NLP Assignment 3
Sentiment Analysis

Alberto Parravicini

1 Introduction

The goal of the assigment is to experiment with different techniques used for sentiment analysis, and compare the results obtained.
As a starting point, it was used the dataset of **Digital Music** reviews on **Amazon**, compiled by **Julian McAuley**. The goal was to predict the score given to a product by analysing its review, by taking advantage of *Natural Language Processing* and *Machine Learning*.
This report will present various preprocessing and modelling techniques that have been tried, such as **Latent Semantic Analysis** and **Support Vector Machines**, and discuss their efficiency.

The first section of the report will detail the data that have been used, and the preprocessing techniques applied to them.
The second section is focused on the models that were used for sentiment analysis, and on the selection and validation techniques that have been adopted. The third and last section will present the results of the models, and discuss problems and potential improvements that can be adopted.

2 Data analysis and pre-processing

The dataset used in the assignment is a collection of **Digital Music** reviews of songs and albums sold on **Amazon**.
Each review is stored as a **JSON**, with different fields such as the *reviewer name*, the *review date*, how many people found it *useful*, and more.
Our goal is to process the **review text** (and the **review title**), in order to predict the review score. Scores range on a $1 - 5$ scale, and can be interpreted as the **sentiment** of the reviewer towards the product he has bought.
It should be noted that if our goal was to predict the scores as accurately as possible, then all the information in the review should be taken into account

(such as the *reviewer name*); however, our focus is on the text analysis, and we can discard those fields.

The dataset contains 64.706 reviews, but we will use a smaller subset in order to reduce the computational costs of the algorithms, and to check whether using subsets of higher size can be beneficial.
Indeed, moving from a subset of 10000 entries to a subset of 40000 entries seems to positively impact the quality of the predictions. It can be assumed that using even more data could give further improvements, at the cost of greatly increased execution time of the training.

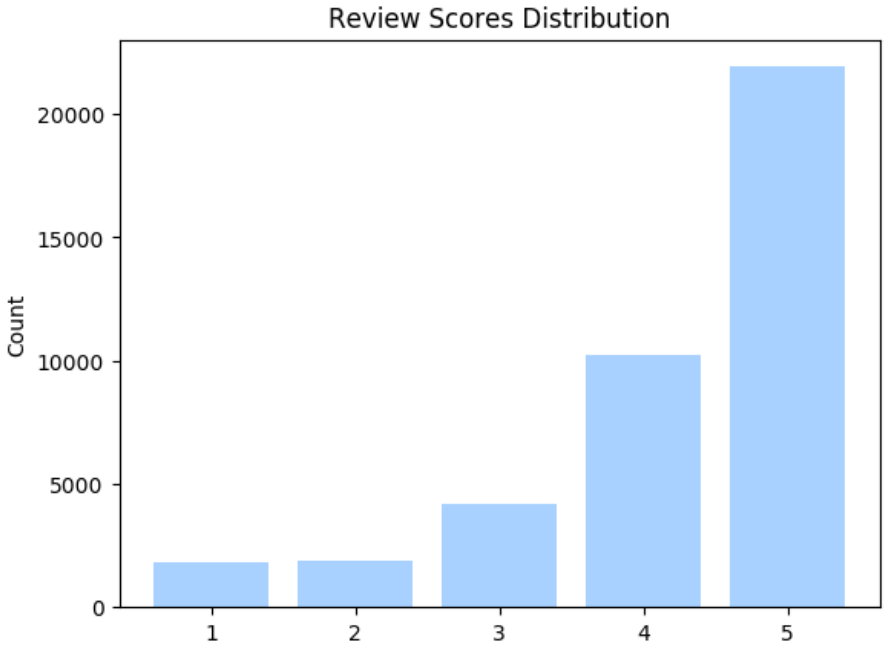

Figure 1: *Distribution of the review scores in the dataset.*

If we look at the distribution of the review scores in the dataset, it is clear how reviews with high scores are overrepresented (it's not common to buy a song one doesn't like). This has a few important implications:

- Any model we use will have to consider the *a-priori* probability of each class, as they are different.
- If our goal is to infer the sentiment of a sentence and not to predict the review score, it could be beneficial to build a new dataset in which every score has the same probability of appearing. However, this would heavily reduce the data at our disposal, and overall lead to worse results.
- Models based on regression are likely to perform badly, compared to multi-class classification models. Most regression models (such as any model based on linear regression) assume the output to be *normally* (or at least symmetrically) distributed, while we have a highly skewed distribution. Models won't be able to accurately predict the values at the extremes of the distribution. On the other hand, multi-class classification will ignore the ordinal relation in the scores, hence using less information that they could.
- A trivial predictor that always give the majority class will have an accuracy of 54% This value is our baseline, and will prove surprisingly hard to beat.

2.1 *Preprocessing*

sad

References

[1] William B Cavnar, John M Trenkle, et al. N-gram-based text categorization. *Ann Arbor MI*, 48113(2):161–175, 1994.

[2] Paolo Frasconi, Giovanni Soda, and Alessandro Vullo. Hidden markov models for text categorization in multi-page documents. *Journal of Intelligent Information Systems*, 18(2-3):195–217, 2002.

[3] Ioannis Kanaris, Konstantinos Kanaris, Ioannis Houvardas, and Efstathios Stamatatos. Words versus character n-grams for anti-spam filtering. *International Journal on Artificial Intelligence Tools*, 16(06):1047–1067, 2007.

[4] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *AAAI*, volume 333, pages 2267–2273, 2015.