Multi-Class Sentiment Analysis on Amazon Fine Food Reviews

Nikolaos Gounakis
Department of Computer Science
University of Crete
Voutes University Campus, 700 13 Heraklion, Crete, Greece
nicolaig@csd.uoc.gr

Abstract

In Web applications it is useful to understand various concepts of text. One of them is the ability to understand if a piece of text provides a negative, a positive or a neutral meaning. That can result to a new way of producing analytics for big companies when analyzing reviews or social media posts. In this study we compare TF-IDF [6] and Doc2Vec [1] for feature extraction of text data as the study [2]. We evaluate them both on multi-class and binary classification comparing 5 machine learning algorithms, using the Amazon Fine Food reviews dataset [3].

1 Introduction

Text classification is challenging process. Machine learning techniques such as feature extraction enables us to map high dimensional data to a smaller number of variables. In our case text is the high dimensional input and using TF-IDF and Doc2Vec we will extract variables that represent this text in a lower dimension.

The Growth of modern web apps, such as Tweeter, Facebook and the million posts of users have brought in the surface the need of these big companies to analyze and gain information of these posts or reviews in our case. That will enable the companies to have better analytics (analyze feedback) and also spotting users that create unwanted content in the platform.

2 Dataset: Amazon Fine Food Reviews

This dataset consists of reviews of fine foods from Amazon. The data span a period of more than 10 years, including all 568,454 reviews up to October 2012. Reviews include product and user information, ratings, and a plain text review. It also includes reviews from all other Amazon categories. https://www.kaggle.com/snap/amazon-fine-food-reviews

The main columns of the dataset that we focus:

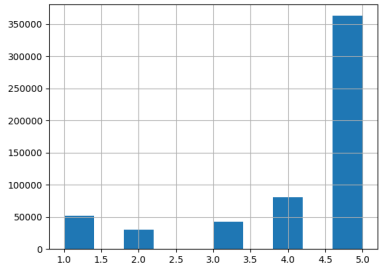
- Score: ranges from 1 to 5 indicating good or bad review (int)
- Text: the actual review (String)

3 Methodology

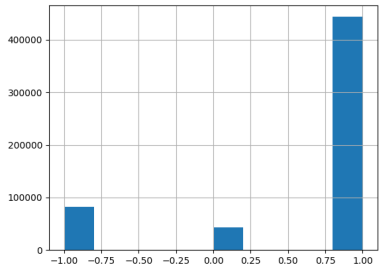
As training data we will use the extracted features from the **Text** Column of the dataset 2 and labels from the **Score**. We consider a **positive** review with a Score of 4 or 5, a **negative** review with a Score of 1 or 2 and a **neutral** review with a Score of 3. We can see how the count distribution changes for

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

the Score from 1a to 1b. Now the lowest count belongs to the neutral reviews with a total of 42.640 reviews. We sample another 42.640 negative and positive reviews and constructing a dataset of size 127.920.



(a) Score value distribution before



(b) Score value distribution after dimensionality reduction

Figure 1: Score value count distribution

3.1 Feature Extraction

Now we are ready to perform the feature extraction using the two techniques we mention earlier (TF-IDF [6], Doc2Vec [1]). We must mention that due to the size of the features we couldn't make any plots of the exported datasets.

3.1.1 TF-IDF

TF-IDF [6] is short form for term frequency-inverse document frequency.TF-IDF is one of the largely used methods in information retrieval and text mining.TF-IDF is a weight metric which determines the importance of word for that document.

We use the sklearn [4] implementation of the TF-IDF vectorizer. We use the hyperparameters proposed by the authors of the paper [2] with a slight change to min_df and max_df,The configurtation is displayed in table 1.

We feed the vectorizer with the **Text** column, and extracting 638 features per row, resulting in a 638*127.920 shape of the final dataset.

min_df	0.01
max_df	0.8
encoding	utf-8
sublinear_df	True
use_idf	True
stop words	English

Table 1: Hyper parameters for TF-IDF Vectorizer

3.1.2 Doc2Vec

Is an extended version of word2vec, Doc2Vec model was put forward the study [1] to improve the learning of embeddings from word to word sequences. Doc2Vec can be applied for word n-gram, sentence, paragraph or document. Doc2Vec is a set of approaches to represent documents as fixed length low dimensional vectors.

We used the Genism [5] implementation of and the proposed hyperparameters [2] displayed in table 2.

The output dataset consists of 100 features per row resulting in a 100*127.920 shape.

min count	1
window size	10
vector size	100
sample	1e-4
negative	5
workers	7
dm	1

Table 2: Hyper parameters for Doc2Vec Vectorizer

4 Experiments

In this section we evaluate the 5 classifiers (KNN, SVM-rbf, Naive Bayes, Random Forests, Logistic Regression) over the produced datasets. We used the sklearn implementations of the classifiers with default hyperparameters. Because of the large number of samples we randomly select 10.000 samples in total, with equal class distribution, and we perform a stratified 10-fold cross validation. The metrics we compute are: Accuracy, f1 and AUC.

4.1 Multi-Class Classification

In multi-class classification the results were interesting. In table 3 we see that **SVM** and **Logistic Regression** outperform the rest classifiers with **SVM** having the highest scores. We see greater AUC than accuracies because it is calculated for each class and then the weighted average is computed considering the number of samples of each class.

TF-IDF performs 45% better than Doc2Vec in multi-class classification with the SVM classifier, but we see very low scores in general.

Feature Extraction Method	TF-IDF			Doc2Vec		
	Accuracy	F1	AUC	Accuracy	F1	AUC
KNN (n=5)	47.990	46.228	65.965	39.580	39.367	56.897
SVM-rbf	**64.910**	**64.883**	**82.694**	**44.139**	**43.741**	**62.465**
Logistic Regression	63.080	62.941	80.915	37.909	37.087	54.663
RF	40.519	40.473	58.931	40.519	40.473	58.931
NB	36.980	31.211	55.265	36.980	31.211	55.265

Table 3: Multi-Class Classification scores with weighted AUC

4.2 Binary Classification

Because of the low scores than the previous experiment performed we wanted to check how the existence of the neutral class can affect the scores.

We removed the neutral class from the datasets keeping only the negative and positive (Score values {-1,1}) and randomly selected 10.000 samples with equal class distribution. We now perform a Binary classification. Again we used a stratified 10-fold cross validation and computed the same metrics. Because of the equal class distribution using stratified folds the AUC can be ignored.

In table 4 we see again that the SVM classifier has the best overall performance. In comparison with the Doc2Vec, TF-IDF is better by 31%.

Binary Classification performance was better than multi-class by 29%. That states, the neutral class was not well separable from the other classes and caused a problem.

Feature Extraction Method	TF-IDF			Doc2Vec		
	Accuracy	F1	AUC	Accuracy	F1	AUC
KNN (n=5)	71.120	71.888	71.132	58.710	63.232	58.425
SVM-rbf	**83.670**	**83.569**	**83.668**	**63.540**	**66.232**	**63.374**
Logistic Regression	82.780	82.684	82.778	57.980	63.065	57.657
RF	81.580	81.435	81.577	59.949	60.927	59.924
NB	79.260	79.566	79.267	54.190	63.736	53.533

Table 4: Binary Classification scores

5 Colnclusions

Considering the 4.1, 4.2 an SVM with rbf kernel is preferable in this kind of situations, where features are extracted by a TF-IDF, Doc2Vec coming to an agreement with the study [2]. We can say that TF-IDF performs better in extracting features from review-like text. In 4.2 Binary classification we saw that the neutral class caused a problem in 4.1 multi-class classification. Thanks to that in a future work we could focus in to making the neutral class more separable using a projection or feature selection techniques. Finally, we could focus on tuning the SVM-rbf hyperparameters for even better accuracy.

References

[1] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1188–II–1196. JMLR.org, 2014.

[2] Avinash Madasu and Sivasankar E. A study of feature extraction techniques for sentiment analysis, 2019.

[3] Julian John McAuley and Jure Leskovec. From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, pages 897–908, New York, NY, USA, 2013. Association for Computing Machinery.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[5] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA.

[6] Claude Sammut and Geoffrey I. Webb, editors. *TF–IDF*, pages 986–987. Springer US, Boston, MA, 2010.