Text Analytics HW3

1. Dataset statistics

The dataset used in this exercise is from Yelp (yelp_academic_dataset_review)
The rating (star) is from 1 to 5 and I transformed it into binary (1: star >3, 0: star<=3)

Summary statistics:
documents: 500,000
labels: 5
Average word length of documents: 108
Label distribution:

label	# of reviews
1	70468
2	40577
3	55778
4	112802
5	220375

binary label	# of reviews
1	333177
0	166823

Relevant code:
https://github.com/MSIA/zzm7646_msia_text_analytics_2020/blob/homework3/dataset_stats.py

2. Logistic regression

Relevant code: https://github.com/MSIA/zzm7646_msia_text_analytics_2020/blob/homework3/logistic_regression.py

Here, I tried different number of grams, which is either 1 gram that takes apart all words, and 1 + 2 gram that could contain both single word and two-word phrase. Also, implemented smaller C: Inverse of regularization strength, smaller values specify stronger regularization. Especially in high dimensionality, it could help to prevent overfitting.

	unigram	unigram (c=0.6)	1+2gram	1+2gram (c=0.6)
Accuracy	0.8589	0.8590	0.8588	0.8589
Precision	0.8792	0.8791	0.8794	0.8792
Recall	0.9138	0.9142	0.9135	0.9138
F1-score	0.8962	0.8963	0.8961	0.8962
Micro-averaged F1 score	0.8589	0.8590	0.8588	0.8589

All the four model actually have very similar performance in terms of all the metrics. In my case, the second one which uses unigram and penalty strength C=0.6 has the best performance.

3. SVM

Relevant code:
https://github.com/MSIA/zzm7646_msia_text_analytics_2020/blob/homework3/svm.py

Here, I tried different number of grams, which is either 1 gram that takes apart all words, and 1 + 2 gram that could contain both single word and two-word phrase. Also, implemented hinge as loss function rather than square hinge and Inverse of regularization strength C. Different from logistic, here I implemented larger C trying to not punish so hard.

	unigram	unigram (loss='hinge',C=10)	1+2gram	1+2gram (loss='hinge',C=10)
Accuracy	0.8588	0.8595	0.8592	0.8595
Precision	0.8780	0.8781	0.8787	0.8786
Recall	0.9164	0.9164	0.9150	0.9156
F1-score	0.8963	0.8969	0.8965	0.8968
Micro-averaged F1 score	0.8588	0.8595	0.8592	0.8595

Using SVM, in default setting, 1+2gram performs better than unigram. For both unigram and 1+2gram models, using loss function as hinge and penalty strength C=0.6 gave better results than default setting and have similar performance, thus we will be using the second model highlighted to predict reviews' rating (1 or 0)

4. fasttext

Relevant code:
https://github.com/MSIA/zzm7646_msia_text_analytics_2020/blob/homework3/fasttext.py

I've tried different combination of learning rate and number of grams. Smaller learning rate can better capture the precise local optimum but could need longer time to converge. Max length of wordNgram is again the length of word representation — 1 means single word and 2 refers to word phrase.

	Learning rate: 1.0 wordNgram: 1	Learning rate: 1.0 wordNgram: 2	Learning rate: 0.8 wordNgram: 2	Learning rate:0.05 wordNgram: 2
Precision	0.9015	0.9116	0.9066	0.9110
Recall	0.9015	0.9116	0.9066	0.9110
F1-score	0.9015	0.9116	0.9066	0.9110

Using fasttext, in default setting, 2 gram performs better than unigram. In my case, model with learning rate 1.0 and using 2 gram gave the best results.

5. Prediction

I used the best SVM model to predict five reviews.

*1: positive, 0: negative
* confidence score in SVM is the signed distance of that sample to the hyperplane (confidence score > 0 indicates label 1, vice versa).

Relevant code:
https://github.com/MSIA/zzm7646_msia_text_analytics_2020/blob/homework3/predict_svm.py

Review	**Actual label**	**Predicted label**	**Confidence score**
I love this restaurant! It is sooooo good!	1	1	2.6968
The food is okay but too pricy.	0	0	-3.8354
The delivery never came. I had to call them to cancel the order.	0	0	-0.1882
The place is nice with large space and nice decoration.	1	1	1.1903
It is easy to park nearby, furnished recently, but too many people in the gym.	0	1	1.0917

The model performed really well with 80% accuracy (4 out of 5), even though the last review is quite ambiguous.