Project 3: Movie Review Sentiment Analysis

sh53, 655723273, SICHENG HE

I use 'pd.read_csv(file, sep='\t', encoding='utf-8')' and 'open(file, encoding='UTF-8')' in Python to read the data. If any encoding error is raised, please contact me since there is no pre-evaluation for this project.

Introduction

Sentiment analysis is a challenging subject in machine learning. People express their emotions in language that is often obscured by sarcasm, ambiguity, and plays on words, all of which could be very misleading for both humans and computers.

We are provided with a dataset consisting of 50,000 IMDB movie reviews, where each review is labelled as positive or negative. The goal is to build a binary classification model to predict the sentiment of a movie review.

The input of the model is movie reviews from some customers and the model will classify those reviews into two groups, positive reviews and negative reviews.

Vocabulary Construction

I pick some terms which can be interpreted easily by their t-statisitcs. Then logistic regression with l1 penalty is applied to reduce the vocabulary size to 999. Please see my HTML file for more details.

Data processing

Set text converter using myvoca.txt

My customized vocabulary list 'myvocab.txt'(999 words) is sent to 'sklearn.feature_extraction.text.CountVectorizer()' to get the text converter.

Get Document Term matrix

After initializing the text converter, I read in the train/test data and remove the html tags in reviews. Then reviews are transformed into DT matrix.

Standardization

'sklearn.linear_model.LogisticRegression' does not standardize the data automatically, so I decide to standardize the DT matrix using 'sklearn.preprocessing.StandardScaler()' to make the logistic regression with l2 penalty converge more stable.

Model fitting & performance

Logistic regression with l2 penalty

I fit a logistic regression with l2 penalty using following parameters, which gives a relatively good result.

- penalty = 'l2'. Add a L2 penalty term.
- C=0.0007. Inverse of regularization strength.
- random_state=2021. Random number seed.
- max_iter=1000. Maximum number of iterations taken for the solvers to converge

Performance and running time

Fitting the logistic regression with l2 penalty does not take too much time. Most of the running time is used to construct the DT matrix from 'myvocab.txt'.

I'm using Python so AUC is computed using 'sklearn.metrics.roc_auc_score'

My computer system: RedmiBook Pro 14, 1.80GHz, 16GB memory, Windows10.

Splits	Running time(s)	AUC
1	69.95	0.9619
2	70.08	0.9606
3	70.02	0.9607
4	70.06	0.9608
5	70.06	0.9601
	Total:350.17	Min: 0.9601

Table 1: Running time and AUC for 5 splits

Discussion

Interpretability of the model

Since I use logistic regression to train a classifier, we can take a look at estimated coefficients.

Based on the 5th split. I choose five words with largest positive coefficients and five words with smallest negative coefficients to briefly describe the interpretability.

Words	coef
great	0.195
best	0.161
excellent	0.152
perfect	0.120
wonderful	0.108

Words	coef
worst	-0.198
awful	-0.175
bad	-0.148
waste	-0.142
poor	-0.134

Each word is assigned a coefficient. If the coefficient is positive, then the presence of this word will increase the probability of such review to be a positive one and vice versa. For example, if a review contains word 'great', probability of being positive is increased since the coefficient is 0.195 here. And if a review contains word 'worst', probability of being positive is decreased since the coefficient is -0.198 here.

You can see that words with positive coefficients are mostly likely to be found in positive reviews while words with negative coefficients are mostly likely to be found in negative reviews. That gives a reasonable interpretability.