Sentiment Analysis for Amazon Fine Foods Reviews

Group 4

April, 2019

1 Introduction

Amazon and similar e-commerce websites are used vastly for online shopping purposes and these websites allow their users to write reviews about the products or services they received. These reviews have significant influence on the other users while deciding to buy a product or not. Therefore, it is valuable information to know the essence of a specific product's reviews. Furthermore, a classification made by the information gathered from these reviews can be applied to services such as product summary and product recommendation system. In this project, we intend to classify the usefulness of each product by studying their reviews using different learning models like Supervised Learning, Deep Learning and Unsupervised Learning. Until now, we have implemented Supervised Algorithms like Linear Regression 3.1, Logistic Regression 3.2, Naive Bayes 3.3 (Gaussian 3.3.1, Multinomial 3.3.2 and Bernoulli 3.3.3) and k-NN 3.4.

2 Background Information

The comment texts in the dataset have been preprocessed. The texts have been changed to lowercase and removed non alphabatic characters and stop words.
The positive and negative classes are generated by the ratings which are given by the users. We have used rating 4 and 5 as positive and 1, 2, 3 as negatives. There are 443,777 positives and 124,677 negatives in the dataset. This imbalance situation has caused a problem for prediction approximately every comment as positive. Thus, we have randomly selected 124,677 positive comments to combine with negatives. The dataset is under sampled and the imbalance problem has solved.
The dataset has randomly split into 2 pieces for train and test. %75 of the data is used for train and remaining %25 is used for test.

3 Completed Algorithms

TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative

$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{recall} = \frac{TP}{TP + FN}$$

$$\text{f1-score} = 2 \times \frac{precision \times recall}{precision + recall}$$

$$\text{accuracy} = \frac{TP + TN}{all}$$

3.1 Linear Regression

LinearRegression from scikit-learn is used.

Confusion Matrix	
TPs: 15650	FPs: 15516
FNs: 15633	TNs: 15540

	precision	recall	f1-score	support
positive	0.50	0.50	0.50	31166
negative	0.50	0.50	0.50	31173

Accuracy = 0.50032

3.2 Logistic Regression

LogisticRegression from scikit-learn is used. We have used SAGA[1] algorithm to solve the optimization problem of logistic regression.

Confusion Matrix	
TPs: 15655	FPs: 15511
FNs: 15633	TNs: 15540

	precision	recall	f1-score	support
positive	0.50	1.50	0.50	31166
negative	0.50	0.50	0.50	31173

Accuracy = 0.50040

3.3 Naive Bayes

3 different Naive Bayes classifiers are used to make predictions.

3.3.1 Gaussian

GaussianNB from scikit-learn is used.

Confusion Matrix	
TPs: 19195	FPs: 11971
FNs: 19082	TNs: 12091

	precision	recall	f1-score	support
positive	0.50	0.62	0.55	31166
negative	0.50	0.39	0.44	31173

Accuracy = 0.50186

3.3.2 Multinomial

MultinomialNB from scikit-learn is used. The training data has been transformed to TF-IDF (Term Frequency–Inverse Document Frequency).

$$TF(t) = \frac{\text{Number of times term t appears in a review}}{\text{Total number of terms in the document}}$$

$$IDF(t) = ln\frac{\text{Total number of comments}}{\text{Number of comments with term t in it}}$$

Confusion Matrix	
TPs: 15978	FPs: 15188
FNs: 15930	TNs: 15243

	precision	recall	f1-score	support
positive	0.50	0.51	0.51	31166
negative	0.50	0.49	0.69	31173

Accuracy = 0.50082

3.3.3 Bernoulli

BernoulliNB from scikit-learn is used.

Confusion Matrix	
TPs: 17071	FPs: 14095
FNs: 16951	TNs: 14222

	precision	recall	f1-score	support
positive	0.50	0.55	0.52	31166
negative	0.50	0.46	0.48	31173

Accuracy = 0.50198

3.4 k-NN

KNeighborsClassifier from scikit-learn is used. Because of the slowness of the algorithm, number of neighbours has been selected as 2.

Confusion Matrix	
TPs: 7536	FPs: 23630
FNs: 7670	TNs: 23503

	precision	recall	f1-score	support
positive	0.50	0.24	0.33	31166
negative	0.50	0.75	0.60	31173

Accuracy = 0.49790

4 Remaining Algorithms

1) Decision Tree
2) SVM
3) Random Forest
4) K-Means
5) Deep Learning
6) Neural Networks

5 Work Division

- Boran Yildirim: preprocessing, Linear Regression, Logistic Regression
- Deniz Evrensel: Gaussian Naive Bayes
- Batihan Akca: Multinomial Naive Bayes
- Furkan Arif Bozdag: Bernoulli Naive Bayes
- Sekip Kaan Ekin: k-NN

References

[1] SAGA – Defazio, A., Bach F. & Lacoste-Julien S. (2014). SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives https://arxiv.org/abs/1407.0202