Text Sentiment Analysis Project Report: Classification Movie Reviews into Positive or Negative Review base on Machine Learning Models

Pooya Danandeh Bagharabad

Department of Computer Science , University of Tabriz , Tabriz , Iran

Pooya144@gmail.com

Abstract

Reviews for an special movie shows us the movie is a good movie or not. Also movie scores is determined base on this reviews. Because there is millions of movies with thousands of reviews , humans cannot determine movie score for all of movies manually. So it's important to find a system to detect positive or negative reviews. In this report will check out some of methods for this base on Machine Learning(ML).

1 Introduction

Task is an implementation for a system base on ML methods and compare them and use the best one. Implementations are in python with scikit-learn library. We use dataset contains 5000 sentences that most of them are longer than 200 words and its labels(0 for negative and 1 for positive).
For this task we should convert this dataset to a matrix(5000 vector) , this is called Vectorization. We do it with Unigram + Bigram + TF-IDF method.

2 Proposed method:

We proposed SGDClassifier(Linear model with Stochastic Gradient Descent), because with this model we can improve itself like Neural Networks(NN). It has iterations to improve its accuracy and pick up the best state with best score. So it may perform better than other methods.

3 Evaluation

In this part we train this methods on the our train data set and predict on validation data set and test data set.
We use 5 method for classify reviews:

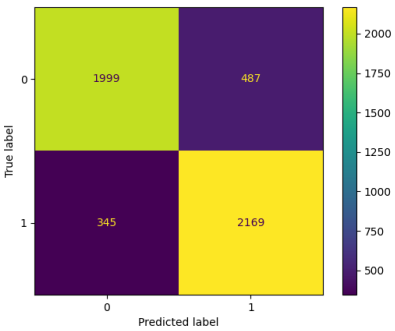
1. Adaptive Boosting
2. Random Forest
3. Logistic Regression
4. Support Vector Machine
5. SGDClassifier

SGDClassifier and Support Vector Machine(SVM) training phase may take too long. SGDClassifier Because set hyperparameter such as Learning-Rate(α) and optimizing loss function with SGD algorithm. And SVM because using non-linear methods.

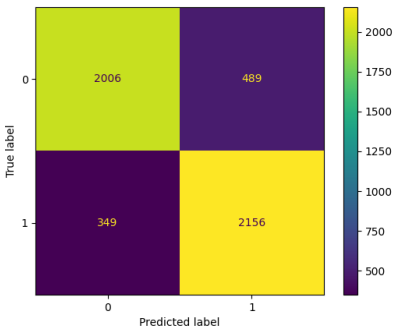
We have check all of these states with python scikit-learn library.

3.1 Adaptive Boosting

Metrics Name	Score (Validation)	Score (Test)
Accuracy	**0.8336**	**0.8324**
F1-Score	**0.8390**	**0.8372**
AUC-Score	**0.8334**	**0.8323**



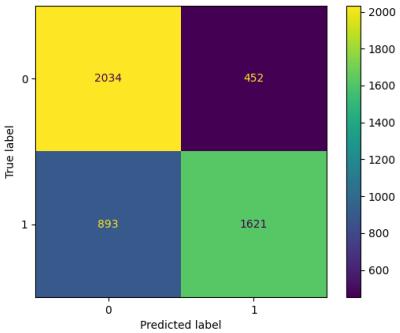
Validation Data



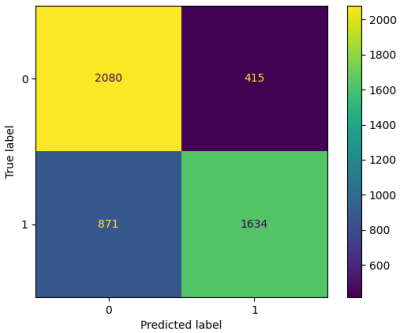
Test Data

3.2 Random Forest

Metrics Name	Score (Validation)	Score (Test)
Accuracy	0.7310	0.7428
F1-Score	0.7067	0.7176
AUC-Score	0.7314	0.7429



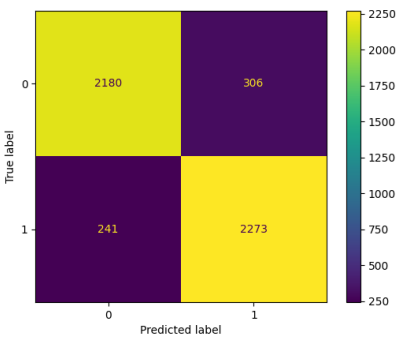
Validation Data



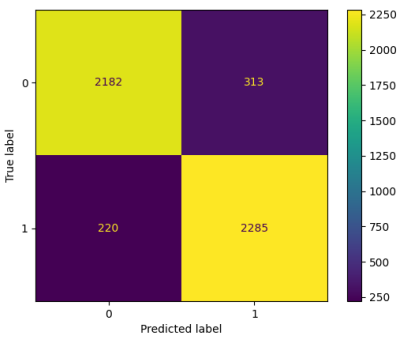
Test Data

3.3 Logistic Regression

Metrics Name	Score (Validation)	Score (Test)
Accuracy	0.8906	0.8934
F1-Score	0.8925	0.8955
AUC-Score	0.8905	0.8933



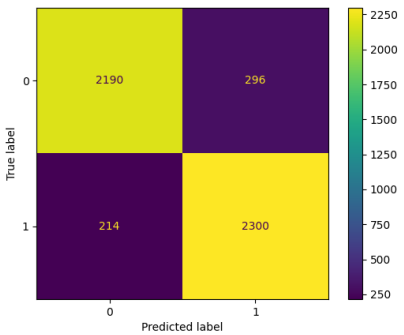
Validation Data



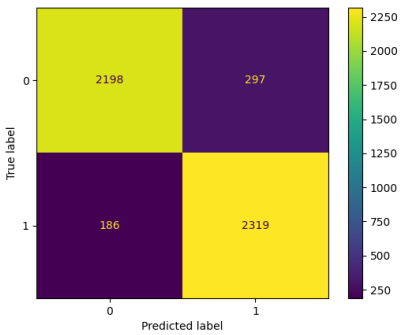
Test Data

3.4 Support Vector Machine

Metrics Name	Score (Validation)	Score (Test)
Accuracy	0.8980	0.9034
F1-Score	0.9001	0.9056
AUC-Score	0.8979	0.9033



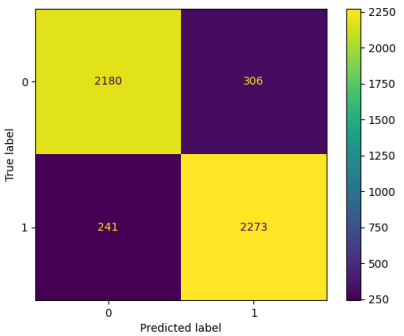
Validation Data



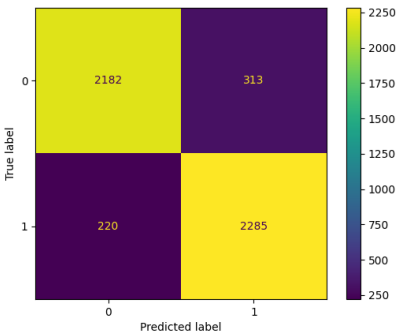
Test Data

3.5 SGDClassifier

Metrics Name	Score (Validation)	Score (Test)
Accuracy	0.9118	0.9148
F1-Score	0.9125	0.9148
AUC-Score	0.9117	0.9147



Validation Data



Test Data

4 Conclusion

This problem is binary classification binary problem , so we split two classes with a line , therefore , it's better use linear classification. It's simple and comprehensible. Linear model may has low accuracy lonely , but with SGD algorithm it can improve its accuracy. As you saw SGDClassifier shows most accuracy.
Comparison between classifiers result:
(This table is base on average of metrics on validation data and test data)

Classifier	Avg Accuracy	Avg F1-Score	Avg AUC-Score
SGD	0.9133	0.9140	0.9132
SVM	0.9007	0.9028	0.9006
Logistic Reg	0.8920	0.8940	0.8919
Adaptive boosting	0.8330	0.8381	0.8328
Random Forest	0.7369	0.7122	0.7371