LuleåUniversity of Technology
Predictive Analytics

Assignment 2

Replication study

Alghisi Giovanni Angelo

February 10, 2023

Abstract

In this report the results obtained replicating the study [3] will be discussed. In particular, the emulation has been developed by means of RapidMiner. This very powerful tool slightly differs from the ones exploited by Müller et al.; for this reason, and for others that will be presented throughout this essay, the outcomes deviate from the original ones, but allow to present the knowledges that have been acquired during the processing of this assignment.

Contents

1 Some words for the software developed

In this section the structure of the software developed in the RapidMiner environment will be described.[1] In particular, describing the software components it will be possible to depict the reasoning above which the entire analysis lies, pinpointing the differences between this replica and the original study in terms of *data preparation* and *modelling*. For convenience, a subsection will be reserved to each prepared process, but please refer to the comments spread throughout the code itself for further information.

1.1 Opening data

The first process to be run is `1a-opening_data`, that converts the provided `.json` dataset into an `ExampleSet`;[2] this is the starting point of the entire analysis in RapidMiner.

[1] The software is available at [4].
[2] For this analysis it has been used the dataset available on Canvas.

1.2 Filtering and preparing the data

The purpose of this task is to clean the dataset, filtering it, and to prepare the data to the text preprocessing phase.

As discussed in [3], to increase the reliability of the analysis all the reviews with less than two helpfulness ratings have to be removed. Moreover, the process generates the following new features:

- `helpfulness` the target feature obtained evaluating the ratio of the number of positive helpfulness ratings over the number of helpfulness feedbacks (per each review); if this ratio is over 0.5, then the feature assumes `true` value, otherwise `false`.
- `text_corpe`, obtained by the concatenation of review summary and the text itself. it is important to highlight that Müller et al. are a little bit vague, because they talk about "the corpus" of the reviews; for this replica it has been decided to consider both the summary and the actual text of each review.
- `text_corpe_length`, calculated by counting the number of words in the `text_corpe` feature.

1.3 Preprocessing the text

This phase is crucial to sensibly apply the *LDA* algorithm, as it consists of filter the noise as much as possible from the text to be processed. This has been done with the following sequence of operations:

1. *converting all characters to lower-case*;
2. *tokenizing* each `text_corpe` occurrence into single words;
3. *removing stop words*, that is deleting uninformative but frequent words. Actually, the stop words can be classified in two different families:
   - *standard stop words*, and for this particular study the only English stop words have been considered, like "the", "and" or "I"; this is done by
   - *custom stop words*, related to the main topic of our analysis (i.e. video games reviews), like "game" or "play".
4. *stemming*, an operation that consists of reducing a word to its stem, like the words "analyze" and "analysis" to "analy"; in particular, the *Snowball algorithm* has been used.[3]

At this point a clarification is required: the original study performed by Müller et al. relies on the exploitation of the *Lemmatizing* algorithm. This means that words like "dog", "Dog", "dogs", and "Dogs" would all change to "dog" (word are transformed into its dictionary form). The lemmatizing approach is, without any doubt, more gentle, but, due to the fact that RapidMiner does not provide this algorithm, stemming approach has been exploited instead, as suggested by the community itself.

1.4 Preparing video games stop word list

In the last subsection it has been said that the stop words a list of custom stop words should be perpared in order to properly filter them from the text. However, this list has to be prepared, and this is done with the help of this process.

The game-related stop word list is used to filtered the text, but a problem could arise. Indeed, the Stem algorithm is applied to the original `review_corpe` text. This causes the words to be truncated (e.g. "games" could become "game"). Thus, the stop word list include just the the word "games" in our dictionary of game-related stop words list, it will be useless to filter the `review_corpe` text considering it, because the Stem algorithm prevents it to appear in the output (it will appear "game", but this word differs from "games"). In order to improve the robustness of our analysis, the game-related stop words list has been developed applying the same Stem algorithm to the words specified by the user through the .txt file.

[3] Not knowing in deep the differences that different stemming algorithms present, for this work it has been chosen the most suggested by the RapidMiner community.

2 First section

...

First subsection

...

3 Second section

...

References

[1] Stefan Debortoli et al. "Text mining for information systems researchers: An annotated topic modeling tutorial". In: *Communications of the Association for Information Systems (CAIS)* 39.1 (2016), p. 7.

[2] Julian McAuley, Rahul Pandey, and Jure Leskovec. "Inferring networks of substitutable and complementary products". In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 2015, pp. 785–794.

[3] Oliver Müller et al. "Utilizing big data analytics for information systems research: challenges, promises and guidelines". In: *European Journal of Information Systems* 25.4 (2016), pp. 289–302. DOI: 10.1057/ejis.2016.2. eprint: https://doi.org/10.1057/ejis.2016.2. URL: https://doi.org/10.1057/ejis.2016.2.

[4] Alghisi Giovanni Angelo. *GitHub Repository - PA_assignment_2*. 2023. URL: https://github.com/g002alghisi/PA_assignment_2.