.1em

COMP 550, Fall 2020: Programming Assignment 1

Wei FAN

1 Problem setup

In this assignment, we have two data files containing movies reviews, one for 5331 positive ones and the other 5331 negative ones. We would do sentimental analysis, classify these reviews as either positive or negative. Specifically, I trained four classification models, then compare their performances and conclude which one works well for sentence-level sentiment classification.

2 Experimental procedure

Preprocessing and feature extraction Firstly, I loaded the reviews, randomly divided them into training and testing sets with an 90%-10% split respectively. Secondly, I used the sklearn countvectorizer combined with the TF-IDF transformation to transform text reviews to feature vectors, including preprocessing, tokenization and uni-grams generation, filtering stop-words, stemming and lemmatokenizing, as well as looking at the number of word co-occurrences in the each sample.

Classification Methods After the preprocessing, I chose four Classification models: Logistic Regression, Support Vector Machine with a linear kernel, Naive Bayes classifier, Random Forest classifier.

Building a pipeline I used the sklearn Pipeline to apply sequentially CountVectorizer, TfidfTransformer and classifier.

Parameter tuning using grid search I used sklearn GridSearchCV to searching hyper-parameters for the best cross validation score in the training set. The cross-validation splitting strategy is 5fold.

Evaluating performance I re-used the same pipeline to do the prediction on the same test set and evaluated models performances in terms of prediction accuracy.

3 Parameter settings

I manually designed the hyper-parameters for CountVectorizer, TfidfTransformer and different classifiers and passed them to GridSearchCV to get the best parameters.

CountVectorizer TfidfTransformer I tested different max_df(the threshold for ignoring the token with high document frequency): 0.5, 0.75, 1.0, using stop words or not, using lemmatokenizer or stemmerTokenizer, using idf or not. Over all the models, the best parameter set is: no stop words, using lemmatokenizer, use idf, the max_df can be 0.5 or 0.75.

Logistic Regression I tested the regulation penalty: 'l2' or no penalty, and the different C(Inverse of regularization strength): 0.1, 1.0, 10.0. The best parameter set is C: 10.0, 'l2' penalty.

Support Vector Machine with a linear kernel I tested the different regulation penalty 'l1' or 'l2', the regulation strength constant 'alpha': 1e-2 or 1e-3, as well as the lost function: hinge or squared_hinge. The best parameter set is 'l2' penalty, alpha: 0.001, and the squared_hinge loss function.

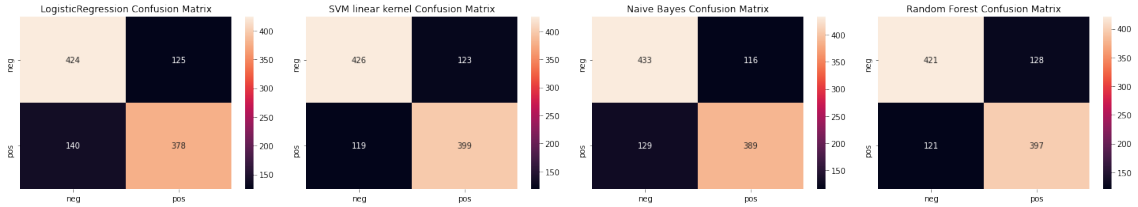
Naive Bayes classifier I tested the different alpha(Additive smoothing): 0.1, 0.5, 1.0. The best parameter set is alpha: 1.0, Laplace smoothing.

Random Forest I tested the different criterion(splitting quality measure function): 'gini' or 'entropy', and max_features(number of feature for best split): 'sqrt' or 'log2'. The best parameter set is max_features: 'log2',criterion':'entropy'.

Table 1: Classifier Performances

Classifier	**Test Accuracy %**	**Training Accuracy %**
Random Guess	49.8	50.3
Logistic Regression	75.2	76.9
Support Vector Machine	77.3	76.3
Naive Bayes	77.0	78.0
Random Forest	76.7	76.0

4 Results and Conclusions



(a) Logistic Regression (b) SVM (c) Naive Bayes (d) Random Forest

Figure 1: Confusion Matrix

Evaluation From Table 1, we can see: the four classifiers have absolutely better performances than Random Guess Baseline. The Support Vector Machine with linear kernel produces the best test accuracy. Naive Bayes is slightly worse.

I have better test Accuracy than train for SVM and Random Forest, It could be because of the random split test/train set, the model fall into a local optimal that happens to be good with the test data.

From Confusion Matrix Figure.1 we can see: The Naive Bayes predict more accurately negative reviews then the other models. While the SVM and the Random Forest models predict more accurately the positive reviews. The Random Forest predictions are biased towards positive for both positive/negative reviews, and the logistic Regression failed more when predicting the positive reviews.