UNIVERSITÀ
DEGLI STUDI
DI MILANO

Deep Learning on Yelp Reviews

By

Delnavaz Fotouhi

Prof. Dario Malchiodi

Algorithms for Massive Datasets

Department of Computer Science

Sunday 11th February, 2024

Abstract

This project focuses on the prediction of the rating of a review from the Yelp Reviews data set by processing its textual context. To achieve this objective two distinct methodologies were employed.

The main method is to train a multi layer perceptron to predict the ratings. The second method uses Logistic Regression, a statistical model, to predict the review's rating by classifying the reviews into five classes.

By employing these two methodologies, this project aims to provide insights into the effectiveness of deep learning-based approaches and traditional statistical modeling techniques in review rating prediction on the Yelp Reviews data set.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

- The exponential growth of online user-generated content has underlined the need for advanced natural language processing (NLP) techniques to extract meaningful insights from massive datasets.

- The objective of this project is sentiment analysis of reviews. This objective was attained by developing predictive models capable of determining the review's rating based on their textual context.

- The first approach is based on feature extraction from the text using the *word2vec* library in *PySparknlp.ml* and feeding the result to a neural network.

- The second method uses Logistic Regression to classify reviews into five distinct classes and predict their ratings by calculating the probability of a review belonging to each class.

Chapter 2

Data Manipulation

This chapter discusses the steps taken to manipulate and prepare the data before using it for model training.

2.1 Data Acquisition

The Yelp dataset was downloaded from Kaggle as a zip file containing five JSON datasets. The only required data set to reach the project objective was reviews. Therefore, other data sets were dropped after unzipping the main file.

2.2 Column Selection

The only useful columns for the purpose of this project were ”stars” and ”text”. Consequently, other columns were dropped from the data set.

2.3 Text Length Limitation

To simplify the dataset and manage computational resources, texts exceeding 500 characters were removed from consideration. This step aimed to ensure efficiency in subsequent processing stages.

2.4 Data Balancing and Sampling

Addressing the class imbalance is a crucial step in data preparation. The distribution of data over the classes is depicted in a pie chart depicted in Figure 2.1.

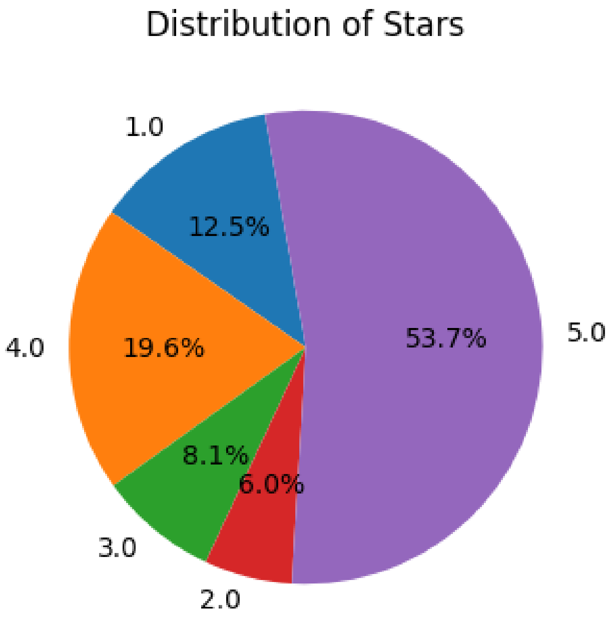


Figure 2.1: Data Distribution

As can be seen in the diagram, over 50 percent of the reviews have 5 stars. This fact indicates the class imbalance in the data set, which can cause some issues during model training.

To solve this issue, data sampling techniques were applied to obtain nearly equal proportions of data for each rating class. The formula to determine the fraction of each class is as follows:

$$FractionofClass = \frac{1}{PercentageofClass} \tag{2.1}$$

This formula indicates that, in the sampled data set, we need to take a smaller fraction of the classes with the larger fraction in the main data set to reach classes with almost equal portions.

This table shows the distribution of data over classes in the sampled data set.

Table 2.1: Distribution of Data over Classes in sampled data set

stars	count
1.0	41619
2.0	42776
3.0	40878
4.0	40696
5.0	44501

Chapter 3

Data preparation

This chapter explains the process of preparing the textual data and the process of transforming text into numerical vectors.

3.1 Text transformation

- The text was broken down into individual words by tokenization.
- Tokens were lemmatized to convert them to their base or dictionary form.
- The lemmatized tokens were normalized for consistency in format and structure. Subsequently, punctuations and digits were removed from the text since they add no sentimental value to the text.
- Commonly occurring words, known as stop words, were removed to reduce noise in the data.

3.2 Word Embedding with Wor2vec

Word Embeddings is the collective name for a set of language modeling and features of learning techniques in Natural Language Processing where words or phrases are represented in the form of real number vectors.[1]

Word2vec is a technique in natural language processing (NLP) for obtaining vector representations of words. These vectors capture information about the meaning of the word and their usage in context. The word2vec algorithm estimates these representations by modeling text in a large corpus.[2]
In this project, Word2Vec takes the preprocessed data (clean text) as input and produces word embeddings for each word in the vocabulary. The 'minCount' parameter is set to 5 which means the word must be repeated in the documents at least five times to be considered. The 'vectorSize' parameter specifies the dimensionality of the word embeddings. In this case, the 'vectorSize' is set to 200 which means the dimensionality of produced vectors is 200.

Chapter 4

MLP Model

This chapter discusses the training and evaluation of a multi layer perception to predict the rating of reviews.

4.1 Model Training

After splitting the data into train and test data sets, the featured vectors in the train data set were fed into an MLP model containing four layers. The size of the first layer must match the dimensions of the featured vectors. Correspondingly, the size of the last layer must match the size of the classes which is five.

4.2 Model Evaluation

Model evaluation is done by using metrics such as accuracy, precision, recall, and f1-score on each class on the test data set.

- Accuracy is one of the most popular metrics in multi-class classification and it is directly computed from the confusion matrix. The formula of the Accuracy considers the sum of True Positive and True Negative elements at the numerator and the sum of all the entries of the confusion matrix at the denominator[3]

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \tag{4.1}$$

The overall accuracy of this model is calculated as 0.56, indicating that approximately 56 percent of the predictions made by the model align with the actual ratings in the test dataset.

- Precision expresses the proportion of units our model says are Positive and they are actually Positive. In other words, Precision tells us how much we can trust the model when it predicts an individual as Positive[3]

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

For this model, the precision values vary between 0.46 and 0.68, with higher values for reviews with 1 and 5 stars.

- The Recall is the fraction of True Positive elements divided by the total number of positively classified units (row sum of the actual positives). In particular False Negative are the elements that have been labelled as negative by the model, but they are actually positive.[3]

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

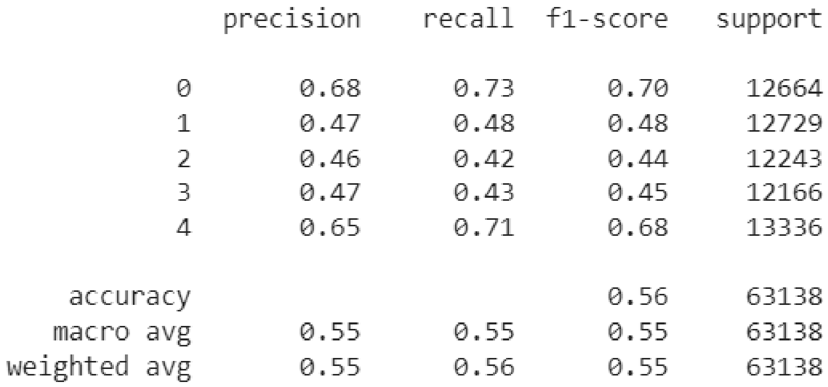
This model reached recall values of 0.73 and 0.71 for reviews with 1 and 5 stars respectively. However, the recall values in the other three classes are approximately 45 percent.

- The F1-Score assesses the classification model's performance starting from the confusion matrix, it aggregates Precision and Recall measures under the concept of harmonic mean. The formula of F1-score can be interpreted as a weighted average between Precision and Recall, where F1-score reaches its best value at 1 and worst score at 0.[3]

$$F1 - Score = 2.(\frac{precision.recall}{precision + recall}) \tag{4.4}$$

The F1-scores range from 0.44 to 0.70, reflecting the overall effectiveness of the model in capturing both precision and recall for each class.

The final result of the model evaluation can be seen in Figure 3.1. The evaluation metrics are calculated for each class. Additionally, the macro average and weighted average are calculated for the whole data set. Since the distribution of sampled data over classes was balanced, the values of macro and weighted average for evaluation metrics are the same.



```
              precision    recall  f1-score   support

           0       0.68      0.73      0.70     12664
           1       0.47      0.48      0.48     12729
           2       0.46      0.42      0.44     12243
           3       0.47      0.43      0.45     12166
           4       0.65      0.71      0.68     13336

    accuracy                           0.56     63138
   macro avg       0.55      0.55      0.55     63138
weighted avg       0.55      0.56      0.55     63138
```

Figure 4.1: Evaluation Results for MLP

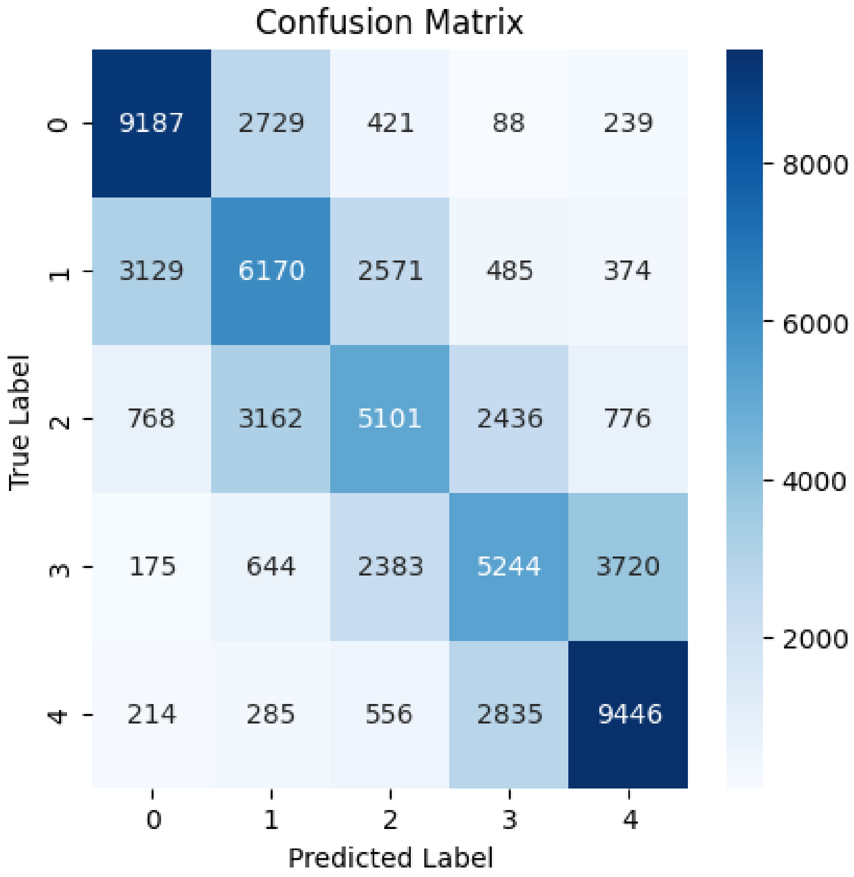


Figure 4.2: Confusion Matrix MLP

Chapter 5

Logistic Regression

5.1 Introduction

Logistic regression is a popular method to predict a categorical response. It is a special case of Generalized Linear models that predicts the probability of the outcomes. In spark.ml logistic regression can be used to predict a binary outcome by using binomial logistic regression, or it can be used to predict a multiclass outcome by using multinomial logistic regression. Use the family parameter to select between these two algorithms, or leave it unset and Spark will infer the correct variant.[4]
In this project, I applied logistic regression to predict the rating of each review. In other words, by using logistic regression, I classified the reviews into five distinct classes(the number of stars are considered as classes).

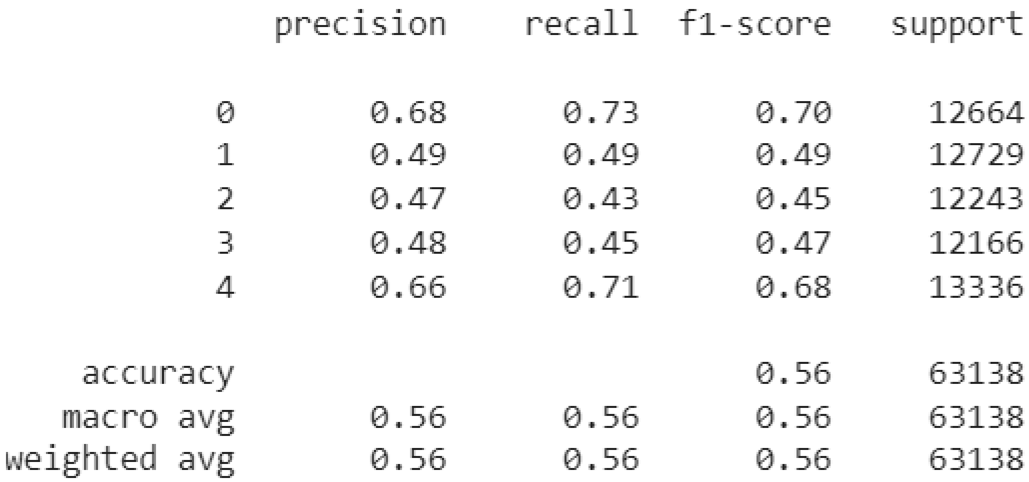
5.2 Model Training

Before training the logistic regression model, several preprocessing steps on the raw text data must be performed, such as tokenization, lemmatization, removal of stop words, and encoding of text data into numerical feature vectors.
These steps have been done previously in this project. Therefore, to train this model, I used the same featured vectors retrieved from the text using word2vec. The model is

trained on the training set.

5.3 Model Evaluation

The performance of the logistic regression model is evaluated on the test data set using metrics such as accuracy, precision, recall, and F1-score. Same as before, these evaluation metrics are calculated separately for each class. In addition to that, the macro average and weighted average are calculated for each evaluation metric.
The final results can be seen in Figure 5.1 and Figure 5.2.

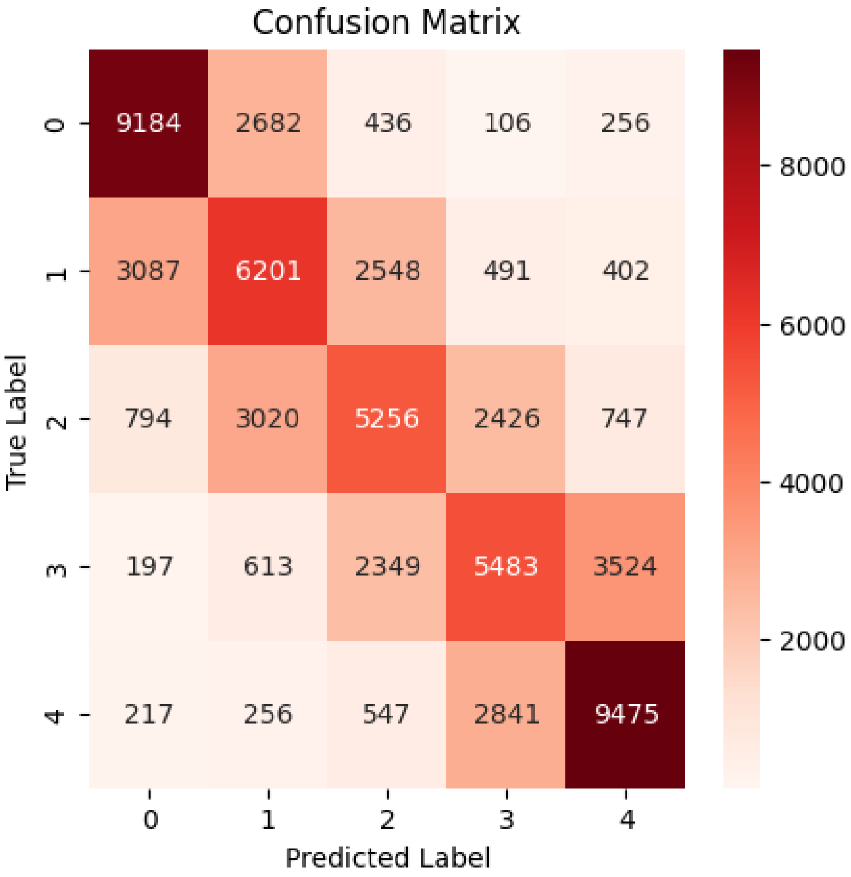


```
              precision    recall  f1-score   support

           0       0.68      0.73      0.70     12664
           1       0.49      0.49      0.49     12729
           2       0.47      0.43      0.45     12243
           3       0.48      0.45      0.47     12166
           4       0.66      0.71      0.68     13336

    accuracy                           0.56     63138
   macro avg       0.56      0.56      0.56     63138
weighted avg       0.56      0.56      0.56     63138
```

Figure 5.1: Evaluation Results for LR



	0	1	2	3	4
0	9184	2682	436	106	256
1	3087	6201	2548	491	402
2	794	3020	5256	2426	747
3	197	613	2349	5483	3524
4	217	256	547	2841	9475

Figure 5.2: Confusion Matrix LR

Chapter 6

Conclusion

In conclusion, logistic regression and multi layer perceptron reached approximately similar results in predicting the rating. The performance of both models was better in reviews with 1 and 5 stars. However, there is room for improvement in reviews with 2,3, and 4 stars. In the latter case, the results of the logistic regression model were slightly better than the MLP model.

[1]

[1] I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me/us or any other person for assessment on this or any other course of study.

Bibliography

[1] A. A. S. Derry Jatnika, Moch Arif Bijaksana, "Word2vec model analysis for semantic similarities in english words," p. 8, 2019.

[2] "Word2vec." https://en.wikipedia.org/wiki/Word2vec#cite_ref-mikolov_1-6.

[3] G. V. Margherita Grandini, Enrico Bagli, "Metrics for multi-class classification: An overview," *Multimedia Tools and Applications*, p. 17, 2020.

[4] "Spark.mllib documentaion." https://spark.apache.org/docs/latest/ml-classification-regression.html.