Luleå University of Technology
Predictive Analytics

Assignment 2

Replication study

Alghisi Giovanni Angelo

February 12, 2023

Abstract

In this report the results obtained replicating the study [4] 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 development of this assignment.

Contents

1 Some words for the developed software

In this section the structure of the software developed in the RapidMiner environment will be described.[1] In particular, by 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 process prepared, 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 [5].

[2]For this analysis it has been used the dataset available on Canvas and provided by Prof. Jaap van de Beek and Prof. Yomn Elmistikawy.

1.2 Filtering and preparing the data

The purpose of the task `2-filtering_and_preparing_data` is to clean the dataset, filtering it, and to prepare the data for the text preprocessing phase.

As discussed in [4], to increase the reliability of the analysis, all the reviews with less than two helpfulness ratings should 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 total amount 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. Thus, `3b-processing_text` deals 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 to delate 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 has been done by means of already implemented primitives, available in RapidMiner;
    - *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 list of custom stop words should be prepared in order to properly filter them from the text; this is done with the help of `3a-preparing_videogame_stopword_list` process. Indeed, a problem could arise. 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, if the stop word list include just the word games (without "game"), it will be useless to filter the `review_corpe` text considering it, because the Stem algorithm prevents it to appear in the output (maybe "game" would arise there, or again "gam", but this words differ from "games"). In order to improve the robustness of our analysis, the list has been produced applying the same Stem algorithm to the words specified by the user through a `.txt` file. In this way, the mechanism is more robust and easier to be used.

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

1.5 LDA execution

The process `3c-LDA_execution` does exactly what the name suggests: by means of an already-arranged function, it is possible to recognise topics using the LDA method. The algorithm has been tuned to search for 100 topics and consider 10 words per each one.

Considering these results, the dataset has now, in addition to `overall` and `text_corpe_length`, 100 new descriptive features consisting in the probability of each instance to belong to the related topic. What about the list of 10 words per topic, it has been used to entitle each topic itself in order to simplify the result analysis.

1.6 Building and evaluating the Random Forest

Finally, by means of `4-building_random_forest`, it is possible to train and evaluate the *Random Forest* algorithm.

As suggested by the article, the dataset has been split as follow:

- 80% of the instances for the *training set*;
- 20% of the instances for the *testing set*.

Considering that the *stratified sampling* approach is not required in this case, the *random sampling* technique has been preferred over the *linear sampling* one in order to avoid the risk of introducing bias. Indeed, due to the fact that the ordering problem of instances has been neglected, linear sample could lead to negative results.

RapidMiner allows the user to deeply tune the random forest algorithm; the main choices taken for this study are the followings:

- as Müller et al. have done, a model consisting of 128 trees trained on bootstrapped sub-sets of the *training set* has been used (*bagging*); the prediction strategy chosen in case of dissenting tree model predictions is the *confidence vote* one, thus the class that has the highest accumulated confidence.
- the *Gini-index* measure of impurity has been selected *to make the trees grow*, in accordance to Müller et al.;
- a variety of *prune* techniques can be exploited, in order to prevent the predictive model to overfit the training set; the only strategy applied in this context during the present project has been to set the *maximal depth* parameter to 10, in order to limit the depth of each random tree.

The performances of the model can then be evaluated using the *testing set*, and the output of this process consists in both the *weight of each feature* and some performance indexes, like *precision*, *accuracy*, *recall* and *AUC* and *ROC* plots.

2 Results

2.1 The main performance indexes

The time comes to analyse the results obtained testing the model developed. As first index, we can consider the *accuracy*, calculated directly in RapidMiner: the value of 77.48% results, and this means that the model is able to guess the correct value of the target feature about 77.5 times over 100 trials. However, even if this seems a good result, is necessary to partially consider this achievement.

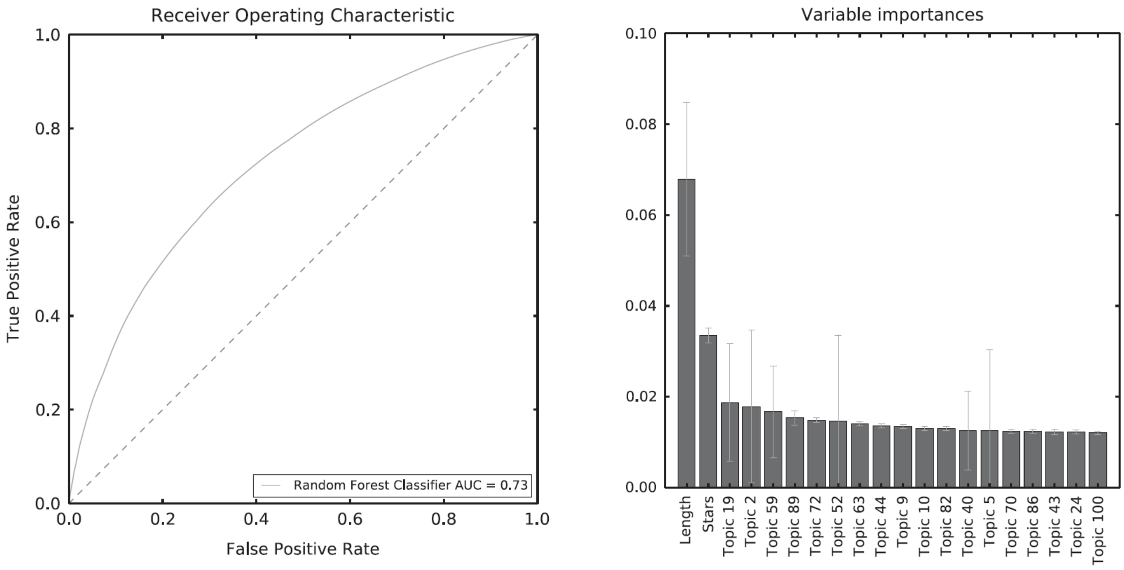
The final dataset (used both to train and test the model) is indeed imbalanced: there are 362,246 positive helpful reviews, and 106,934 negative ones. This, considering also the fact that accuracy does not take into account the cost of an incorrect prediction, could lead to prefer a different metrics to gauge the performances of the resulting model. Often the *AUC* index, i.e. the Area Under the ROC Curve, is more sensible, as it can be viewed as a measure of *separability.* Lets thus consider Figure 2a and compare it with the ROC curve obtained by Müller et al. The AUC got is 69.2%, slightly less than the one of the authors (73%).

However, the *best* parameter does not exist: the designer of the model should always consider the purpose of it. If, for instance, the model was developed by Amazon to rank the reviews to select and propose them to people visiting the e-commerce, than the primary goal would be to pick out *only* the helpful reviews. In this situation, the model should be taken in order to maximise its *precision*, i.e. the fraction of positive predictions that are actually positive (in terms of helpfulness). Indeed, the main aim of the company would not be to choose a model able to correctly distinguish a good review from a bad one (supposing that the pool of available reviews is big enough), but to propose the purchaser less useless reviews as possible. In this cases comes in hand the *confusion matrix* (Table 1), and in particular the positive class of precision, which its value is of 77.54%.[4]

2.2 Variable importance

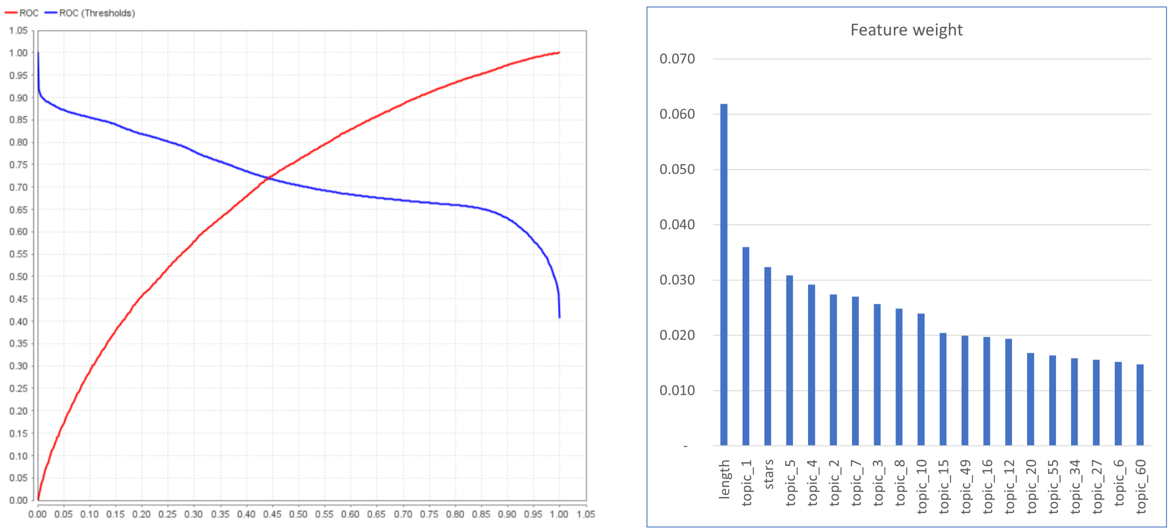
As stated in the article, random forest algorithms are very obscure and cryptic, preventing the person who implemented them to find the cause-effect relationship between the describing features and the target one. However, some insight can be given by the analysis of weights possessed by each descriptive feature. Lets thus consider and compare Figures 2a and 1b which show the impact of the twenty most affecting features. The first thing that can be noticed is the (relative) big weight of the `review_corpe_length`; both the models show that the longest the review, the most helpful it is. For what concerns the other features, it is very difficult to do comparisons because this work relies on several different decision taken with respect the ones of [4], like the exploitation of the stemming algorithm instead of the lemmatizing one.

[4]The value of accuracy and positive class precision are very close due to the fact that the dataset is imbalanced and most of the reviews are helpful.



(a) (b)

Figure 1: Results obtained during this study.



(a) (b)

Figure 2: Results obtained during this study.

Table 1: Confusion matrix obtained during study.

	True false	True true	Class precision
Pred. false	224	139	61.71%
Pred. true	20994	72479	77.54%
Class recall	1.06%	99.81%	

Figure 3: Topic list of the eighteen most relevant topics obtained by Müller et al.

Topic	*Most probable words ordered by probability*	*Interpretation*
19	bad buy make money graphic terrible horrible waste good before	Negative affect
2	call duty cod map multiplayer ops good black battlefield campaign	Call of Duty series
59	dont good buy thing alot didnt bad make people doesnt	Don't buy it
89	ea city drm server buy internet online simcity make connection	Digital rights management
72	great graphic amaze love awesome recommend story good gameplay buy	Gameplay and storyline
52	gta city de mission la grand auto theft car el	Spanish reviews
63	love son gift buy christmas great year grandson purchase daughter	Gift for a family member
44	good pretty graphic fun great bad nice cool thing lot	Graphics
9	work download window computer install run steam software problem instal	Installation problems
10	money buy worth waste time spend save pay good work	Not worth the money
82	fun great lot love good time recommend enjoy challenge easy	Fun and enjoyment
40	amazon customer send work product return back service support receive	Customer service
5	halo mutliplayer xbox campaign good reach great stroy map weapon	Halo series
70	great work good love buy prices product recommend awesome deal	Quality of console hardware
86	price buy great worth good pay amazon deal cheap find	Good deal
43	love awesome cool buy fun great good thing graphic make	Positive affect
24	review star give read buy people rate write good bad	Other customers' reviews
100	kid love fun year child young great adult age enjoy	Appropriateness for kids

Table 2: Topic list of the eighteen most relevant topics obtained during this study.

Topic	Most probable stems ordered by probability	Interpretation
1	sim expans pack hous make build creat get stuff lot	The Sims series
5	peopl say know think review want whi thing get make	Game recommend by friends
4	stori charact scene voic act cut end movi gameplay plot	Plot
2	fallout oblivion quest skyrim max world bethesda scroll elder morrowind	The Elder Scrolls series
3	charact battl final fantasi rpg stori system rpgs time squar	Final fantasy series
7	love old year kid son bought great fun daughter christma	Game for children
8	get want thing make time good use peopl take find	Good review - general comments (1)
10	love great amaz awesom graphic buy get recommend perfect fan	Very good graphics
49	fun great lot recommend enjoy good love get time high	Good review - general comments (4)
15	amazon order product work ship purchas receiv item return box	Good service provided by Amazon
12	hour fun get time day bore got beat enjoy keep	Ideal game to kill time
16	dont get buy good cant alot that thing say got	Bad review - general comments (3)
55	player allow featur option set abil addit system chang includ	Extra features and options
20	get time see look move run thing make head take	(???)
34	money buy wast time worth bought want spend worst piec	Bad review - waste of money
27	got work bought tri day went start buy thought came	Good review - general comments(3)
6	keyboard key use type light mous macro switch press usb	Review related to peripherals and controllers
60	good pretti get graphic bad great thing nice look lot	Good review - general comments (5)

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] J.D. Kelleher, B.M. Namee, and A. D'Arcy. *Fundamentals of Machine Learning for Predictive Data Analytics, second edition: Algorithms, Worked Examples, and Case Studies.* MIT Press, 2020. ISBN: 9780262361101. URL: `https://books.google.se/books?id=UM%5C_tDwAAQBAJ`.

[3] 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.

[4] 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`.

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