Project Report: Text Classification

Qian Wu

Question 1: Yelp dataset with binary bag-of-words

The results for three models that we considered and two baselines are listed below.

Data: yelp

Bag-of-words: Binary

	Hyperparameter	best_param	validation f-score	training f-score	test f-score
Naïve Bayes	alpha	0.01	0.4389	0.734	0.461
Decision Tree	spliting criterion, tree depth	gini, depth = 3	0.4788	0.4958	0.483
SVM	C	0.01	0.522	0.8497	0.5286
Random	NA	NA	0.1837	0.1836	0.185
Majority	NA	NA	0.5213	0.5251	0.5196

The f1-score from 3 classification models are similar, with SVM slightly outperforms the two others. As for the training f-score, SVM yields the best result on the training data, but that does not generalize much to the test data. Consider this is a five-class classification task, those methods all yield significant better result than random classifiers. However, compared to another baseline, which is majority classifier, the performances are relatively the same.

For improvement, I suggest not constraint the SVM model to linear kernel, using Gaussian kernel here would lead to much better result as this is a more complicated classification task.

Question 2: Yelp dataset with frequency bag-of-words

Data: yelp
Bag-of-words: Frequency

	Hyperparameter	best_param	validation f-score	training f-score	test f-score
Naïve Bayes	--	--	0.3223	0.73815	0.3326
Decision Tree	spliting criterion, tree depth	gini, depth = 7	0.38109	0.4449	0.371
SVM	C	100	0.2738	0.3136	0.3135
Random	NA	NA	0.1879	0.1827	0.1813
Majority	NA	NA	0.5213	0.5251	0.5196

Overall, changing the bag-of-words representation to frequency does not yield better result. The three classifiers still yield very similar results, with decision tree slightly outperform the others. It is noted that the decision tree model here select depth 7 as the max_depth which is much deeper than that for binary representation, but that does not lead to much overfitting to the training data.

For improvement, I suggest the removal of stop words during preprocessing. Stops words tend to have a higher frequency but they deliver nearly no information for the classification task. We could also represent bag-of words using inverse frequency as the words which do not frequently appear in all reviews will be more informative for classification.

Question 3: IMDB with binary bag-of-words

The results for IMDB are similar to that of yelp, in that having frequency bag-of-words representation does not lead to better performance. The relative performances of three classification models are also similar to before. For binary representation, SVM yields the best result. And for frequency bag-of-words representation, decision tree is the best model.

Data: IMDB
Bag-of-words: Binary

	Hyperparameter	best_param	validation f-score	training f-score	test f-score
Naïve Bayes	alpha	0.01	0.844196	0.87205	0.82812
Decision Tree	spliting criterion, tree depth	entropy, depth = 7	0.7451	0.7631	0.7509
SVM	C	0.01	0.8806	0.9658	0.8743
Random	NA	NA	0.492	0.4981	0.5016
Majority	NA		--	--	--

For Binary representation, the overall performance is much better than that of yelp dataset and the three models all outperform random model. This could be that IMBD data set is a two class classification problem while yelp dataset presents a five-class classification problem. As we are constrained to linear kernel SVM, it is more suitable for cases like this.

Data: IMDB
Bag-of-words: Frequency

	Hyperparameter	best_param	validation f-score	training f-score	test f-score
Naïve Bayes	--	--	0.7537	0.5659	0.6758
Decision Tree	spliting criterion, tree depth	entropy, depth = 7	0.7475	0.7682	0.7511
SVM	C	100	0.6289	0.6347	0.6185
Random	NA	NA	0.4998	0.5042	0.5023
Majority	NA		--	--	--