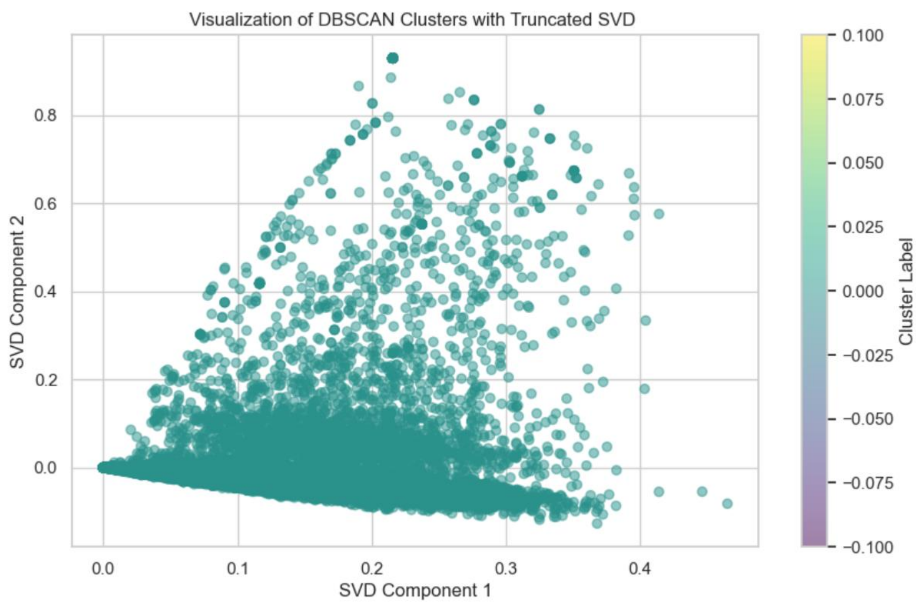
	Precision	recall	F1-score
Happines	0.37	0.11	0.07
Love	0.27	0.51	0.42
Neutral	0.38	0.50	0.07
Sadness	0.38	0.17	0.22
Worry	0.32	0.21	0.07
Accuracy			0.28
Macro avg	0.34	0.23	0.17
Weighted avg	0.34	0.28	0.18

The confusion matrix is:



In column number 2, representing the 'love' predicted label, the highest values are for the actual labels 'love' (1501) and 'neutral' (1488). This indicates that DBSCAN often correctly identifies 'love' sentiments but also frequently misclassifies 'neutral' sentiments as 'love'.

For a better visualization, it is necessary to reduce the dimensionality of the dataset. This procces allows to transform the high-dimensional data into a lower-dimensional space. By applying techniques such as Truncated Singural Value Decomposition, we can obtain the most significant features of the data while simplifying the visualization.



The plot shows a significant number of data that appear as noise. In DBSCAN, noise is represented by points not assigned to any cluster. Despite the high noise, there are some regions where data points are denser and potentially form clusters. However, these clusters are not distinctly visible in the plot.

Conclusion

There are 2 methods approached, K-means and DBSCAN, for a social media dataset for sentiment analysis, highlightinh challenges and areas for improvement. The results demonstrate that while some progress has been made, particularly with the TF-IDF representation showing promise, there remains significant room for optimization and the negative silhouette scores and misclassifications suggest a need for further parameter tuning.