Using Machine Learning to Predict the Users Ratings on TripAdvisor Based on Their Reviews

Chapter · September 2021

DOI: 10.1007/978-3-030-85710-3_11

4 authors, including:

João Carneiro
Polytechnic Institute of Porto
52 PUBLICATIONS **279** CITATIONS
SEE PROFILE

Jorge Meira
University of A Coruña
11 PUBLICATIONS **34** CITATIONS
SEE PROFILE

Paulo Jorge Novais
University of Minho
570 PUBLICATIONS **3,556** CITATIONS
SEE PROFILE

Using Machine Learning to Predict the Users Ratings on TripAdvisor Based on their Reviews

João Carneiro[1[0000-0003-1430-5465]], Jorge Meira[1,2[0000-0002-1502-780X]], Paulo Novais[3[0000-0002-3549-0754]] and Goreti Marreiros[1[0000-0003-4417-8401]]

[1] GECAD – Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Institute of Engineering, Polytechnic of Porto, 4200-072 Porto, Portugal
{jrc,janme,mgt}@isep.ipp.pt
[2] CITIC – Centro de Investigación en Tecnologías de la Información y las Comunicaciones, University of A Coruña, 15071 A Coruña – Spain
[3] ALGORITMI Centre, University of Minho, Guimarães 4800-058, Portugal
pjon@di.uminho.pt

Abstract. Argumentation-based dialogue models have shown to be appropriate for decision contexts in which it is intended to overcome the lack of interaction between decision-makers, either because they are dispersed, they are too many, or they are simply not even known. However, to support decision processes with argumentation-based dialogue models, it is necessary to have knowledge of certain aspects that are specific to each decision-maker, such as preferences, interests, limitations, among others. Failure to obtain this knowledge could ruin the model's success. In this work, we intend to facilitate the acquiring information process by studying strategies to automatically predict the tourists' preferences (ratings) in relation to points of interest based on their reviews. We explored different Machine Learning algorithms (Logistic Regression, Random Forest, Decision Tree, K-Nearest Neighbors and Recurrent Neural Networks) and Natural Language Processing strategies to predict whether a review is positive or negative and the rating assigned by users on a scale of 1 to 5. The experiments carried out showed that the developed models can predict with high accuracy whether a review is positive or negative but have some difficulty in accurately predicting the rating assigned by users.

Keywords: Machine Learning, Natural Language Processing, Sentiment Analysis, Argumentation-based Dialogues, Tourism, TripAdvisor.

1 Introduction

Argumentation-based dialogue models are extremely useful in contexts where a group of agents is intended to find solutions for complex decision problems using negotiation and deliberation mechanisms [1-3]. In addition, they allow human decision-makers to understand the reasons that led to a given decision (enhancing the acceptance of decisions) and to define mechanisms for intelligent explanations [4, 5].

These models receive the decision-makers' preferences as input (for instance, regarding criteria and alternatives) that are typically used to model the agents that represent them [6]. However, obtaining these preferences is not a simple process: first, in the contemporary and highly dynamic world in which we live, it is less and less comfortable for decision-makers to answer questionnaires and second, it is sometimes difficult to express preferences through questionnaires [7, 8]. To facilitate this task, strategies that aim to automatically identify the users' preferences have been proposed. One of those strategies consists in using Machine Learning (ML) algorithms and Natural Language Processing (NLP) to automatically extract from a text corpus the users' opinion through different strategies such as: text wrangling and pre-processing, named entity recognition and sentiment analysis [9, 10]. However, there are many algorithms and strategies that can be applied. Therefore, it is mandatory to develop specific procedures according to the application topic, to achieve the best results.

In this work, we studied the problem previously described under the topic of group recommendation systems, more specifically in the context of tourism, in which there has been an increased interest in the development of technologies capable of making recommendations according to the interests of each group member. We assumed as habitual that users/tourists express their opinions regarding Points of Interest (POI) on social networks (such as, TripAdvisor, Facebook or Booking.com) and we intend to take advantage of that to automatically predict their preferences non-intrusively. For this, we used a public dataset (available in Kaggle) and applied the development lifecycle for intelligent systems using concepts of NLP defined in [11]. More specifically, we developed forecast models using 5 ML algorithms (Logistic Regression, Random Forest, Decision Tree, K-Nearest Neighbors and Recurrent Neural Networks), using each of them both as a classification and regression methods. In addition, we used NLP to extract more knowledge from the users reviews and various libraries of Sentiment Analysis (Vader, TextBlob and Flair) to find those that best fit this context.

The rest of the paper is organized in the following order: the methodology is presented in the next Section and in the last Section some conclusions are put forward alongside with suggestions of work to be done hereafter.

2 Methods

In this Section, we describe the methodology in detail. We start by enlightening the problem that we intend to address. Next, we justify the choice of the dataset, carry out its analysis, cover preprocessing and feature engineering. Finally, we approach the used computational techniques and describe the tests and results obtained.

2.1 Understand the Problem Statement

The problem we want to overcome is to predict, non-intrusively and with a high level of accuracy, how much a tourist likes/dislikes a given POI. Subsequently, we intend

to use the predicted preferences to model intelligent agents that represent tourists in a group recommendation system, who seek to jointly decide (using an argumentation-based dialogue model) and recommend to the group of tourists the set of POI to visit. For this, we chose to use the reviews that tourists write in social media (TripAdvisor) to predict their preferences.

2.2 Collect dataset

The chosen dataset was selected based on 2 criteria: it needed to be a public dataset and should best represent the context in which this work intends to be applied. Therefore, a dataset available at Kaggle[1] and which is composed by more than 20 thousand hotel reviews extracted from TripAdvisor was selected. The fact that there already are many works that use this dataset allowed us to know beforehand that it would be very difficult to get good results, since, for example, for 5-class problem the presented accuracy of the large majority varies between 30% and 60%.

2.3 Analyze dataset, preprocessing and feature engineering

The dataset is composed by the attributes "Review" and "Rating". Table 1 shows some examples of the type of records that make up the dataset. The "Rating" is between 1 and 5, where 1 is the worst and 5 is the best possible evaluation.

Table 1. Small example of the used dataset.

Review	Rating
nice hotel expensive parking got good deal sta...	4
ok nothing special charge diamond member hilto...	2
nice rooms not 4* experience hotel monaco seat...	3
unique, great stay, wonderful time hotel monac...	5
great stay great stay, went seahawk game aweso...	5

The dataset consisted of 20491 records and 2 columns, and it did not have any missing data. Fig. 1 shows the distribution by "Rating". As it is possible to verify, the dataset is quite unbalanced, with many more records with positive evaluation (Rating 5: 9054; Rating 4: 6039) than with negative evaluation (Rating 2: 1793; Rating 1:1421). Furthermore, the number of records with intermediate evaluation is also much lower than the number of records with positive evaluation (Rating 3: 2184).

[1] https://www.kaggle.com/andrewmvd/trip-advisor-hotel-reviews

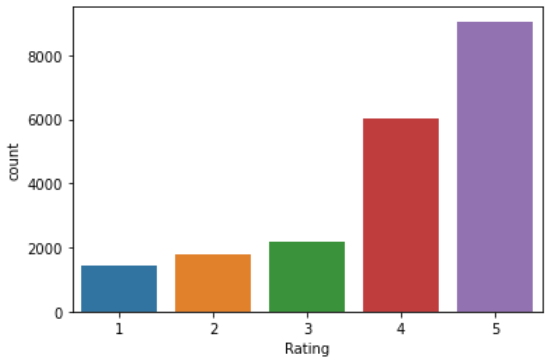

Fig. 1. Distribution by "Rating".

To study possible correlations between the "Review" and the assigned "Rating", we created 3 new attributes: "Word_Count", "Char_Count" and "Average_Word_Length". The "Word_Count" stands for the number of words used in the "Review", the "Char_Count" stands for the number of characters used in the "Review" and the "Average_Word_Length" stands for the average size of the words used in the "Review". The "Average_Word_Length" did not show statistical relevance, but we found that the most negative reviews tended to be composed of more words than the most positive reviews (Fig. 2), which made us believe that the attribute "Word_Count" would be very relevant for the creation of the model.

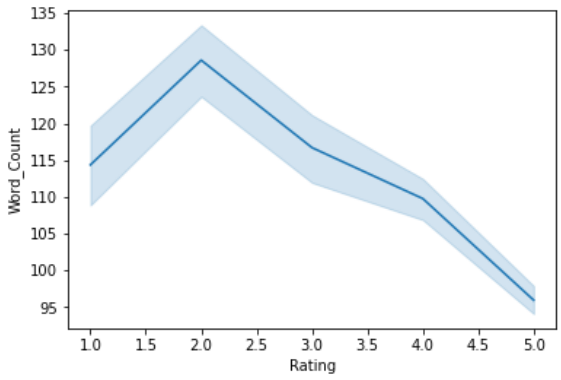

Fig. 2. Correlation between the average number of words in the "Review" with the assigned "Rating".

In the next step, we analyzed which words were most used in the reviews. In addition, we analyzed which words were most used in negative reviews (Rating 1 and 2) and in positive reviews (Rating 3, 4 and 5). We found that many of the most used words were the same, both in positive and in negative reviews. In Table 2 are presented the most used words considering all the reviews. The fact that many of the

most used words are the same, in both positive and negative reviews, made us wonder if eliminating these words would be a good strategy in creating the model.

Table 2. List of the most used words in reviews.

Word	#	Word	#	Word	#	Word	#	Word	#
hotel	42079	not	30750	room	30532	great	18732	n't	18436
staff	14950	good	14791	did	13433	just	12458	stay	11376
no	11360	rooms	10935	nice	10918	stayed	10022	location	9515
service	8549	breakfast	8407	beach	8218	food	8026	like	7677
clean	7658	time	7615	really	7612	night	7596	…	…

Then we use some libraries to perform sentiment analysis. We applied 3 different libraries: Textblob, Vader and Flair. Textblob and Vader presented similar results, while Flair did not obtain results that correlated with the "Rating". With Textblob we got 2 new attributes (Polarity and Subjectivity) and with Vader we got 3 new attributes Positive_Sentiment, Negative_Sentiment and Neutral_Sentiment. Fig. 3 presents the density of the "Polarity" attribute obtained with Textblob. We found that the "Polarity" is mostly positive, which makes sense since, as we saw earlier, most reviews are also positive.

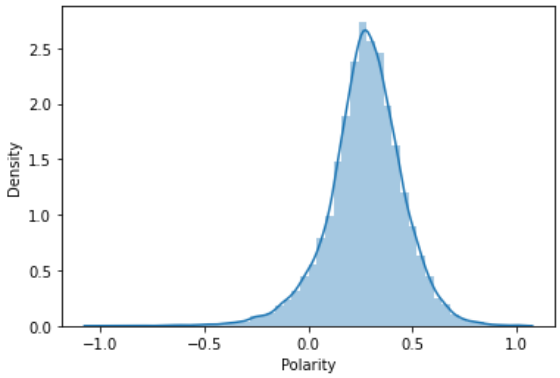

Fig. 3. Density of the "Polarity" attribute obtained with Textblob.

Fig. 4 presents the correlation between "Polarity" and "Rating". We can see that the polarity rises as the rating increases, which clearly demonstrates the existence of correlation. However, we also found that the boxplots of each rating level are superimposed, which is a strong indicator of the difficulty in achieving success in creating classification models. In addition, we verified the existence of many outliers, which may not actually be, as is the case for "Rating" equal to 1, in which we verified the existence of many records with polarity between -1 and -0.65. Fig. 5 presents the correlation between "Subjectivity" and "Rating". As we can see, does not seem to exist any kind of correlation between subjectivity and rating.

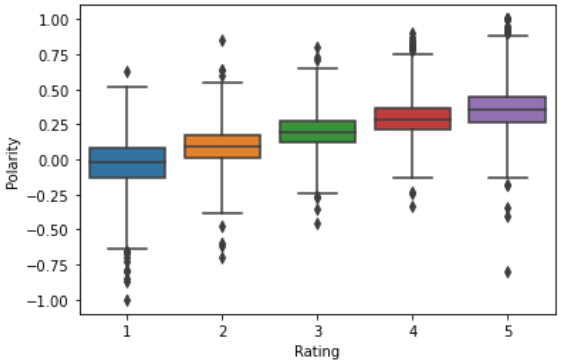

Fig. 4. Correlation between “Polarity” and “Rating”.

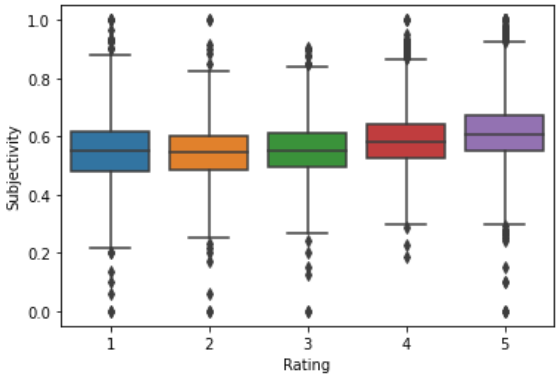

Fig. 5. Correlation between “Subjectivity” and “Rating”.

To create a more simplified version of the assessment made by tourists, we generated a new attribute called “Sentiment” with a value equal to 1 for records where the “Rating” was equal to or greater than 3 and with a value equal to 0 for records where the “Rating” was less than 3. This attribute will allow us to distinguish positive ratings from negative ratings.

We also carried out important preprocessing activities that allowed us to prepare the dataset and discover some important aspects. First, we put all the corpus in lowercase. Then, we tokenize all the corpus and performed the lemmatization and removed all the punctuation. In addition, we used other techniques that did not allow us to obtain better results, such as: removing stopwords, stemming and considering only the characters of the alphabet. Finally, we used the MinMaxScaler to normalize the data.

2.4 Computational techniques

Considering the objective of this work, we believed that it would be important to test the results that would be possible to obtain with different algorithms, both as classification methods and as regression methods. We anticipated that if algorithms as classification methods failed due to previously identified limitations that algorithms as regression methods could be an acceptable alternative in the context of the objective of this work. The algorithms used were: Logistic Regression, Random Forest, Decision Tree, K-nearest neighbors and Bidirectional Long/Short -Term Memory. The first 4 used the Scikit-learn library and the last one used the Keras library.

2.5 Tests and evaluation

Several experiments were carried out with the selected algorithms to tune parameters for optimization. However, no significant differences were found, ending up with the default configuration in the used libraries. To improve the estimated performance of the ML models we performed cross validation with 5 repetitions.

We defined 6 different scenarios to create models. In the first 3 scenarios (#1, #2 and #3) the set of most used words that did not express feeling were removed (hotel, room, staff, did, stay, rooms, stayed, location, service, breakfast, beach, food, night, day, hotel, pool, place, people, area, restaurant, bar, went, water, bathroom, bed, restaurants, trip, desk, make, floor, room, booked, nights, hotels, say, reviews, street, lobby, took, city, think, days, husband, arrived, check and told) and in the other 3 (#4, #5 and 6) all words were kept.

For all scenarios we used the TfidfVectorizer class from the Scikit-learn library to transform the “Review_new” feature to feature vectors and we defined max_features equal to 5000. In addition, in scenarios #1 and #4 the features considered were: “Review_new”, “Polarity”, “Word_Count”, “Char_Count”, “Average_Word_Length”, “Positive_Vader_Sentiment” and “Negative_Vader_Sentiment”; in scenarios #2 and #5 the features considered were: “Review_new” and “Polarity”; and in scenarios #3 and #6 only the feature “Review_new” was considered. Each algorithm was applied to each scenario with both the classification method and the regression method. Finally, all combinations were applied to a 5-class problem (Y = “Rating”) and a 2-class problem (Y = “Sentiment”).

Fig. 6 presents the results obtained with the 5 algorithms for each of the scenarios defined with the classification method for the 5-class problem (Y = “Rating”). As can be seen, the Logistic Regression algorithm obtained the best results for all scenarios, with an accuracy always higher than 0.6, followed by the Random Forest algorithm. The other 3 algorithms obtained considerably lower results, and in the case of the BiLSTM algorithm the results were very weak, as it classified all cases with a “Rating” of 4.

Since scenario 4 was the one that allowed achieving the best results, in terms of accuracy, Table 3 presents Precision and Recall for each of the algorithms in scenario

4 with the classification method for the 5-class problem. We verified that the Logistic Regression and Random Forest algorithms present interesting results. It is possible to verify that relatively high values were obtained for the extreme cases ("Rating" = 1 and "Rating" = 5), but the quality is quite low in the classification of intermediate values.

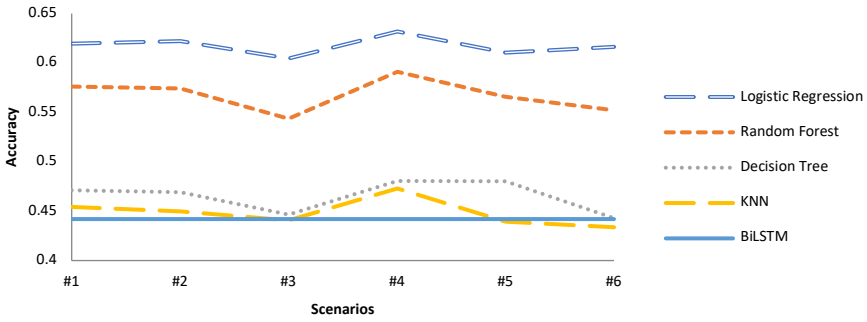

Fig. 6. Algorithms accuracy for the classification method (Y = "Rating").

Table 3. Precision and Recall for scenario 4 with the classification method (Y = "Rating").

	Precision					Recall				
	L 1	L 2	L 3	L 4	L 5	L 1	L 2	L 3	L 4	L 5
Logistic Regression	0,66	0,47	0,46	0,53	0,72	0,65	0,40	0,27	0,52	0,82
Random Forest	0,63	0,48	0,42	0,47	0,64	0,70	0,27	0,04	0,39	0,90
Decision Tree	0,49	0,33	0,23	0,39	0,62	0,50	0,32	0,23	0,39	0,62
KNN	0,37	0,20	0,19	0,40	0,64	0,60	0,22	0,18	0,31	0,67
BiLSTM	0	0	0	0,29	0	0	0	0	1	0

Fig. 7 presents the results obtained with the 5 algorithms for each of the scenarios defined with the classification method for the 2-class problem (Y = "Sentiment"). As can be seen, the results were quite good. Once again, the Logistic Regression and Random Forest algorithms obtained the best results, with the Logistic Regression algorithm showing an accuracy very close to 0.95. Decision Tree and K-Nearest Neighbors algorithms obtained reasonable results mainly in scenarios where more features were considered. The BiLSTM algorithm returned the worst results.

Table 4 presents Precision and Recall for each of the algorithms in scenario 4 with the classification method for the 2-class problem. The results presented by the Logistic Regression algorithm are quite solid. It is verified that the Recall for L 1 (Sentiment = 0) is lower than desirable, but this is probably explained by the dataset being unbalanced.

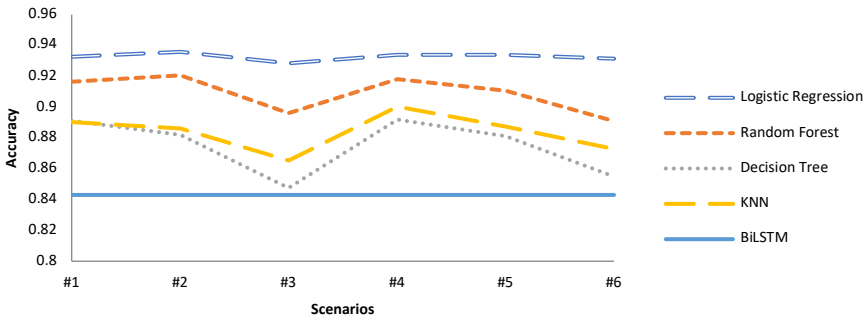

Fig. 7. Algorithms accuracy for the classification method (Y = "Sentiment").

Table 4. Precision and Recall for scenario 4 with the classification method (Y = "Sentiment").

	Precision		Recall	
	L 1	L 2	L 1	L 2
Logistic Regression	0,849624	0,946389	0,702736	0,976846
Random Forest	0,873541	0,922977	0,558458	0,98495
Decision Tree	0,657431	0,934858	0,649254	0,937022
KNN	0,735152	0,923111	0,569652	0,96179671]
BiLSTM	0	0,843061	0	1

The next experiences concern the application of the algorithms to the previously presented scenarios with the regression method. Fig. 8 presents the Mean Absolute Error obtained with the 5 algorithms for each of the scenarios defined with the regression method for the 5-class problem (Y = "Rating"). As it turns out most algorithms got bad results. However, the Random Forest algorithm presented very interesting results, obtaining a Mean Absolute Error of 0.69 in scenario 4 (which is quite good considering the problem in question).

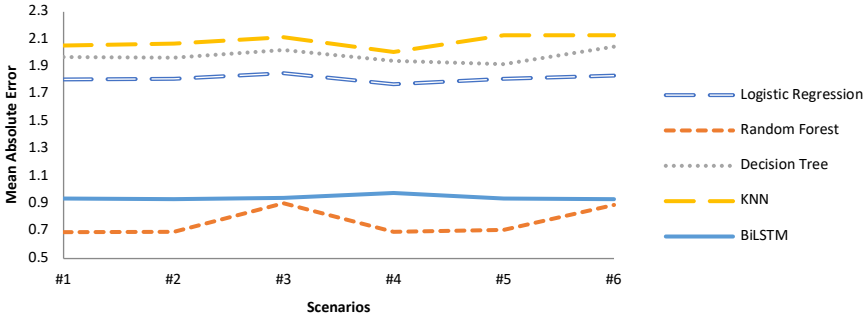

Fig. 8. Algorithms Mean Absolute Error for the regression method (Y = "Rating").

Table 5 presents Mean Squared Error, Root Mean Square Error and Mean Absolute Error for each of the algorithms in scenario 4 with the regression method for the 5-class problem. Once again, it is possible to verify that the Random Forest algorithm obtained very good results, unlike the other algorithms. Although the BiLSTM algorithm seems to give reasonable results, this only happens due to the fact that it always generates the same output and most reviews are positive.

Table 5. Mean Squared Error, Root Mean Square Error and Mean Absolute Error for scenario 4 with the regression method (Y = "Rating").

	Mean Squared Error	Root Mean Square Error	Mean Absolute Error
Logistic Regression	4,140872	2,034913	1,77198
Random Forest	0,733771	0,856604	0,694623
Decision Tree	8,018544	2,831703	1,942417
KNN	5,818965	2,412253	2,007092
BiLSTM	1,522414	1,233862	0,978359

Fig. 9 presents the Mean Absolute Error obtained with the 5 algorithms for each of the scenarios defined with the regression method for the 2-class problem (Y = "Sentiment"). We verified that in this case all algorithms, with the exception of the BiLSTM algorithm, obtained very good results.

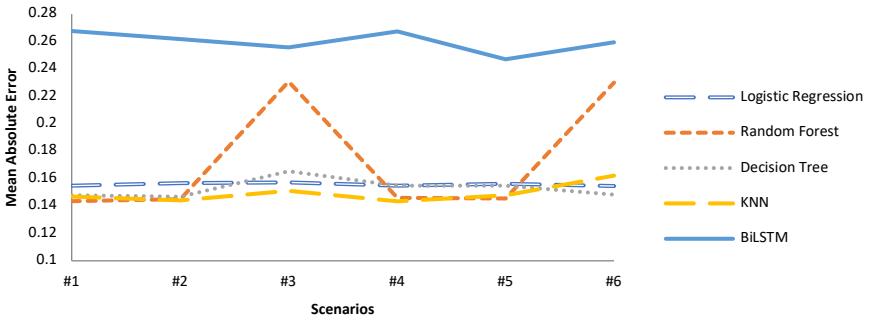

Fig. 9. Algorithms Mean Absolute Error for the regression method (Y = "Sentiment").

Table 6 presents Mean Squared Error, Root Mean Square Error and Mean Absolute Error for each of the algorithms in scenario 4 with the regression method for the 2-class problem. The Logistic Regression algorithm again presents very good results that were consistent across all experiments. In this scenario, the K-nearest neighbors algorithm also presented interesting results.

Table 6. Mean Squared Error, Root Mean Square Error and Mean Absolute Error for scenario 4 with the regression method (Y = "Sentiment").

	Mean Squared Error	Root Mean Square Error	Mean Absolute Error
Logistic Regression	0,097812	0,312749	0,154837
Random Forest	0,07346	0,271034	0,146088
Decision Tree	0,154987	0,393684	0,154987
KNN	0,095474	0,308988	0,143406
BiLSTM	0,132327	0,363768	0,267391

3 Conclusions and Future Work

This work aimed to study strategies to automatically predict tourists' preferences regarding tourism points of interest. The method consisted in using Machine Learning algorithms and Natural Language Processing techniques on reviews that tourists make on TripAdvisor® to predict their assigned ratings. The chosen dataset had a lot of issues making it difficult to get better results (the top 3 were: being unbalanced, having comments that were not about the POI and having comments with very poor writing quality). Since it is a public dataset, we already knew it would be extremely challenging because most existing works present accuracy rates between 30% and 60%. However, we decided to use this dataset as it is a good example of the reality and type of problems that exist in the context of the topic of this work.

The work carried out allowed us to find important conclusions. First, the inclusion of sentiment analysis had a much smaller positive impact than expected. Furthermore, it was possible to notice that, for this dataset, the Vader and TextBlob models obtained a good correlation with the ratings associated with comments while Flair did not. Second, although negative comments are usually longer, the inclusion of the "Word_Count" attribute did not prove to be relevant. Third, the Logistic Regression algorithm proved to be, for classification, the one that achieved a greater accuracy, while the Random Forest algorithm, for regression, proved to be the one that obtained the smallest error. Finally, the Bidirectional LSTM algorithm obtained very poor results for both classification and regression, most likely because the dataset was not large enough. Finally, the conducted study showed that there is a much greater difficulty in predicting intermediate levels, which can have different explanations. If on the one hand, the dataset may not be sufficiently representative, for example in comments with a level 3 rating, on the other hand, the fact that people are different can also have a big impact on a scale from 1 to 5, i.e., the same words have different meanings/weights for different people and people who evaluate a POI with the same rating may express it in a completely different way.

As future work, we intend to replicate this study with a much larger dataset and in which comments/evaluations are about different points of interest. Furthermore, we intend to test with a balanced dataset. Finally, we intend to create a model to identify only those tourists who really like or dislike a particular point of interest, in which the

main objective is not to identify everyone, but fundamentally not to fail those who are identified in those conditions.

Acknowledgments

This work was supported by the GrouPlanner Project under the European Regional Development Fund POCI-01-0145-FEDER-29178 and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within the Projects UIDB/00319/2020 and UIDB/00760/2020.

References

1. Carneiro, J., Martinho, D., Marreiros, G., Jimenez, A., Novais, P.: Dynamic argumentation in UbiGDSS. Knowledge and Information Systems 55, 633-669 (2018)
2. Carneiro, J., Martinho, D., Marreiros, G., Novais, P.: Arguing with behavior influence: a model for web-based group decision support systems. International Journal of Information Technology & Decision Making 18, 517-553 (2019)
3. Carneiro, J., Alves, P., Marreiros, G., Novais, P.: A multi-agent system framework for dialogue games in the group decision-making context. In: World Conference on Information Systems and Technologies, pp. 437-447. Springer, (2019)
4. Thimm, M.: Strategic argumentation in multi-agent systems. KI-Künstliche Intelligenz 28, 159-168 (2014)
5. McBurney, P., Parsons, S.: Dialogue games for agent argumentation. Argumentation in artificial intelligence, pp. 261-280. Springer (2009)
6. Carneiro, J., Andrade, R., Alves, P., Conceição, L., Novais, P., Marreiros, G.: A consensus-based group decision support system using a multi-agent MicroServices approach. In: Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, pp. 2098-2100. (2020)
7. Carneiro, J., Alves, P., Marreiros, G., Novais, P.: Group decision support systems for current times: Overcoming the challenges of dispersed group decision-making. Neurocomputing 423, 735-746 (2021)
8. Carneiro, J., Saraiva, P., Conceição, L., Santos, R., Marreiros, G., Novais, P.: Predicting satisfaction: perceived decision quality by decision-makers in web-based group decision support systems. Neurocomputing 338, 399-417 (2019)
9. Sun, S., Luo, C., Chen, J.: A review of natural language processing techniques for opinion mining systems. Information fusion 36, 10-25 (2017)
10. Chen, X., Xie, H., Cheng, G., Poon, L.K., Leng, M., Wang, F.L.: Trends and features of the applications of natural language processing techniques for clinical trials text analysis. Applied Sciences 10, 2157 (2020)
11. Thanaki, J.: Python natural language processing. Packt Publishing Ltd (2017)