ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

Master Thesis Data Science and Marketing Analytics

Local Interpretable Model-agnostic Explanations for Long Short-Term Memory Network used for Classification of Amazon Customer Reviews

Theodora Kousta

Student ID Number: 546601

Supervisor: Luuk van Maasakkers

Second Assessor: Clement S. Bellet

26/07/2020

Abstract

The analysis of customer reviews is a matter of exhaustive research lately since reviews often convey insights decisive to the decision-making process of a business. In their attempt to perform sentiment analysis on reviews in the most effective way, many analysts have employed the usage of Long Short-Term Memory (LSTM) networks because of their competence in analyzing sequence input data (e.g. text). However, a neural network remains a black-box model, meaning that the patterns behind the algorithm's operations are unknown. Therefore, another issue which draws the attention of many researchers is the interpretation of black-box models, under the scope of fidelity. Trusting a model implies that we are able to comprehend why and how a certain prediction was made, and it is of supreme importance both to the scientific and the marketing world. Local Interpretable Model-agnostic Explanations (LIME) is a local interpretation technique which assists us to zoom in into every single instance (e.g. a customer review) in order to provide simple explanations on how the black-box model made a specific prediction. The usage of LIME has not thoroughly been studied on LSTM networks for text classification problems, and this is the main issue being examined in this paper. More precisely, we apply LIME in order to locally interpret an LSTM network used to classify a set of thousands of Amazon's customer reviews into positive or negative. Afterwards, we present the "explanations" produced by LIME, which consist of a set of the most important words of each review text, which determined each specific classification decision. We show that this information is crucial to an analyst in a way that if words with insignificant semantic value are considered to be important by LIME, then there is probably something wrong with our network - which has possibly captured the wrong patterns – and proper modifications are needed. We also indicate the importance of the interpretation results, by means of feedback, to a marketeer. For example, when aspects of a product are assigned with negative weights by LIME, then, this is an indication of the dissatisfaction of the consumers regarding these aspects. Thus, the business needs to take the apt decisions.

Keywords: Classification of customer reviews, Text classification, Sentiment analysis, Recurrent Neural Networks, RNN, Long Short-Term Memory, LSTM, Local Interpretable Model-agnostic Explanations, LIME, Submodular pick.

Table of Contents

1. Introduction 4
2. Literature 8
2.1. Theoretical Background 8
2.2. Related Research 13
3. Methodology 16
3.1. Recurrent Neural Networks 16
3.2. The Vanishing Gradient Problem 18
3.3. Long Short-Term Memory 19
4. Local Interpretation of a Black-box Model 23
4.1. LIME 24
4.2. Global Interpretation with Submodular Pick 28
5. Analysis 31
5.1. The Data 32
5.2. Application of the Methodology on Amazon Customer Reviews 32
5.2.1. Data Pre-processing 32
5.2.2. The LSTM Network 34
5.2.3. Local Interpretation with LIME 35
5.2.4. Global Explanations 35
6. Results 35
6.1. The LSTM Network 35
6.2. Local Interpretation with LIME 36
6.3. Global Explanations 40
7. Closing Remarks 43
7.1. Marketing Implications 44
7.2. Limitations of the Model 46
7.3. The Future Ahead 48
8. References 49
9. Appendix 52

1. Introduction

The rapid development of the web and the immense increase of internet users, which have been taking place during the last couple of decades, have caused internet to appear as a really auspicious market which attracts numerous businesses of varying sizes and prospects, from niche to multinational level. Nowadays, almost every business, apart from its physical store, also manages its own online page from which it conducts electronic retail or wholesale transactions. Increasingly prevalent is the case where a business only has an electronic form, meaning that they do not own a physical store or offices. This kind of online retailers provide their services or products online, assisting innumerable amounts of customers, usually on a global scale. Amazon is the greatest online retailer, in terms of market share and revenue, in this category (Reno, 2014), and it is regarded to be one of the "Big-Four tech companies" worldwide (Ritholtz, 2017), (Rivas, 2017). Having a limited number of tangible stores or offices, some of the most common ways, for these kind of retailers, like Amazon, to get in contact with their customers is through forums and the reviews section of their online store, where every user is free to communicate their experience with the business' product or service. These reviews constitute essential source of information for the retailer, since, they provide crucial insights regarding the consumers' satisfaction with their products, and thus, indications for further quality improvement.

In the contemporary "one-to-one marketing era", in which the target-market size has shrunk to an individual level, by means of adapting the marketing approach to each distinct consumer, these insights can be of supreme importance to the marketing department of the business as well. More specifically, this information can assist the site's marketeers to compose the appropriate marketing mix in order to target each individual user according to their specific needs. Consequently, a proper way of analyzing these reviews in order to deduct the required information out of them and, ultimately, utilize them accordingly, is of prime importance to Amazon's (and not only) administrators.

In this paper we will focus on assisting this purpose by applying a classification technique on Amazon's customer reviews. More specifically, our work intends to classify the customer reviews to positive and negative ones, and define which words in the customer reviews (aspects of the products or sentiments/feelings of the consumers) are accountable for determining whether a review is to be considered positive or negative. For this purpose, an adaptation of deep Recurrent Neural Networks (RNN), a so-called Long Short-term Memory network

(LSTM), is trained and tested on thousands of reviews for Amazon products, in order to perform sentiment analysis on them.

The decision for applying artificial neural networks emerges from the fact that they provide quite accurate predictions, relatively to other machine learning techniques. Compared to other types of artificial neural networks, the ability of LSTMs to store information of the data which the network has already processed, and, afterwards, use this information to accurately forecast an outcome, makes them the most suitable network when it comes to making predictions based on sequence input data, such as text. Some significant examples of the LSTM applications are the time-series forecasting, such as the stock-market prices prediction, and the natural language processing, such as voice recognition.

An LSTM network, just like every single type of artificial neural networks, is a so-called "black-box" model, meaning that the structure of the patterns behind the algorithm is unknown or too vague for an analyst – or any other human that just applies the algorithm without carrying any knowledge on how it was developed – to understand. In other words, no apposite reasoning can be provided on how the network made a specific prediction or produced a particular result. For instance, in the case of text processing by an LSTM network, and more specifically, in the case of classifying customer reviews into positive or negative ones, justifying why the algorithm assorted a specific review as positive can be intriguing. Was it the existence of a specific word, or group of words, that triggered the algorithm to conclude at this decision? Does this mean that certain words (e.g. aspects of the product or certain feelings of the customers) are more likely to be associated with positive reviews? Someone would expect that this is the case, and, indeed, this should be. The network, throughout its training process, should "learn" and "remember" which groups of words are commonly related to positive or negative reviews, and afterwards, base its predictions on this memory, but in a way that "makes sense". More specifically, if, for instance, the majority of negative Amazon reviews, by coincidence, also refer to the "Amazon" store or/and criticize it, the algorithm should not conclude that the occurrence of the word "Amazon" is instantly related to a negative review. This simply does not make sense and it was just coincidental in this specific set of reviews.

A network like this could demonstrate a high predictive accuracy on this specific dataset, which, as already mentioned, is falsely based on insignificant data points. But what if this highly accurate network is afterwards tested on a dataset of another web-shop's reviews, where the word "Amazon" is absent? Will the algorithm perform as good in predicting the negative

reviews here? Probably not, and that is the reason why an analyst should be able to comprehend the manner in which an algorithm operates in order to trust it. High accuracy does not always abide by trustworthy results. However, an analyst or a marketeer should be able to trust the program and the predictions it produces in order for them to take proper decisions and apply the apt strategy. Therefore, several techniques have been developed in order to assist analysts to see what exactly goes on behind a black-box model's algorithm.

When it comes to review data, each different customer share their experience on different products, and therefore, totally different words (explanatory variables) are responsible each time for a positive or negative rating of a product. Therefore, it is reasonable to think that a "global" interpretation technique would miss a lot of valuable information when it comes to individual customers' needs and personalized targeting. In other words, although a global interpretation technique might, for instance, reveal that the majority of positive reviews were about "Alexa", that does not imply that Amazon should recommend Alexa to every single user. Although only a minority of consumers might have left a quite negative review about Alexa, this information is neglected and further promotion of Alexa should not target these specific customers. Therefore, a "local" interpretation technique would be more suitable in cases such as the one mentioned above, and in general, in problems where the number of explanatory variables is large, or when personalized targeting is the aim of the business. A technique called "Local Interpretable Model-agnostic Explanations (LIME)" is one of these "interpretation techniques" and it serves us to zoom in on every single instance (local interpretation) in order to understand the model's predictions thoroughly. It can be used to plainly explain the predictions of any classifier or regressor. In our case, it can indicate which specific words triggered the LSTM algorithm to classify a specific customer review as positive or negative. This research utilizes the LIME algorithm in order to provide simple explanations to some specific predictions the LSTM network produced.

The aim of this paper is to examine and assess the contribution of the LIME algorithm on an LSTM network into two sections: the scientific and the marketing one, since it has not been extensively researched and examined yet.

Regarding the scientific objective, this research attempts to expand the application of LIME on recurrent neural networks in order to assist us to get a grasp of how this black-box model works in forming some specific predictions, especially when it comes to text classification problems. For instance, LIME could indicate to us which specific words of a consumer's review triggered

the LSTM model to classify it as a negative one. This type of information could be remarkably valuable to an analyst, since it is an indication of the validity of the algorithm. Having a highly accurate model is not enough when it comes to further decision making; the way the algorithm works is crucial information for trusting the model. Local interpretation techniques, such as LIME, are considered the most suitable ones (compared to global ones) when it comes to models that utilize many features as explanatory variables (e.g. numerous different words from customer reviews data).

When it comes to the marketing contribution, the proposed work aims to explain why (local) interpretation techniques would be an indispensable tool when it comes to marketing or managerial decision making. First and foremost, just in the same way that it applies to an analyst, being able to trust a model and its predictions is crucial for a business' operations and decision making. Decisions come after the observation of a model's results, but in order for these decisions to be optimal, the validity of the model's results need to be undoubtful. Taking the wrong decisions could lead to loss-making results for a business. Secondly, local interpretation techniques, such as LIME, assist us to zoom in into every single instance (e.g. every customer review), and thus, observe each single customer's needs. Therefore, local interpretation techniques could be an essential tool that enhances personalized marketing strategies. More specifically, a marketeer could be able to analyze every single customer independently and personalize the marketing mix accordingly by promoting the right products to the right customers. Interpretation techniques can assist the business perceive their customers' reviews as feedback in order to assess their performance, as well as to aid them grip a thorough understanding of their customers' specific needs. For instance, they can indicate which features of a product each customer is dissatisfied with, so that the business can provide the appropriate support, solutions or alternatives. These features of the product are also the ones which need improvement. The same applies for the features of the product that are associated with positive feedback. Namely, it would be beneficial for the company to exploit these features or/and accentuate them in its upcoming marketing campaign.

To sum up, the purpose of this research, is the attempt to examine the usage of LIME in a text classification problem, analyzed by a black-box LSTM network, and explain the predictions made. This analysis aims to be carried out in a way that serves both the scientific and the marketing contribution described above, and thus in a way that serves the stakeholders' need to get the most out of the underlying information concealed in the data. In Section 3 and 4, a thorough description of the RNN's and LSTM's architecture, as well as an exhaustive

illustration of how LIME produces its explanations to any model, are provided, while, in Section 5 and 6, the application of them, specifically on an Amazon customer reviews' dataset is described and the experimental results are discussed. Finally, Section 7 concludes the proposed analysis.

2. Literature

2.1. Theoretical Background

Target-market size has been shrinking, during the last years, since we have been shifting from a "mass marketing era" to a "niche marketing" and, eventually, to a "one-to-one marketing era". This is mainly driven by the customers' heterogeneity, and it is primarily a result of a group of factors, such as the rise of media and the multiple communication media, as well as the breakthroughs in printing and manufacturing (Palmatier & Sridhar, 2017). Customer heterogeneity implies that each distinct customer has special needs which they seek to satisfy, and these differences usually emerge from customer's characteristics, such as life experiences, functional needs or self-image (Palmatier & Sridhar, 2017). However, they can also be driven by marketing activities. Retailers attempt to achieve a personalized targeting so that all separate consumers' special needs are satisfied. As Palmatier and Sridhar (2017) mention, this can be achieved by combining one or more features of the firm's marketing strategy to each distinct customer, by eventually targeting them with individual recommendations ("behavioral targeting"). This sounds more like an ideal situation, since identifying customers' true needs is exceedingly challenging. The increasing demand of online retailers for behavioral targeting, accompanied by their growing desire for the most efficient marketing budget allocation, have raised the need for a machine learning technique that produces the most accurate predictions as possible.

In the contemporary era, where social media dominate the internet world, everyone is free to express their opinion on online forums or blogs, by means of leaving reviews about products or services they purchased and had an experience with (Khan, Baharudin, & Khan, 2011). According to Gräbner et al. (2012), it is a fact that quite a large part of potential buyers is scrolling through the reviews section of an online store' page before reaching a decision for finalizing a specific transaction. More specifically, the majority of potential customers tend to base their pre-purchase research on information gathering which is mainly focused on comments and feedback that people, who already had a post-purchase

experience with a specific product/service, have shared online (Gräbner, Zanker, Fliedl, & Fuchs, 2012). From the retailer's perspective, customer reviews provide essential feedback and information about the specific customers' needs which they attempt to gratify, and, thus, they appear to constitute an essential information source for marketeers (Bagheri, Saraee, & Jong, 2013). Consequently, as Khan et all. (2011) state, *"The reviews about any entity … are useful in decision making for both the customer and manufacturer"*. The rapidly increasing volume of customer reviews, during the last decade, makes it arduous for businesses to interpret all the insights provided by each distinct review, and therefore, a method that aids retailers to summarize all this information is of supreme necessity (Bagheri, Saraee, & Jong, 2013). Preferably, a classification technique based on sentiment analysis, in order to categorize each different review as positive or negative, is needed (Khan, Baharudin, & Khan, 2011).

According to Bagheri et al. (2013), sentiment analysis (a form of text analysis) appeals a lot to "natural language processing" (NLP) problems, such as the examination of consumers' reviews. The most crucial problems, with which sentiment analysis deals, are: identifying aspects or/and opinions/sentiment that certain words imply, where with "aspect" we refer to a specific characteristic of a product, on which a customer's review is based (Bagheri, Saraee, & Jong, 2013). The "sentiment" corresponds to the feeling (positive or negative) a customer has towards a specific aspect of a product or service (Kiritchenko, Zhu, Cherry, & Mohammad, 2014). Therefore, a customer's sentiment is always related to an aspect. For instance, when it comes to the following hotel review: "The room was excellent, but the staff was quite unfriendly", we can conclude that the customer demonstrates a positive feeling concerning the room, but when it comes to the staff, that feeling is negative. According to Gräbner et al. (2012), sentiment analysis assists retailers to quantify the customers' reviews as if the customers were to rate the product with number of stars. Notably, with sentiment analysis, a business seeks to pinpoint which factors or aspects of its products are responsible for offering the customer a good or bad post-purchase experience (Gräbner, Zanker, Fliedl, & Fuchs, 2012). In order to do so, they attempt to identify specific words in the review text which are related to a good or bad rating of the product. In this paper, we will perform sentiment analysis on a set of thousands of customer reviews on Amazon's products by utilizing a recurrent-like architecture of neural networks, a so called "Long Short-Term Memory" network, in order to, afterwards, classify the reviews into positive and negative ones. The network, throughout its training process, will

"learn" and "memorize" which specific words are related to a positive or negative sentiment, and will form its predictions accordingly.

The applications of artificial neural networks have expanded during the last years with deep learning techniques demonstrating an increased popularity from 2014 and afterwards, as it is indicated by Google Trends data (Rai, 2019). From, initially, being used for tasks such as image or voice classification, and originating from mathematical neurobiology (A. Vellido, 1999), neural networks now appeal a lot to business and marketing purposes, mainly because of their accurate forecasting ability (Zhang P. G., 2004). According to Zhang (2004), the capability of predicting future outcomes as precise as possible is of prime importance when it comes to numerous business fields, marketing strategy, purchasing or supply decisions. During the years, forecasting was mainly carried out by linear models, because of their plainness in use and comprehensibility (Zhang P. G., 2004). However, as Zhang P. G. (2004) mentions, these methods suffer from some significant limitations, with one of the most crucial one being their inability to observe and explain non-linearity between the features of the model and the independent variable. In the real-world, the non-linearity of the data relationships is almost always the case, and thus, stakeholders, should not focus on the results produced by a model that ignores it. By being able to seize these complex real-world relationships, because of their structure, artificial neural networks are regarded as auspicious machine learning techniques and, thus, yield fairly accurate results (Zhang P. G., 2004).

Another powerful characteristic of neural networks lies in the fact that they are a non-parametric technique, meaning that no significant restrictions are required (e.g. for the distribution of the population) for determining and controlling the data generation process (Yochanan, 2002). Besides, according to Zhang (2004), a neural network is a data-driven technique, and its learning ability relies on the data themselves. Furthermore, by being able to handle and operate on an enormous volume of data, deep learning and artificial neural networks appeal a lot to the contemporary big-data era (Kasun, Zhou, Huang, & Vong, 2013), (Zhang, Yang, Chen, & Li, 2018). Finally, regarding the learning process of an algorithm, the traditional statistical approaches allow the creator of the algorithm to insert "knowledge" to the system by designing the algorithm's architecture accordingly (Schuster & Paliwal, 1997). However, as Schuster and Paliwal (1997) mention, when it comes to neural networks, the system can deduce "knowledge" from the data themselves.

Various types of neural networks have been developed, each one competent to deal with different types of problems. The differences in the architecture of the different network types allow each one of them to be capable of operating on different input data (e.g. image, text, sequence input) and, according to their output results, to be characterized as supervised (e.g. Recurrent Neural Networks, Convolutional Neural Networks) or unsupervised (e.g. Autoencoders) learning techniques (Allaire, 2018). For instance, Convolutional Neural Networks (CNN) are the most suitable type of neural networks for image classification or recognition problems (Sharma, Jain, & Mishra, 2018), while, Recurrent Neural Networks (RNN) work best for sequence input data, such as text or speech input (Lipton, Berkowitz, & Elkan, 2015). Generative Adversarial Networks (GAN) are capable of creating new data (e.g. new images) and distinguishing whether this output looks natural or factitious (Rai, 2019). What is more, Autoencoders, an essential unsupervised (or self-supervised) deep learning technique, are often used for reducing the dimensionality and diminishing the noise of the data (Allaire, 2018), (Rai, 2019). When it comes to marketing, unsupervised learning problems include market segmentation (i.e. clustering of the consumers according to their similarities), and artificial neural networks are efficient enough to deal with such problems (Shah, 2017). However, supervised learning (e.g. forecasting) is currently the most often required deep learning technique in the business and marketing field (Allaire, 2018).

Different problems require different solutions, and, thus, to identify the most appropriate algorithm to operate on them. Regression, classification and time-series problems, of engineering or business importance, allow a wide range of algorithms to operate on them and provide solutions (Schuster & Paliwal, 1997). Each unique machine learning technique has its own limitations and the most appropriate one should be selected according to the specific problem, the input data and the desired solution (Schuster & Paliwal, 1997). A really common problem category is the one in which data are in a temporal sequence form. Such kinds of problems might be: time-series forecasting (e.g. stock prices predictions), dynamic control systems, or natural language processing such as speech recognition and text classification (Schuster & Paliwal, 1997). As Schuster and Paliwal (1997) mention, in all these cases input data are in a sequential form, meaning that each data point provides information for the next in sequence data point. In other words, “sequence matters”.

Recurrent Neural Networks (RNN) provide an optimal solution when it comes to this type of problems, since they incorporate correlations between data points which are close in

temporal sequence (Graves, Mohamed, & Hinton, 2013). According to Allaire (2018), the reason why recurrent neural networks are so competent with input data for which sequence has an essential meaning (e.g. customer reviews), is because, compared to other neural networks, they have "memory". More specifically, with a feed-forward network, in order for a sequence of data to be processed, the network would consider each separate word input as independent and would process it ignoring the connection between the previous and the next word input (Allaire, 2018). Therefore, according to Allaire (2018), in order for such connections to be considered by the network, each piece of sequence input (e.g. a whole customer review) should be transformed into a single data input and be read by the network at once. On the other hand, an RNN keeps each input's information as knowledge for the following inputs. Therefore, considering, for instance, a text input, each single word can be processed independently with the information of the words already processed not being lost, but preserved and constantly being updated in order to, eventually, provide a relevant meaning of the entire text input (Allaire, 2018). The same applies for time series data, in which the previous state of a variable (e.g. temperature, stock prices) is important for predicting the next outcome. This powerful self-feedback attribute of RNNs is what makes them the dominant type of neural networks for "predicting what comes next" and, thus, being preferred in forecasting problems (Allaire, 2018).

Recurrent neural networks, just like every type of artificial networks, are black-box models, meaning that the structure of the patterns behind the algorithm, as well as the manner in which the results are produced, is vague. As a result, interpretation of the model's predictions is insufficient. Data analysts, who are not aware of how these machine learning algorithms have been developed, but just apply the models, cannot utterly trust them if they are unaware of the relationship between the input data and the eventual prediction (Tamagnini, Krause, Dasgupta, & Bertini, 2017). According to Tamagnini et al. (2017), evaluating just the predictive accuracy of a model alone is not enough in many science fields where thorough understanding of the prediction process is required in order for the model to be faithful. They also insist that the incompetence of many machine learning techniques to provide explanations of their predictions impedes their efficiency, and, in order to achieve more interpretable results, by shifting to a more simplified and interpretable model, a drop in predictive accuracy usually occurs (trade-off between model accuracy and model interpretability). Therefore, interpretation of black-box models can provide crucial insights for the prediction process, as well as for whether we should trust a

specific model or not, by unveiling what elements drive its results. In this way, the proper decisions would be taken by humans (e.g. managers, analysts or marketeers) and the apt strategy would be applied.

Figure 1 demonstrates how the interpretation of a black-box model's results can assist the decision-making process: The black-box model (e.g. a neural network) makes predictions given a specific dataset. Considering a classification problem, green predictions represent Class 1, while pink predictions represent Class 2. The interpretation of these predictions, as well as the way in which they were produced, are unclear to a human being. The "Pick step" refers to the specific instances on which a local interpretation technique will be applied in order to select the features which contributed to each different classification decision. The resulting features selected and demonstrated, for each different prediction, are the "Explanations" produced by the local interpretation algorithm. These explanations are now understandable enough by any human being, and thus, knowledge on how the black-box model produces its predictions is obtained. Now that humans can trust the model, they are able to make decisions according to the model's predictions.

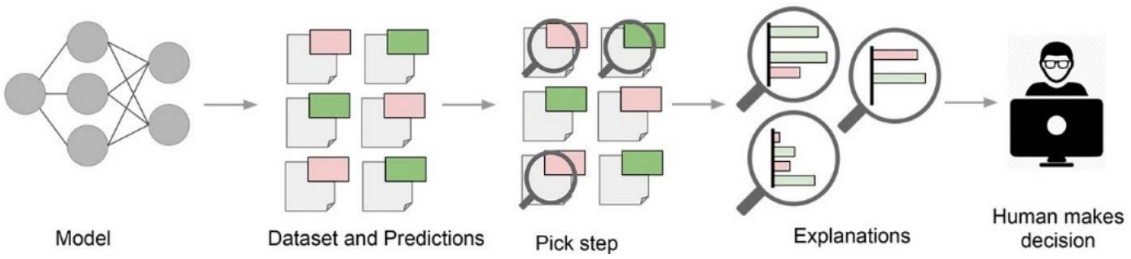

Figure 1: Black-box model local interpretation for decision making. With green and pink color are demonstrated the two different classes to which the black-box model classifier assorts each input during the prediction making process. In the "Pick step" we select a set of predictions to interpret, for which "Explanations" are produced. These explanations tell the human how the black-box model arrived at each different prediction, in a simple and comprehensible manner. Now that the human is able to understand and trust the model, decisions can be taken *(Ribeiro, Singh, & Guestrin, 2016)*.

2.2. Related Research

A lot of interest has been expressed towards sentiment analysis on customer reviews by multiple researchers. With the expansion of internet usage, a notable amount of individuals exhibit a high preference on completing transactions online, because of the convenience that internet renders. Such actions might include online orders, ticket and hotel reservations, or internet banking (Khan, Baharudin, & Khan, 2011). With that being the case, people's

comments and reviews are the prime source of information to these businesses, regarding their customers' opinion. Therefore, much research has been conducted in order to identify the most appropriate method to extract the most valuable information out of these customer reviews. This research, so far, is focused on sentiment analysis techniques which use an established, in advance, list of seed words, in order to withdraw aspects and opinion-indicating words, which assist the researcher to classify the consumers' reviews into negative or positive. Examples of this work are the papers of Wei et al. (2009), Khan et al. (2011), Gräbner et al. (2012), Broß (2013), Bagheri et al. (2013), Kiritchenko et al. (2014), and more.

The application of Recurrent Neural Networks (RNN) has started to expand in the business, marketing and finance fields lately, since they have exhibited significant contribution, "especially in machine learning problems in which the input is of variable length" (Chung, Gulcehre, Cho, & Bengio, 2014). This increasing preference in favor of RNN is chiefly triggered by the fact that they are efficient in producing accurate predictions based on the information they accumulate during the data processing procedure. When it comes to opinion-mining, Liu et al. (2015), Chaudhuri & Ghosh (2016), and Agarap & Grafilon (2018) have published notable works of RNN applications on customer reviews. More specifically, Liu et al. (2015), by applying word embeddings[1] in combination with RNNs on product reviews, evinced that RNNs can outperform other types of deep models, even when they are not affixed with extra features apart from the word embedding values. Chaudhuri and Ghosh (2016), compared the performance of multiple RNN models in order to predict sentiment features of customer reviews, on a review input level. Their study pinpointed that, among Long Short-Term Memory (LSTM) RNNs, Bidirectional LSTM, and Hierarchical Bidirectional Recurrent Neural Networks (HBRNN), the latter yield, indisputably, the most accurate results, in terms of F1 score, recall and precision[2]. Lastly, Agarap & Grafilon (2018), validated that bidirectional LSTMs are able to perform

[1] **Word embedding** techniques are used for conceiving natural language understanding from a text in order to perform sentiment analysis, in a deep learning discipline. Words are represented by vectors of numbers - where this continuous representation is similar for words with related meaning – and the features of these vectors are learned in a similar to a shallow neural network way (Brownlee, 2017).

[2] **Precision** and **recall** are metrics used to measure the performance of a model, with precision indicating the percentage of the results that are relevant, and recall measuring the percentage of the relevant results which were correctly predicted by the model (Koehrsen, 2018). **F1 score** is also a measure of performance and is defined as the "harmonic mean" of recall and precision (it takes values between 0 and 9) (Koehrsen, 2018).

satisfyingly good when it comes to sentiment, as well as recommendation, classification of reviews.

When operating with a black-box model, like a neural network is, it is assured that the model is expected to achieve higher accuracy in its predictions, compared to alternative methods, but at the cost of low interpretability. In other words, the algorithm will produce highly accurate results, but the manner in which these predictions are generated is quite obscure (Olden & Jackson, 2002). The query that arises at this point questions the fact whether accuracy is enough to trust a model. The answer would be "no", and that is because these highly accurate predictions might be based on absurd features which just happen to co-exist in each of all the alike predicted or classified cases. In other words, the predictions could be correct, but for the wrong reason. Tamagnini et al. (2017), Bastani et al. (2019), as well as other researchers, attempted to tackle this issue by applying instance-level visual explanations and model extraction techniques on black-box models, respectively. Instance-level explanations indicate to the user which features of the model are responsible for a specific prediction (Tamagnini, Krause, Dasgupta, & Bertini, 2017), while model extraction is based on a decision tree-like interpretation of the black-box model (Bastani, Kim, & Bastani, 2019).

Although Ribeiro et al. (2016) did a profound job with introducing "Local Interpretable Model-agnostic Explanations (LIME)" and its remarkable contribution in dissolving the uncertainty described above, RNN and LSTM networks, for text classification problems, have not yet been an object of thorough study under the scope of LIME. However, the contribution of LIME has been analyzed on convolutional neural networks (CNN) for acoustic data (Mishra, Sturm, & Dixon, 2017), on Computer-Aided Diagnosis systems (CAD) to address the importance of stability in medical analysis (Zafar & Khan, 2019), on Bayesian predictive CNN models (Peltola, 2018), on image CNN networks for tumors detection (Sousa, Vellasco, & Silva, 2019), and more. Also, an "Autoencoder-based" adaptation of LIME, to tackle the issues of instability (because of the randomly generated data process) and the local, only, faithfulness of the explanations, has been proposed (Shankaranarayana & Runje, 2019). Not surprising is the fact that the applications of LIME on the medical field are a lot, since validity and fidelity of the model's results are of crucial importance in the field.

The explanations that LIME produces are based on qualitative features (e.g. words in an examined part of text) which denote the reason why the model arrives at a specific prediction each time (Ribeiro, Singh, & Guestrin, 2016). According to Ribeiro et al. (2016), when it comes to text classification problems, these interpretable explanations could take a binary form (1 or 0), indicating whether a specific word exists in the text or not. This is an explanation quite intelligible to a human being, compared to the complex techniques the classifier might use (e.g. word embeddings) (Ribeiro, Singh, & Guestrin, 2016). However, the explanations that LIME produces, are "local", meaning that they differ for each different prediction, and, therefore, cannot be "globally" reliable. All in all, LIME aims to make complicated machine learning techniques comprehensible and, thus, trustworthy and more efficiently used by humans. In this paper, the expansion of LIME algorithm, in locally interpreting an LSTM model for text classification, will be the main aspect scrutinized.

3. Methodology

In this section of the paper, the methods applied for this research are exhaustively described. Compendious insights on how these algorithms operate are provided.

3.1. Recurrent Neural Networks

Recurrent neural networks (RNNs) are a specific category of artificial neural networks, most suitable for sequence input data. An RNN can be plainly considered as the incorporation of extra loops to a deep[3] feed-forward network (Brownlee, 2016). Namely, according to Brownlee (2016), in a certain layer, the neurons, apart from the forward signal of information they send to the next layer, they can also convey a signal sideways. In addition, the output that the RNN produces, is fed as input to the network, accompanied by the next sequence input. In this way, the previous output serves as feedback to the network. The recurrent links of RNNs assist them to preserve "memory" and, thus, become competent to cope with more conceptual problems. The memory of the RNN algorithm (i.e. the previously calculated outputs) is preserved in the so-called "state matrices", which are used, accompanied with the new input each time, to determine the subsequent output (Sethi, 2019). Although it can be claimed that simple feed-forward neural networks and MLPs preserve some memory as well, the case is different with RNNs (Venkatachalam, 2019).

[3] "Deep" refers to the existence of more than one hidden layer in an artificial neural network.

According to Venkatachalam (2019), a feed-forward network is able to "remember" information, but only while it is being trained on a certain dataset. On the other hand, a recurrent network is able to preserve memory of previously trained data and apply it on future problems. In order to do so, RNNs have a recurrent hidden state (h_t) which is activated according to its previous state of activation information (Chung, Gulcehre, Cho, & Bengio, 2014). This hidden state is responsible for operating on the input sequence data each time.

When operating on an RNN input, three parameters need to be specified: the batch size, which is the number of randomly selected input data to be processed at the same time, the number of steps for each batch, denoting the number of time steps or segments of data to be processed for each different batch input, and the number of features to be included in each different step (Helmini, 2019). As Helmini (2019) mentions, an RNN layer can be perceived as a single recurrent cell that is reeled-out according to the "number of steps" parameter that is pre-specified. When processing from one step to another, each previous step's hidden state provides information, which, combined with the current input, determines the specific step's hidden state (Helmini, 2019). It is important to pinpoint that when operating on input data of different length (e.g. reviews data) the length of each different input sequence cannot be pre-specified for an RNN. Consequently, each different connection inside the network cannot have a unique weight and bias parameter. Therefore, RNNs operate under the "parameter sharing" principle, meaning that the hidden state weights, the input weights and the output weights are identical for each different step (Venkatachalam, 2019), (Helmini, 2019).

Formulating an RNN in a more mathematical approach can be achieved as follows: Let us consider an input data sequence $x = (x_1, x_2, \dots, x_T)$, (e.g. a review text), where T is the number of timesteps, and x_i, $(i = 1,2, \dots, T)$, the different elements (e.g. words) of the input sequence. A recurrent neural network updates its hidden state $h_t = (h_1, h_2, \dots, h_T)$ and its output $y_t = (y_1, y_2, \dots, y_T)$, each time, by:

$$h_t = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (1)$$

$$y_t = W_{hy}y_t + b_y \quad (2)$$

respectively (Graves, Mohamed, & Hinton, 2013). In the equations above, W are weight matrices, b are biases vectors, and H is an activation function; usually a sigmoid or a

tangent function (Chung, Gulcehre, Cho, & Bengio, 2014). More precisely, the W_{xh} matrix contains the "input-hidden weights", and the W_{hh} matrix the weights of the "previous hidden state" (Graves, Mohamed, & Hinton, 2013).

3.2. The Vanishing Gradient Problem

As far as RNNs are concerned, training the network with backpropagation, in order to update the weight matrices W_{xh} and W_{hh} in equations (1) and (2), turns out to be extremely strenuous (Chung, Gulcehre, Cho, & Bengio, 2014). That is because of the circular type of the recurrent connections which impede the proper adjustment of the network's weights by propagating the errors backwards (Brownlee, 2016). According to Brownlee (2016), a solution to this problem is to train the model with an altered backpropagation procedure, in which the recurrent loops of the RNN network structure are "unrolled". In order for this to happen, the neurons which belong to recurrent links are replicated. Eventually, the network acquires a feed-forward structure, and thus, backpropagation can be practiced on it. This technique is called "backpropagation through time (BPTT)" (2016).

However, as Brownlee (2016) mentions, when the network trained by backpropagation is very deep, or if it is an unrolled RNN, then the calculated gradients, used for adjusting the weights and biases, can either grow too large ("explode") or crucially shrink ("vanish"). The latter is a notable issue called "the vanishing gradient problem", and both cases mentioned above cause the gradients to be quite erratic; consequently, the network becomes unstable and untrustworthy (2016). This issue causes the effect of the long-term dependencies to be ignored, as it gets masked by the short-term ones (Chung, Gulcehre, Cho, & Bengio, 2014). The "short-term memory" can be considered as the ability of any recurrent network to preserve the previous input's information through its feedback loops (Hochreiter & Schmidhuber, 1997). In order for the network to account for the long-term dependences, a couple of approaches have been proposed throughout the years, with one of the most prevalent ones being the addition of an extra layer (or "recurrent unit") to the network, which can be perceived as a more advanced activation function (Chung, Gulcehre, Cho, & Bengio, 2014). This adaptation of the common RNN is referred to as the "Long Short-Term Memory (LSTM)" method, and it is an adaptation of common recurrent neural networks, initially introduced by Hochreiter and Schmidhuber in 1997. However, multiple slight modifications and alterations to the basic technique have been presented, by many researchers, the years after (Chung, Gulcehre, Cho, & Bengio, 2014).

3.3. Long Short-Term Memory

A Long Short-Term Memory (LSTM) network is a recurrent neural network structure used to attenuate the vanishing gradient problem and thus allows the network to properly get trained using backpropagation (Brownlee, 2016). According to Hochreiter and Schmidhuber (1997), an LSTM network is able to impose nonstop "error flow" through its units, with a gradient-like approach, which, contrastingly to the classic gradient approach, curtails at a certain structure-based point, and thus, does not allow the gradient values to "vanish" or "explode". As a result, LSTMs can produce cutting-edge results even if applied on arduous problems with sequence input data (Brownlee, 2016).

According to Brownlee (2016), an LSTM network consists of layers of "memory blocks", where each block has memory, as well as elements that cause them to be superior to simple neurons, since they assist them to store information and preserve memory in a more effective way. Each distinct block is composed of gates which determine both whether the block is active or not and its output. An illustration of a single LSTM memory cell, and the way on how it operates in time t, is demonstrated in Figure 2.

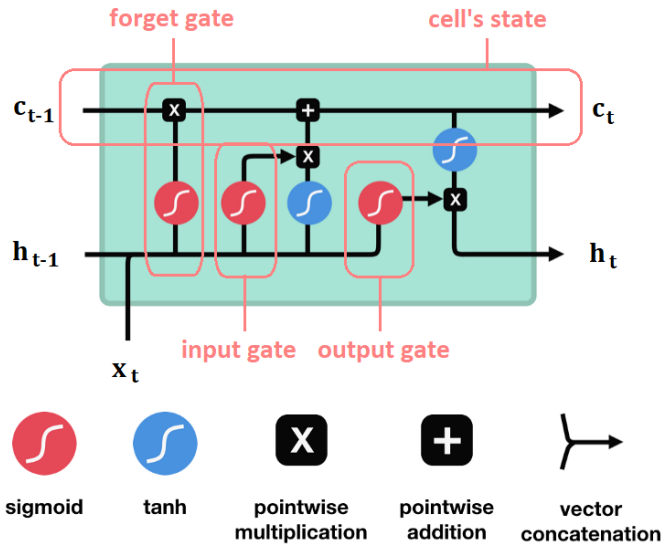

Figure 2: How an LSTM memory cell operates: In time t, the input vector x_t and the previous hidden state h_{t-1} are inserted as inputs to the cell. The forget, input and output gates apply sigmoid activations in order to update their state according to these inputs. The updated forget and input gates, accompanied by the previous cell's state c_{t-1} and a $tanh$ transformations of the inputs, are used to update the cell's state c_t. Finally, a $tanh$ transformation of c_t multiplied by the updated output gate, update the hidden state of the cell h_t *(Phi, 2018)*.

When a sequence input x_t enters a block, the sigmoid function is applied by the gates in order to regulate the state of the block (gets activated or not) and store the corresponding information of this specific input (Brownlee, 2016). According to Brownlee (2016), there are three different kinds of gates: the "input gate", which decides on what information of the input should be stored in order to update the memory of the block (the "cell state" c_t), the "forget gate", which determines what information should be disposed of the memory and what information should be hold, and finally, the "output gate", which controls the output of the block (the "hidden state" h_t) according to its memory (c_t) and the input (x_t). Finally, the output of the cell, the "hidden state" h_t, is responsible for passing the information stored, by the updated cell state and output gate, to the next timestep, and it is the one used for making the predictions of the network. A single LSTM memory cell contains multiple "input gate units", as well as multiple "output gate units" and "forget gate units" (Hochreiter & Schmidhuber, 1997). Therefore, the input, output and forget gates can be mathematically demonstrated as vectors. This, according to Hochreiter and Schmidhuber (1997), restricts non-pertinent input information from spoiling the memory that this single unit preserves, as well as non-pertinent memory from disturbing the next in sequence units.

It is important to pinpoint at this point, that common RNNs' cells also have a hidden state (h_t) – and only this one – which carries the, each time updated, historical information through the next timesteps. However, because of the fact that the RNN's hidden state h_t is updated in a multiplicative way – and so do the gradients –, it often results into the "explosion" or "vanishing" of these gradient values, and thus, to the arduous training of the model. LSTMs, with introducing an extra memory state, the so-called "cell state" c_t, attempt to tackle this exact issue, by still carrying the historical information, but, this time, the update occurs in an additive way. The LSTM's cell state c_t operates in a similar manner as the hidden state h_t does in a common RNN network. When it comes to LSTM architectures, it can be claimed that the hidden state h_t carries all the historical information that the network has gathered throughout its operations, and applies it in the present time, while the cell state caries only a selective part of this memory.

Following the Graves et al. (2013) and Chung et al. (2014) typology, if c^j is a j-th single memory cell, which receives input both from an input gate i^j and an output gate o^j, and uses a forget gate f^j, then, in time t, the cell state c^j of this LSTM unit is updated by:

$$c_t^j = f_t^j c_{t-1} + i_t^j \, tanh(W_{xc} x_t + W_{hc} h_{t-1} + b_c)^j \quad (3)$$

All the gates (forget, input and output gates) use a sigmoid activation which results into output values between 0 and 1. In equation (3), the first part denotes that, when updating the state of a memory cell, the previous cell state is multiplied by the output of the forget gate, which, if it has a value close to 0, results in dropping a certain part of the current memory through the forget gate f_t^j in the following manner:

$$f_t^j = \sigma\left(W_{xf} x_t + W_{hf} h_{t-1} + W_{cf} c_{t-1} + b_f\right)^j \quad (4)$$

However, if this output has a value close to 1, it means that the network should hold this information. The second part of equation (3) indicates that new information passes through the input gate i_t^j, in a way that is demonstrated in equation (5), in order for the cell's state to be updated:

$$i_t^j = \sigma(W_{xi} x_t + W_{hi} h_{t-1} + W_{ci} c_{t-1} + b_i)^j \quad (5)$$

The input gate, by using a sigmoid activation, decides on which elements should be updated by transforming them into values between 0 and 1 as well, where values close to 0 indicate that this element is hardly updated with new information, while 1 denotes that the corresponding element is strongly updated with new information. (Graves, Mohamed, & Hinton, 2013), (Chung, Gulcehre, Cho, & Bengio, 2014).

Distinguishing the conceptual difference between a "cell" and the cell's "hidden state" might be quite intriguing. When referring to updating the "cell" (c_t), what is actually denoted to be updated is the state of the "memory" of the cell, which, as demonstrated above by equation (3), depends on discarding a part of the previous memory (c_{t-1}) through the forget gate, and receiving some input through the input gate. This input includes both new information as well as information from the previous hidden state (h_{t-1}). Then, this cell's state (c_t) contributes to updating the hidden state of this unit (h_t), which can be perceived as the "output" of the cell that transmits the corresponding information to the next in sequence timestep.

After the cell's state c_t is updated, this information, accompanied by the sequence input x_t contribute to activating the output gate of the cell, which regulates what part of the memory will circulate at the time by applying a sigmoid activation σ in the following manner:

$$o_t^j = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)^j \quad (6)$$

with h_{t-1} being the previous hidden state of the cell (Graves, Mohamed, & Hinton, 2013), (Chung, Gulcehre, Cho, & Bengio, 2014).

Finally, the hidden state of the cell is updated according to the cell's updated memory state and the information received from the activated output gate in the following way:

$$h_t^j = o_t^j \tanh(c_t^j) \quad (7)$$

(Graves, Mohamed, & Hinton, 2013), (Chung, Gulcehre, Cho, & Bengio, 2014).

As it can be perceived by the equations (3) and (7) above, the cell state c_t is the memory of the cell, amassed, and which is updated based on the previous output (h_{t-1}), while the hidden state h_t is a "non-linear transformation" of the output gate onto the c_t, which is, in total, the "output" of the cell (Chung, Gulcehre, Cho, & Bengio, 2014).

All gates and states $(i_t, o_t, f_t, c_t, h_t)$ in equations $(3), (4), (5), (6)$ and (7) are vectors, the elements of which are updated accordingly (Graves, Mohamed, & Hinton, 2013). The input, output and forget gates are all controlled by the sequence input x_t and the information from the previous hidden state h_{t-1}. According to Graves et al. (2013), all from-cell-to-gate matrices (W_{co}, W_{cf}, W_{ci}) are "diagonal", denoting that each element in an output, forget and input gate receives information, as input, only from the corresponding element of the cell.

In a text classification problem, when a text document is inserted into the LSTM network, each different word is analyzed at the time. For instance, if the review "I love this product" is inserted into the network, the sequence would be analyzed as depicted in Figure 3: the word: "I" is analyzed first and the cell state as well as the hidden state of the cell is updated. Afterwards, the output information from these states is passed onto the next timestep, accompanied by the new input word "love". The cell's state and hidden state are updated again, and this process keeps on going (See Figure 3).

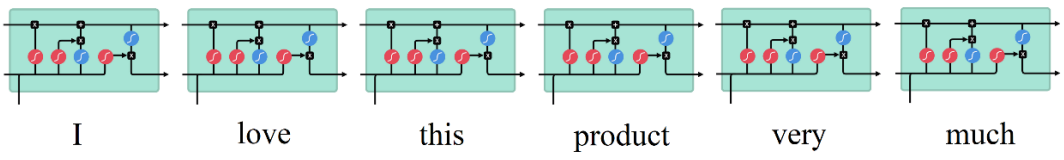

Figure 3: LSTM cells for a text sequence input.

A really simplified explanation on how the algorithm depicted in Figure 3 would work, would be this: Let us suppose that this specific network predicts "negative" if a review is assigned with a value of 0, while it predicts "positive" if a review is assigned with a value of 1. Supposing that the word "love" – a word with a strong positive sentiment – is the input in the current timestep, then, the output of the forget gate would be probably close to 0, denoting that previous information is no longer relevant since this new information is quite important. The input gate would probably treat this value the same way, indicating that the word "love" is important for updating the cell's state. Keeping in mind that the $tanh$ function produces values between -1 and 1, the addition of the forget gate and the input gate would update the cell's state vector with values which indicate a result closer to a "positive" prediction. This value, of the updated cell's memory, will afterwards define the output gate, which is responsible for what the hidden state should be, and thus, what information should determine the network's prediction. The hidden state would also acquire a value which, resulting from a $tanh$ function again, denotes that the prediction yields towards a positive classification. Afterwards, the outputs of the cell's state and the hidden state are carried onto the next timestep accompanied by the next in sequence input: the word "this". Probably, the forget gate would get a value close to 1, denoting that the previous memory (the positive sentiment of the word "love") is important to maintain. The input gate would get a value close to zero, which means that this word is not important for determining the predictions of the network and the cell would not be activated. When the words "very" and "much" pass through the network – words that enhance, in this case, the positive meaning of the word "love" – the cell's state and the hidden state's values would be updated again in a way that they indicate a higher positive outcome, pinpointing the higher possibility of this review text to be positive. The predicted output of the hidden state, after proper transformations have occurred in the final layer, would be a value close to 1, denoting the positive classification of this review.

4. Local Interpretation of a Black-box Model

An LSTM network, just like any type of neural network, is a black-box model, meaning that the way in which the algorithm produces its predictions is unknown – and thus untrustworthy – to anyone, apart from the developers of the program. For instance, in the example above, with the "I love this product" review input, which specific word(s) triggered the algorithm to classify this review as positive? Was it the word "love" or was it the word "product" which, by

coincidence, happened to appear on the rest of the positive reviews that a poorly developed network analyzed? Of course, the latter is a case that is quite undesirable. Therefore, explanations on how the model arrives on a specific prediction are of supreme importance in order for analysts to "trust" the network's predictions and, afterwards, apply the proper (marketing) strategy. In a text classification problem, this would apply on being given an indication of what specific words triggered the network to classify a specific text input as having a positive or negative sentiment. "Local Interpretable Model-agnostic Explanations (LIME)" is a tool that can provide analysts with "local" (individual, on a single document level) interpretations of a prediction made by the LSTM classifier. Local interpretation techniques are the most apt ones when it comes to text classification problems, since the number of explanatory features (words here) is enormous. In a customer review classification case, LIME approaches the LSTM model "locally", by highlighting, each time, the elements (words) that hold the greatest sentiment for a specific review to be assorted to a positive or a negative class.

4.1. LIME

LIME stands for "Local Interpretable Model-agnostic Explanations". "Local" relates to how this algorithm approaches the model in order to produce these explanations and reflects the fact that the explanations provided are trustworthy only on a local scale (i.e. in the neighborhood of the object being explained). What is more, the explanations that LIME produces are "interpretable", meaning that they are plain and comprehensive enough for any human being to conceive them. A neural network or a text classifier usually makes use of complicated techniques and inner structures which are rather vague and incomprehensible to humans. LIME interprets these classifiers with simple and understandable measures (e.g. words), which, however, are not always the measures that the classifier uses (Ribeiro, Singh, & Guestrin, 2016). The LIME algorithm is also "model-agnostic", meaning that it treats every classifier or regressor as a black-box model. Namely, it can be applied on any model, without any assumptions or restrictions needed about this model. In other words, according to Ribeiro et al. (2016), LIME does not need to perceive how the classifier's algorithm operates. On the contrary, it randomly samples the input data, around the prediction attempting to explain, in order to detect how the corresponding prediction responds. Afterwards, the permuted data are being assigned with weights, in a way that they depict how close these random data are to the original input (Ribeiro, Singh, & Guestrin, 2016). For instance, if a prediction that the model strives to explain, has as input the sentence "I love this product", then this input would be randomly sampled into

sentences such as: “I love product”, “I the product”, “love product”, “I love”, and so forth, and the corresponding predictions would be made. Of course, the original model analyzes a greater number of input words, but it makes sense to consider that the word “love” is the most likely one to have led the classifier to arrive at a specific prediction (e.g. classifying this specific review as positive). A modest demonstration on how the LIME algorithm operates on a specific prediction is depicted bellow, in Figure 4.

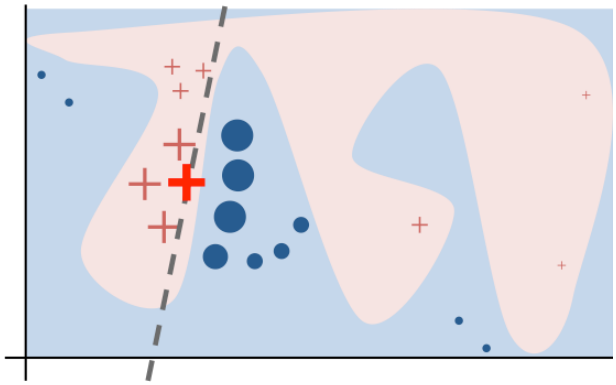

Figure 4: Local explanation produced by LIME: The colored background represents decision areas of the complex binary black-box model, while combinations of the elements of the two axes indicate to which of the two classes an instance belongs. The bold red cross is the instance being explained by LIME, while the rest of the red crosses, as well as the blue dots, are randomly generated samples of the initial instance, which are weighted according to their closeness to it (size represents weight). The red samples are in the “pink” decision region, while the blue samples correspond to the “blue” class. The dashed line is a simple (linear here) explanation produced by LIME for this specific instance, which approximates the black-box model’s behavior locally. *(Ribeiro, Singh, & Guestrin, 2016)*.

The colored backdrop of the figure corresponds to the complex function that the classifier uses to produce its predictions. It represents a binary decision area (e.g. positive or negative rating), while the combinations of the values (e.g. words) on the x-y axes indicate which binary value an instance (e.g. review) can get. Supposing that the bold red cross is the prediction which we wish to explain using LIME, the rest of the crosses and the blue dots are randomly produced sets of data points, which are weighted according to their closeness to the original observation. The red samples are in the “pink” decision region, while the blue samples correspond to the “blue” class. Crosses and dots which are closer to the instance we are interested in have been assigned with greater weight values and thus depicted with a larger symbol than the ones that are more distant. The dashed line is the best approximation (linear here) to the original model’s prediction function that LIME generated after testing the original model on the randomly permuted data points. This is an

extremely simplified function (e.g. usually a linear model or a decision tree) which, to the highest degree, resembles the classifier's function, but only to this specific neighborhood around the instance being explained. The simplified model (explanation) that LIME produced is now quite understandable by a human being, however, as it is clear from Figure 4, it is only trustworthy in this specific neighborhood around the instance examined each time (Ribeiro, Singh, & Guestrin, 2016).

According to Ribeiro et al. (2016), when it comes to text classification, a black-box model usually makes use of word embeddings in order to represent each different word or data point (e.g. of a customer review text) and make its predictions. However, this representation of a single instance (e.g. a review) – let us call it $x \in \mathbb{R}^d$, with d being the dimensions of the space X in which the black-box model operates– is quite complicated for a human being to perceive. A more intelligible representation, that LIME could produce in this case, could be a binary indicator of whether a word appears or not in a certain review, and it could mathematically be represented as $x' \in \{0,1\}^{d'}$ (Ribeiro, Singh, & Guestrin, 2016). d' defines the dimensions of the space X' to which the simplified model applies, with $d' \ll d$, always. This simplified representation can be produced by a model g which is part of a bunch of possible "interpretable models" G (e.g. linear models or decision trees), such that $g \in G$, and we wish for g to be as less complex as possible in order for the explanation to be comprehensible enough (Ribeiro, Singh, & Guestrin, 2016). Therefore, according to Ribeiro et al. (2016), we impose a penalty Ω which denotes the complexity and thus $\Omega(g)$ should be as small as possible.

If LIME applies a generalized linear model in order to produce its explanations, having low complexity $\Omega(g)$ would be equivalent to shrinking quite many weights of the model towards zero (Ribeiro, Singh, & Guestrin, 2016). In a text classification problem, the complexity Ω would be the number of the most influential words (i.e. the words with the strongest sentiment) for classifying a text input as positive or negative. This could be achieved by applying a LASSO regression in order to choose the K most important features (words) – while the weights of the rest of them will be shrunk towards zero – and the number of features K could vary (or not) for the different instances being explained (Ribeiro, Singh, & Guestrin, 2016).

Following the Ribeiro et al. (2016) typology for text classification problems, with f being the black-box model to be explained, $f(x)$ the prediction of this black-box model for the

original representation x (e.g. a review x being classified as positive or negative), and π_x the neighborhood around the instance x, LIME produces its explanations $\xi(x)$ by optimizing the following loss function:

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{10}$$

where $\mathcal{L}(f, g, \pi_x)$ is the infidelity of the simplified model g being a suited approximation of the actual model f in the local π_x neighborhood (Ribeiro, Singh, & Guestrin, 2016). Therefore, LIME's aim is to minimize this infidelity ($\mathcal{L}$), given a low level of complexity ($\Omega(g)$).

In order to do so, the input data points around x' are randomly sampled and weighted according to the proximity π_x. $\mathcal{Z}$ number of samples z', with $z' \in \{0,1\}^{d'}$, are generated and collected, for an original complex representation $z \in \mathbb{R}^d$, and the predictions $f(z)$ are acquired (Ribeiro, Singh, & Guestrin, 2016). In the case of a reviews classification problem, the randomly generated data for a single review, could be produced by applying the values of 0 and 1 to the words of this instance, by indicating, in this way, whether a word appears or not in the perturbated new data. In this way, if, for example, the instance which's prediction we wish to interpret is the review "I love this product", which has an original representation z of word embeddings, then one of the randomly generated data points z': "love product", would be represented as: [0 1 0 1], denoting that only the words "love" and "product" appear in this sample. In Figure 4, two randomly generated samples are demonstrated for the "bold red cross" instance by the sparse red crosses and the sparse blue dots. All of these randomly sampled data points are weighted according to their closeness (π_x) to the instance, and these weights are depicted by adjusting the crosses' size: the ones closer to the instance of interest are larger in size. Afterwards, the predictions (probabilities of a sample sentence being positive or negative) $f(z)$ are produced. A linear, and locally only trustworthy explanation, is demonstrated by the dashed line.

Once again, following the Ribeiro et al. (2016) typology, when it comes to text classification, and for linear interpretable models g, the infidelity loss is defined as:

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z)\big(f(z) - g(z')\big)^2 \tag{11}$$

with:

- $g(z') = \omega_g \cdot z'$ being a set of interpretable generalized linear models, which occur by applying the proper weight value ω_g to the randomly generated data point z', and
- $\pi_x(z) = \exp\left(-\frac{D(x,z)^2}{\sigma^2}\right)$, with D = a cosine distance function, and σ = the width of the neighborhood π_x (Ribeiro, Singh, & Guestrin, 2016).

All in all, when it comes to text classification problems, the result of equation (10) (the explanation $\xi(x)$) would be a bag of K words, which are chosen according to the predictions (probabilities) of the linear model (e.g. K-LASSO) on the randomly sampled dataset $(f(z), x')$, guided by the weights ω. More specifically, $\xi(x)$ is a vector of K variables (words), with the ones with the greater absolute weight values being the most influential ones for the prediction (positive or negative) being made. Words with positive weights indicate that their existence in this certain review causes the classifier to lean towards a positive classification for this specific document, while words with negative weight values are responsible for the opposite result (negative classification), and always by the amount that each different weight represents. For instance, if the classifier predicts that the review "I love this product" is positive with a probability of 95% ($f(z_i) = 0.95$), and the word "love" has a weight value of 0.5, then, if we remove this word from the review, it would be positive with a probability of 95% - 50% = 45%.

In this paper, LIME is applied on an LSTM model, which is trained on customer reviews, and assigns each single instance (review) – of the ones on which it is afterwards tested – to a class (positive or negative) with a probability. Therefore, since this specific LSTM network deals with text classification, the interpretable representations produced by LIME is a "bag of words" with weights assigned to each one of them. Each weight value reflects how much each single word is responsible (or not) for a specific review to be positive (or negative).

4.2. Global Interpretation with Submodular Pick

Although local interpretation can give as valuable insights in evaluating the validity of the black-box model's predictions, an assessment of the total model would be more adequate (Ribeiro, Singh, & Guestrin, 2016). For instance, since the local explanations to be displayed to the user are manually selected, how can we be sure that the explanations chosen are representative and do not only appear once? It would be wise if the most important features (the ones that appear most often in the instances available) were present

to the explanations displayed. Looking at the whole set of explanations would give us a total idea of how the black-box model works, but inspecting a set of hundreds or thousands of explanations would be impossible. Therefore, how can we select a representative set of explanations to inspect, given our limited inspection time and the importance of the features available? When it comes to a text classification problem, LIME, with "submodular pick", gives us the opportunity to look at a large group of instances at the same time, and, accompanied with every instance's explanation that LIME produces, to examine every feature's (e.g. words) importance in determining a classification decision. The maximum set of explanations that a user is willing to examine, B – which is a subset of the total X explanations produced for every instance available – depends on the time, computational resources, etc. that the user has available (Ribeiro, Singh, & Guestrin, 2016). According to Ribeiro et al. (2016), the selection of B out of X total instances is called the "pick step" and it takes into consideration the explanations produced for each instance. That means that the model tries to gather a representative part of the explanations which explain most of the global behavior of the model (high "coverage"), but including only "non-redundant" explanations, meaning that explanations that rarely appear, or similar explanations are neglected (Ribeiro, Singh, & Guestrin, 2016). As Ribeiro et al. (2016) mention, in order for this to be achieved, an "explanation matrix" W is created, which, for a review classification problem would have as rows the available reviews, and as columns the different features (words). The elements (W_{ij}) of the matrix are the absolute weights of each feature computed by LIME. Figure 5 shows how this explanation matrix would be. The total, or "global", importance (I) of a feature j (when a linear model is applied by LIME in order for the explanations to be produced), is calculated by: $I_j = \sqrt{\sum_{i=1}^{n} W_{ij}}$, where n is the number of the instances available (with $n = |X|$), and W the $n \times d'$ explanation matrix. According to Ribeiro et al. (2016), features which are used to explain most of the instances are considered to be more important. Therefore, in Figure 5, $I_2 > I_1$.

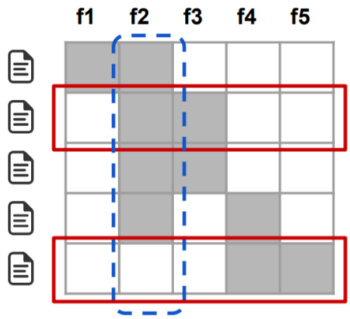

Figure 5: Explanation matrix. Columns are instances (reviews) and rows are features (words). The grey areas indicate non-zero weights of each feature for each corresponding instance. The most important feature is the feature f_2 (blue circled) since it appears in most of the instances explained. The second and fifth instances would be picked under the problem of maximizing the coverage, and they include all the features except for the first one *(Ribeiro, Singh, & Guestrin, 2016)*.

The "pick problem" can be defined as selecting the instances V (with $|V| \leq B$), the explanations of which, altogether, include most of the total features available, but with the most important features being present and with similar and/or rare explanations being neglected (non-redundant explanations). In other words, we have to select V in a way that maximum "coverage" (c) is achieved (Ribeiro, Singh, & Guestrin, 2016). Following the Ribeiro et al. (2016) typology, this can be formulated as follows:

$$Pick(W, I) = \underset{V, |V| \leq B}{\operatorname{argmax}}\, c(V, W, I) \tag{12}$$

where the coverage $c(V, W, I) = \sum_{j=1}^{d'} \mathbb{1}_{[\exists i \in V : W_{ij} > 0]} I_j$ calculates the total importance I of the features that are part of the instances of set V. Therefore, given the example of Figure 5, we can tell that the most important feature is f_2, since it appears in most of the instances explained, and then, f_3 and f_4 follow. Supposing that the maximum number of explanations we wish to inspect is 2 ($B \leq 2$), the model would pick the second instance first, since it contains most of the most important features. Then, the model, in order to select the next explanation, is applying a "greedy" approach, meaning that instance added each time has the next highest coverage from the instance added before, but, at the same time, it is non-redundant. We see that instances 3, 4 and 5, include the next most important features (f_3 and f_4), but instances 3 and 4 contain information (feature f_2) already included in the previously selected instance 2. However, the fifth instance presents totally new information (different features) and thus it is the one picked. With the selection of just these two instances almost all of the features have been covered (Ribeiro, Singh, & Guestrin, 2016).

To sum up, and in order to put it in a simple way, let us suppose that a manager, because of his limited time, is willing to look at 10 explanations, at most, in order to understand how a proposed model works and whether it is a faithful model. Therefore, $B \leq 10$. LIME will operate on all the available instances and will create local explanations for all of them, by assigning weights to every feature present in these explanations. An explanation matrix is created for all these instances and features, and the global importance of each feature is computed. Features that appear in more explanations than others (meaning that they have a non-negative weight in more explanations) are the most important ones. At the end, we can show the manager the most important features as well. When the global importance of each feature is calculated, a representative sample V of at most 10 explanations of the ones that were created, must be chosen and be displayed to the manager (therefore, $|V| \leq 10$). These, at most 10 explanations, must be a representative and "non-redundant" sample (meaning that similar explanations are not repeated) in a sense that most of the features available are included in this set of explanations. This set of explanations is selected in a "greedy" way, by maximizing the coverage function, in the sense that each instance added to the set V each time, has the next highest coverage from the instance added before, but, at the same time, it is non-redundant (meaning that the same features are not repeated). Therefore, the instance with the highest coverage (the one that it includes most of the most important features available) is selected first. Afterwards, the instance which, simultaneously, has a high coverage level and includes new information (new features) is added. This process goes on until all non-redundant explanations with the highest coverage are selected, or until the number of explanations selected reaches the size of 10 (the limit we have set). Now, the manager is able to see a representative set of 10 local – and "sparse", we could say – explanations which contain the most important features. He can now tell if these "important", according to LIME, features carry a semantic validity and if these explanations make sense. If yes, and if decisions are to be made regarding the business' strategy, we can be sure that they will be based on the "right" explanations.

5. Analysis

In this section of the paper, the application of the LSTM architecture is analyzed by being applied on a set of customer reviews on various Amazon products. The prediction accuracy of the LSTM model is also discussed. Afterwards, the LIME algorithm is utilized to interpret the results provided by the recurrent network.

5.1. The Data

For this analysis purposes, we will utilize a dataset with more than 34,000 customer reviews on various Amazon products. The dataset can be found on Kaggle (Kaggle, 2019) and it originates from Datafiniti, an online library of product data for various retail web shops. All the products included in this dataset are manufactured by Amazon and are of various categories, such as: electronic devices (computers, tablets, etc.), accessories and office supplies, house appliances, pet supplies, health and beauty products, and more. The reviews cover a period of almost 9 years, from July 2010 to April 2018. The dataset includes 34,660 observations (customer reviews) of 21 variables each. These 21 variables comprise information about the review itself (such as the text and the title of the review, the date it was written, the rating of the product on a scale from 1 to 5, and so forth), as well as information about the product to which the review corresponds (product name, category, brand, manufacturer, whether the product was bought by the user or not, etc.). Some information about the user (e.g. user id, city and province) is also provided.

However, for the scope of this paper, only the review text and the review rating will be utilized. Therefore, the rest of the variables will be omitted, since they do not provide essential information to the research proposed. What is more, for the purpose of this research, a "sentiment" variable was created, based on the initial "rating" variable: reviews with a rating of 4 or higher will be regarded as having a positive sentiment, while reviews with a rating of 3 or less will be considered negative. Reviews with a 3-star rating could be treated as neutral reviews. However, in doing so, negative aspects of a product, mentioned in 3-star reviews, might be neglected. Therefore, reviews with a rating of 3 are also considered as negative in this specific analysis.

5.2. Application of the Methodology on Amazon Customer Reviews

5.2.1. Data Pre-processing

Data need to be cleaned and pre-processed before inserting them into the network. The first step in the preparation of the data was the transformation of the "sentiment" variable, so that the reviews with a positive sentiment are assigned with a value of "1", while reviews with a negative sentiment are assigned with a value of "0". Furthermore, a small number of really extensive reviews were dropped out, so that only reviews with at most 200 words are included in the analysis. The dataset being analyzed was extremely imbalanced in the sense that the negative reviews made up only a 7%,

approximately, of the total observations. (See Appendix, Figure 13). Therefore, the negative reviews were replicated by 8 times, so that the dataset is characterized by a more balanced allocation between positive and negative reviews. It is important to pinpoint that the dataset was initially split into 80% training and 20% test set, and, afterwards, the training set was further split into 70% training and 30% validation set, and the replication of the negative reviews was performed only on the training set, so that the validation and test set reflect the "real" data. In the training set, the negative reviews now make up more than 40% of the total reviews included (See Appendix, Figure 13). The training set was used to train the LSTM network and the validation set to tune its hyperparameters by evaluating the network's performance each time. Finally, the held-out test set was used in order to evaluate the tuned network's performance.

The next step in the preparation of the data was the "cleaning" of the reviews' text, by removing unnecessary characters and spaces, punctuation, numbers, and transforming all words into lowercase. Stopwords were not removed, since they are important in determining the embedding values of the words, for the embedding layer later added to the network. The word embedding vectors for each different word, which assist in reflecting the meaning of a word and locating similarities, in meaning, among them, are computed according to the surroundings of each single word. Therefore, by removing stopwords, and especially negations such as "not" or "never", the exact meaning of a word could be misconceived.

The reviews, before inserting the LSTM network, need to be transformed, since the network is unable to operate on raw words. Therefore, each different word was "tokenized" by acquiring a unique integer value. For instance, the sentence "this product is great and this is affordable too" could be transformed into a list like: [15 4 2 345 29 15 2 44 87]. In this way, each sentence is a list of "tokens", and we set the maximum length of each different list to 200 tokens (words). Any list (sentence) with less than 200 words is filled up with zeros, so that the total number of tokens of the list reaches the value of 200. In the same manner, the first part of sentences with more than 200 words is cut out, so that these sentences also have a total number of 200 tokens in list. This technique is called "padding", and in the example above, 191 zeroes would be added at the end of the list so that it could appear like: [15 4 2 345 29 15 2 44 87 0 0 0 … 0].

Finally, each different word is extra transformed into a 50-dimensional vector of embedding values, meaning that each word's meaning is expressed through a list of 50 unique elements, and words with similar meaning have a similar representation in this high-dimensional space. For example, the embedding vectors of the words "buy" and "purchase", or "love" and "like", will look quite similar, denoting that these words are synonyms. In this way, the network will be able to perceive, that, when either of these words are used, the expressed sentiment of the customer about a certain product, or aspect of the product, is going to be approximately the same. Adding an embedding layer to the network can be considered as a "pre-training", in the sense that we already let the network know which words have similar meanings (Britz, 2015).

Now the data are ready to be analyzed by the LSTM network.

5.2.2. The LSTM Network

The recurrent neural network utilized for this research, makes use of an embedding layer of 50 cells, a "masking" layer which serves in ignoring the zeros created from padding, and an output layer which is activated by a sigmoid function in order to predict a binary output (positive (1) or negative (0) review). This simplicity of the network intends to tackle the risk of overfitting of the model. With the term "overfitting" someone refers to the estimation of a wrong relationship between the features of the model and the independent variable, because of the fact that the model fits "noise". The more the features of the model, the higher the risk of overfitting. A sign that a model is overfitting would be a really good performance of the network on the training data, but a quite poor one on the validation and test set.

The network was trained for multiple times, and for 100 epochs[4] each time with a batch size[5] of 1000 observations at the time. The performance of the network was each time evaluated on the validation data. Each time, a different number of hidden cells of the LSTM layer was tested, as well as some regularization techniques, such as a "dropout regularization" where a predefined percentage of the cells' outputs or/and the

[4] **Epoch:** a single passing of the network throughout the training data.
[5] **Batch size:** the number of randomly chosen observations, of the training set, on which the network will operate for each different epoch. For instance, if the batch size is 1000, that means that, during the first training epoch the network will randomly select 1000 observations (customer reviews), from the total training set, to operate on them. Afterwards, at the second epoch, a different set of 1000 randomly collected observations will be chosen for the network to be trained on them, and so forth.

activations of certain cells are temporarily neglected. The aim of these kind of "penalties" is to tackle any risk of overfitting.

5.2.3. Local Interpretation with LIME

After the algorithm is trained, LIME utilizes the LSTM network on the test set reviews, makes predictions, and uses an "explainer" function in order to interpret the predictions that the network produces. In this way, it attempts to provide simple to humans explanations by presenting the most important features – words, in our case – that contributed to each specific prediction. The weights of these words, and thus their contribution to a specific prediction, is the output of the LIME algorithm.

5.2.4. Global Explanations

Finally, in order to get a more global idea of the most important features (words), and thus of the trustworthiness of our model's predictions, we aggregate the LIME weights by utilizing the "submodular pick" algorithm on the full test set. The result is a list of the importance values (in absolute numbers) of the words that contributed most in the classification process.

6. Results

In this section of the paper, the results of the methodology, as it was applied according to Section 5, are discussed. Illustrations of some indicative examples are provided, accompanied by the relevant explanations and indications.

6.1. The LSTM Network

After training the network for multiple times, with the different adaptations described above, it was decided that a network with 5 LSTM cells and no further regularization could achieve a quite satisfying trade-off between prediction accuracy and model complexity. More specifically, the trained LSTM model achieved an almost 85% accuracy both on the training and validation set, with a computational time of almost 1 hour. The same level of accuracy was achieved on the test set as well, or, in other words, 85% of the instances in the test set were predicted correctly.

A similar result, but this time using a more complex model with more layers and cells, could be achieved by imposing a regularization on the network, such as a "dropout regularization". For this analysis, both techniques provided similar results, but the network

with no dropout regularization and less cells was chosen for extra simplicity and less computational effort.

A demonstration of how the model performed throughout its training is shown in Figure 6. The model seems to converge to a plateau of an approximately 85% accuracy after the 70th or 80th epoch, meaning that training the model for 70 epochs only is enough for achieving the same result but with even less computational time. The slight fluctuations of the validation loss denote that the model is probably still fitting some noise. However, the aim of this paper is not the development of the perfect LSTM network, but the interpretation of the explanations provided by LIME for a recurrent network for text classification problems. The result of the validation accuracy starting from a higher point than the train accuracy, and the fact that it keeps maintaining higher values than the train one throughout the training process, probably occurs because of the imbalance of the dataset between positive and negative reviews. This relation of the data is also reflected in the validation loss as well. Since the validation dataset does not include replicated negative reviews, they (the negative reviews) only comprise a very small part of the validation data. Therefore, if the network predicts the majority of the positive reviews in the validation set correctly, this is enough for the accuracy there to be quite high. Thus, we can conclude that our model is better at correctly classifying positive reviews, which comprise a larger part of the validation set than the training set.

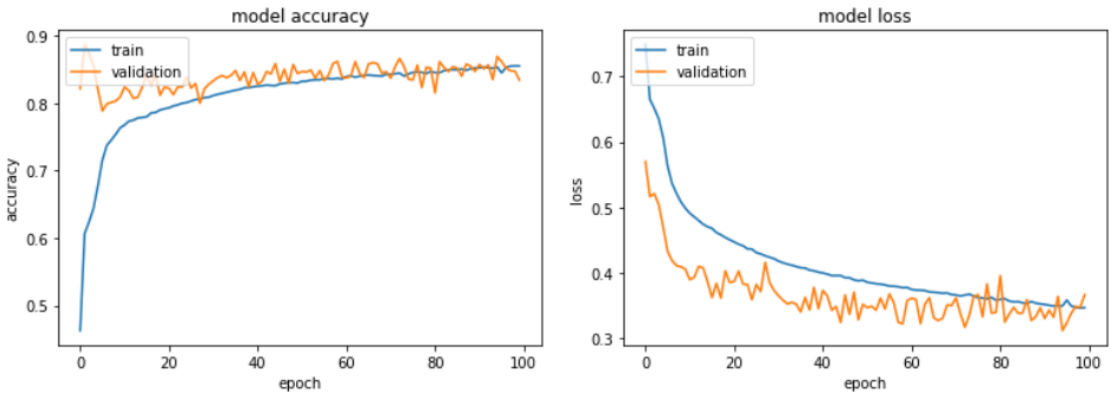

Figure 6: Model accuracy and model loss for training (blue) and validation (orange) data, throughout the LSTM network's training for 100 epochs.

6.2. Local Interpretation with LIME

In Figures 7 and 8 below, two examples of the interpretations provided by the LIME algorithm, on the reviews of the test set, are depicted. Figure 7 shows the interpretation

provided for a (correctly) negatively classified review, while Figure 8 shows the interpretation of why the review explained here was (correctly) classified as positive. In the examples demonstrated below, both instances were classified as negative and positive, respectively, with a 100% probability. The 10 words with the largest weights in absolute value, and thus, the most important ones, are also depicted in the figure. These words are also highlighted, according to their weight value, in the review text, on the right of the figure. The network, throughout its training process, "learned" which specific words are related to negative reviews and which ones are related to positive ones. This "memory" is afterwards applied on the held-out reviews (the test set) and predictions are made accordingly.

The words highlighted in blue, in Figures 7 and 8, are words that the network has related with negative ratings, while the words highlighted in orange, are words that the network has associated with positive ratings. In the review of Figure 7, the word "sucks" has the highest weight that supports the negative classification of this review – a result that makes intuitively sense – while the word "car" has the highest negative impact for this decision. The word "sucks" has a high weight value of 0.36, meaning that if this word was deleted from this specific review, then the probability of this review being negative would be: 100% - 36% = 64%. What is more, the word "movies" has a negative weight denoting that this customer has probably some issues with streaming movies, an aspect for which the company should take notes and probably recommend solutions or alternative products. However, the fact that, although LIME predicted that this review is negative by 100%, and still, words with positive impact appear in the explanation, denotes that the fit of the model on this specific instance is not good enough. In other words, the simplified linear model used by LIME to explain this review, was not the most appropriate fit to this, probably more complex relationship between the data. Therefore, an explanation like this is not trustworthy. For the positively classified review shown in Figure 8, the words "christmas" and "pretty happy" have the highest positive impact in this prediction made. These results are intuitively reasonable considering the positive sentiment these words convey. If the word "christmas", which has a weight value of 0.15, was removed from this review, then, it would be by 15% less positive. The positive weights of "kindle", "kids" and "christmas" probably denote that, firstly, customers who are satisfied with the purchase of the Amazon's "kindle fire tablet" tend to leave more positive reviews, while, secondly, customers who

purchase products for their kids or during Christmas time, also tend to provide a positive feedback.

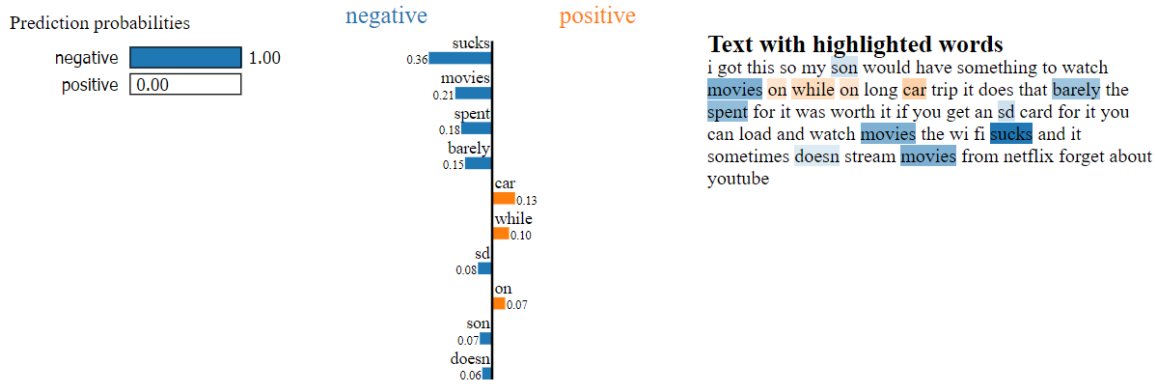

Figure 7: Interpretation provided by LIME for a negatively classified review. The algorithm predicts that this review is negative, with a probability of 100%. The 10 most important words (the words with the highest weights) that contributed to this prediction, are depicted in descending order. Words highlighted in blue color contribute to the review being negative, while words highlighted in orange contribute to the review being positive. The review text is depicted on the right of the figure.

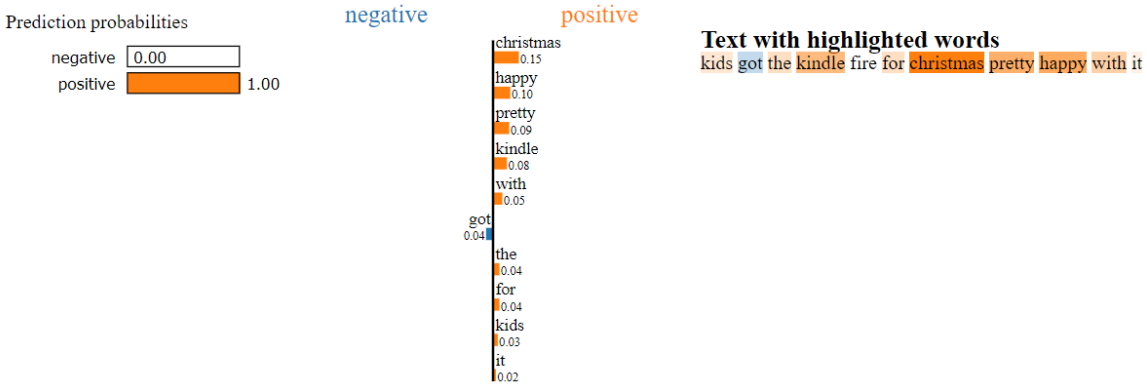

Figure 8: Interpretation provided by LIME for a positively classified review. The algorithm predicts that this review is positive, with a probability of 100%. The 10 most important words (the words with the highest weights) that contributed to this prediction, are depicted in descending order. Words highlighted in blue color contribute to the review being negative, while words highlighted in orange contribute to the review being positive. The review text is depicted on the right of the figure.

Figure 9 shows a prediction for which LIME is not completely sure whether it is a positive or a negative review. More precisely, according to LIME, this review is probably negative, with a probability of 57%, because words such as “runs slow” and “ok” exist in this review. However, there is a 43% probability that this review is positive, since words such as “friday” and “tasks” are part of the review text. Probably, this was a neutral review with a rating of 3 out of 5. Nevertheless, someone can still distinguish the aspects of the product

for which the customer is happy or unhappy about (e.g. the fact that the device runs slow is quite informative for the stakeholders).

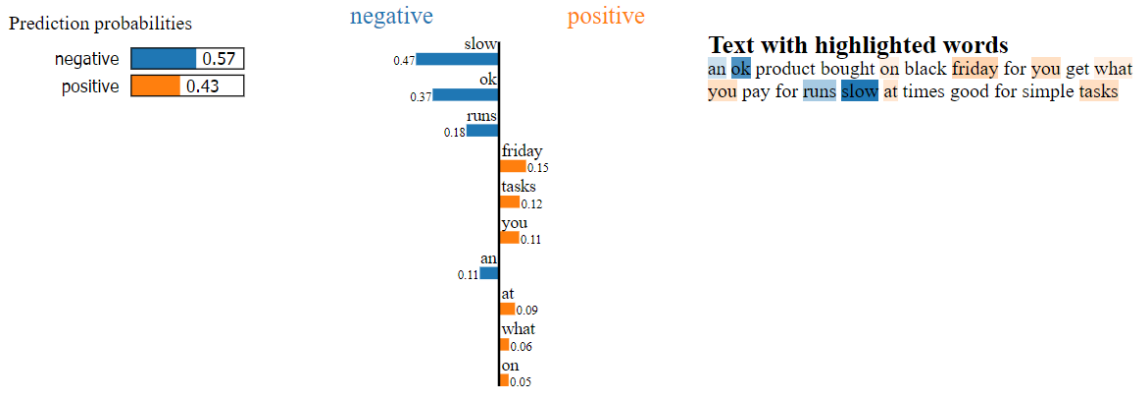

Figure 9: Interpretation provided by LIME. The algorithm is not entirely sure if this is a positive or a negative review, but it suggests that it is probably a negative one by a probability of 57%. The 10 most important words (the words with the highest weights) that contributed to this prediction, are depicted in descending order. Words highlighted in blue color contribute to the review being negative, while words highlighted in orange contribute to the review being positive. The review text is depicted on the right of the figure.

Taking a more attentive look at the interpretations provided by LIME, above, someone can tell that words such as "it", "at", "the", "on", "with", "runs", etc., appear to be connected with the positive or negative classification of each review. This rather weird outcome might be the result of overfitting, or the reflection of the fact that the LSTM algorithm acknowledges the sequence of the words in a sentence. Therefore, the word "runs", for example, was considered to have a negative impact in the review of Figure 9 probably because it was preceding to the negative word "slow". The same could apply for phrases such as "on friday", "for christmas", "the kindle", "happy with it", and so forth, in which, the words "on", "for", "the", "with" and "it" have the same effect (positive or negative) as the word by which the LSTM network has memorized to be accompanied, in the majority of the reviews trained. This effect, however, arises the question of whether stopwords should be removed during the preprocessing of the text, or not, so that meaningless patterns like these are neglected.

Finally, Figure 10 shows a LIME interpretation for a review that was wrongly classified as positive, while it is a negative one. The model predicted that this is a positive review, with a 100% probability, while it undoubtfully conveys a negative sentiment. LIME, with the interpretation of Figure 10, explains why the model arrived at such a conclusion. The reason appears to be that words with high positive weights are part of this review. However,

important – for determining the review's sentiment – words, such as "trouble" (with the "battery"), "not impressed", and "would not repurchase it", seem to be irrelevant or having really low weights (impact) according to the model. Another issue here, is the fact that, at the same time, the model has assigned considerable importance on words such as "having", "and", "in", etc. The word "having", for example, has a quite high and positive weigh value of almost 0.5, denoting that this word, alone, has caused the review to be quite more positive. Although this result does not make sense, the model is a 100% sure that this review is a positive one. This is just one of the examples in which LIME indicates to the user/analyst that something might be wrong with the model (e.g. the model is overfitting or underfitting[6]), and thus, further modifications and optimization need to be done on the LSTM network, before basing any of our decisions on the results of such a model.

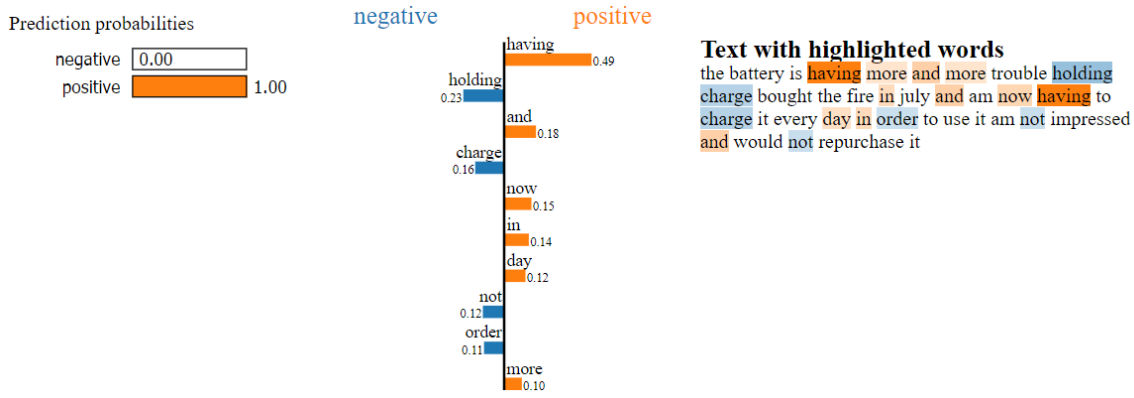

Figure 10: Interpretation provided by LIME for a wrongly positively classified review. The algorithm predicts that this review is positive, with a probability of 100%, while it is actually negative. The 10 most important words (the words with the highest weights) that contributed to this prediction, are depicted in descending order. Words highlighted in blue color contribute to the review being negative, while words highlighted in orange contribute to the review being positive. The review text is depicted on the right of the figure.

6.3. Global Explanations

If we wish to grasp an idea of the most important features from a global perspective, LIME gives us the opportunity to aggregate the weights of the explanations produced for the whole test set in order to examine the total effect of each different element. In this way, we can detect which words were the most influential, in total, in classifying the instances analyzed. Figure 11 shows the 25 words with the highest importance according to the

[6] By "underfitting" we refer to the case of the model not being complex enough to capture all the hidden patterns among the data points.

submodular pick algorithm, after it was applied on the whole test set. Therefore, this graph gives us just a general sense of what the most important words are in most of the explanations. For example, by looking at Figure 11, we can tell that most of the explanations have been based on the existence of the term "not" and/or "great" in the review text. A bit less often, but still important for the interpretation process, is the detection of terms such as "love", "easy", "good", "slow" etc., but also of various stopwords, another sign that the LSTM network is probably overfitting. "TV" is also one of the most important words, meaning that many reviews were about TVs. However, we are not able to tell which classification decisions were led by which words.

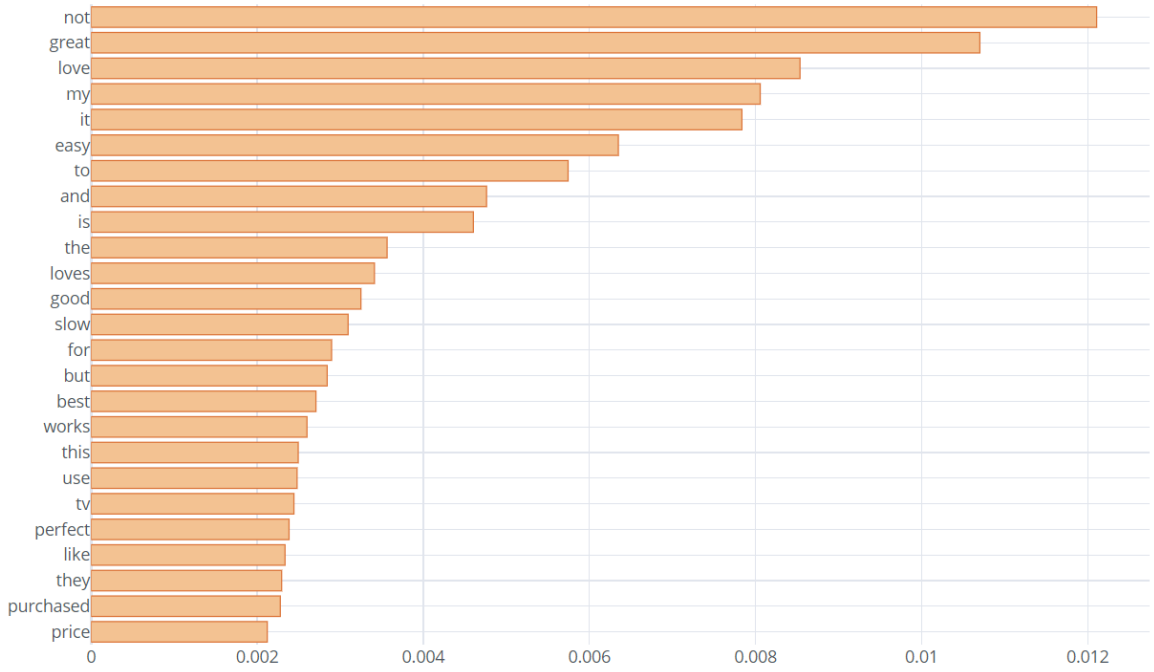

Figure 11: The 25 words with the highest importance (computed using the absolute LIME weights) in explaining the instances of the test set.

The submodular pick algorithm selected for us a set of 10 representative explanations, which include these 25 important features, in order to get a better idea of the positive or negative weight value of each feature, and thus, conclude to the relevant inferences. Figure 12 shows the 6 first explanations, while the rest of them are present in Figure 14 in the Appendix section. By looking at these explanations we can see that words such as "love", "excellent", "easy", "great", "good", "perfect", etc. have been assigned with positive weights, while words such as "issues" and "slow" have negative weight values – a result that makes sense. However, many stopwords still appear in the explanations.

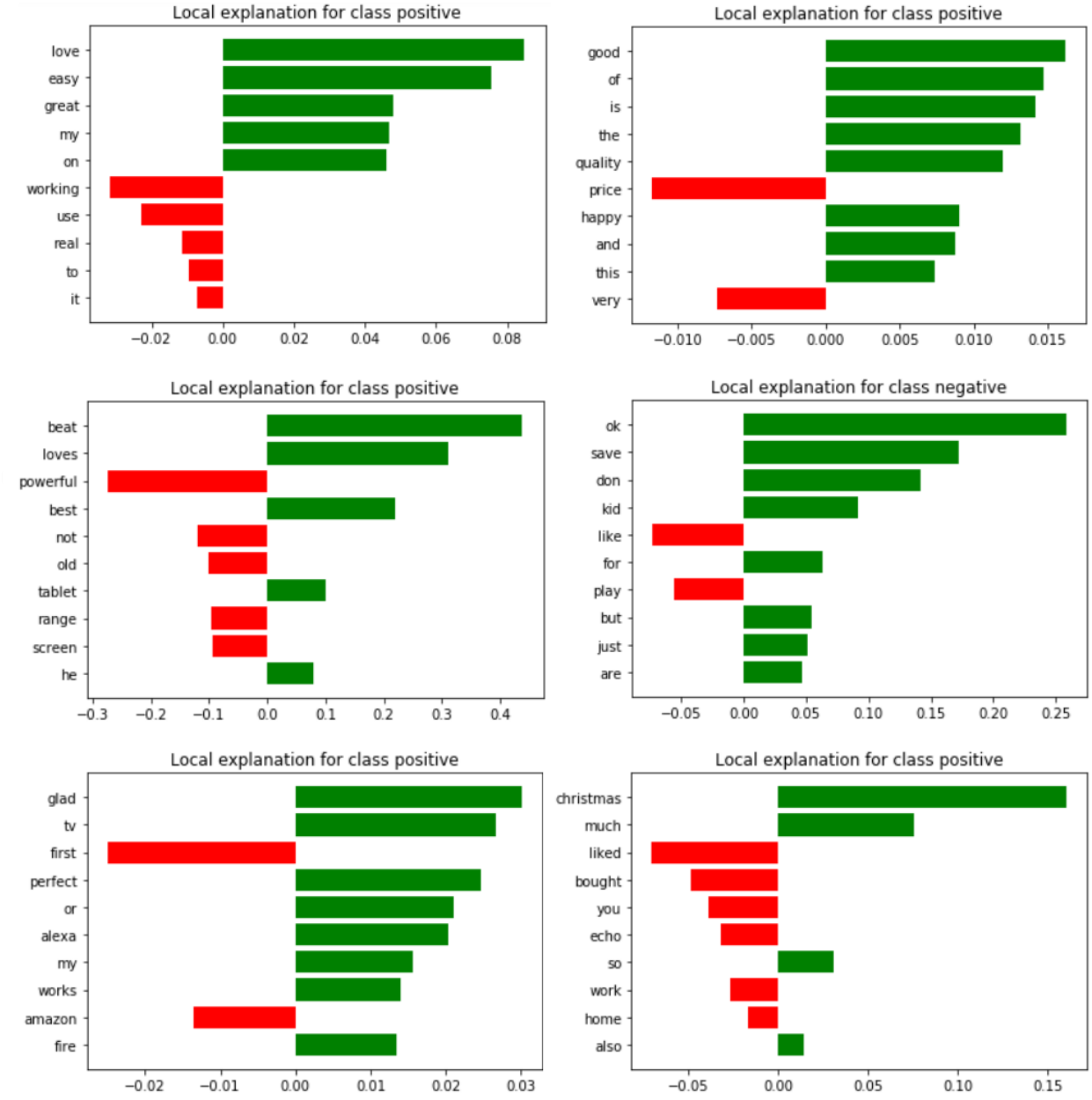

Figure 12: A representative set of explanations selected by the submodular pick algorithm. The features with green (positive) weights support the classification decision (positive or negative), while the features with red (negative) weights contradict it.

Since the explanations of Figure 12 (and 14) are a "representative" part of the whole set of explanation produced by LIME, we can make some conclusions out of them. For example, we can infer that customers who tend to leave a positive review are people who have purchased "gifts" for "family members", they discuss about "video" quality, the "kindle fire" tablet or "tablets" in general, "music", "alexa", and they seem to be satisfied with the "size" of the product, its "quality", the "money" they spent on it, etc. All of these aspects of the products have been assigned with a positive weight value. On the other hand, people who tend to leave negative reviews are the ones who are dissatisfied with "amazon" (in general), "games",

"books", the "price" of a product, its "light" quality and/or its "screen", they face "charging" issues, etc. All of these aspects of the products have negative weights. Now, the manager of Amazon can acquire a general idea of what their customers are satisfied or dissatisfied with, and proper decisions can be taken. For instance, they can improve the aspects of the products for which customers are negative about, they can provide solutions or alternatives to the dissatisfied customers, or promote the appropriate products to the satisfied customers (e.g. similar books to the ones they liked, a smartwatch compatible with a smartphone they purchased, etc.)

7. Closing Remarks

Explanations provided by local interpretation techniques can assist analysts in understanding the underlying patterns of any (black-box) model's predictions. In the examples provided above, an analyst can tell whether the resulting weights make sense or not, and if not, then this is an indication that the model's high accuracy is more than misleading. Consequently, proper modifications need to be made to the algorithm; modifications that address the different issues each time. For instance, taking the examples provided in Section 6.2. into consideration, two possible adjustments to the LSTM network would be either to try tackling any possible issue of overfitting by imposing some form of regularization to the network (e.g. dropout rate, weight regularization, including fewer features, etc.), or to remove stopwords (but probably not negations such as "no", "not" and "never").

The explanation provided by LIME, in Figure 10, is an example only of how high accuracy of a model might be based on the co-occurrence of the wrong features. An analyst should not just rely on the performance of a model, by means of statistical accuracy, but they should rather inspect and evaluate the validity of the model's results. Developing a highly accurate model but not knowing how it works is not a promising achievement. Being able to comprehend and thus trust the model is the crucial part. Interpretation techniques serve this exact objective, by unveiling to the analysts how any regressor or classifier makes its predictions. LIME is a local interpretation technique that can guide analysts towards this way, by indicating the features that triggered each specific prediction, and thus, pinpointing towards which direction any adjustments needed have to be made. By allowing us to zoom in into every single instance, the explanations provided by LIME are instance-specific, something that can provide analysts with even more targeted amendments to an imperfect model.

In this paper, the analysis of a text classification problem, by employing an LSTM network, was carried out, and, afterwards, the predictions of this black-box classifier were explained with the usage of a local interpretation technique, the "Local Interpretable Model-agnostic Explanations (LIME)". More specifically, the network operated on a set of Amazon's customer reviews, in order to classify them into positive or negative ones. It reached an accuracy level of around 85%, but the hidden patterns behind the algorithm, as well as the validity of this accuracy, were obscure. Therefore, trusting this model and its predictions is unreasonable. For that reason, the LIME algorithm was employed in order for the network's concealed functioning to be hinted. More precisely, LIME, based on the LSTM's memory which it preserved throughout its training, approached each single instance (customer review) individually, and pinpointed the most crucial features (words) that determined each instance's classification decision. By looking at these interpretations (or explanations), which are in the form of lists of weights, we could tell whether each classification decision "made sense" or not. In other words, we could tell whether the most important words – as they were indicated by LIME – actually carry an important semantic significance which justifies their position in the important-words list, and thus their power in determining a prediction's outcome. If the predictions are based on features (words) which do not "make sense" (e.g. the word "having" in Figure 10), then, this is an indication that the model's predictions are based on the wrong features. These features might just happen to co-exist with the "right" ones in this specific dataset, and thus the prediction accuracy remains high. This is an indication that our model is not trustworthy enough, and proper alterations and modifications need to be made to the network, so that the predictions are not determined by absurd features. As already mentioned before, a possible adjustment to the LSTM network of this specific analysis, could be to impose some form of "penalty" on the network, by means of a regularization technique, or to drop most of the unnecessary stopwords. After these modifications take place, LIME should be applied again in order to re-evaluate the validity of the networks results. This process should keep going until LIME has no more indications, for network adjustments, to make. In other words, we should stop when we are entirely sure that we can "trust" the network.

7.1. Marketing Implications

Apart from the scientific inferences described above, the applications of LIME can have a crucial contribution to the fields of marketing and management as well. First and foremost, the fact that a local interpretation technique, such as LIME, can assist analysts in proving the validity of their model, reinforces their arguments towards persuading a manger to use

this model. More specifically, they are able to convince a manager that the model they have developed is a "good" model, not only because of its high prediction accuracy, but also because of the fact that they can present, to the manager, tangible explanations, comprehensible enough to them, of why this is the case. In this way, the manager will be able to acquire a slight idea of how the model works, something which significantly enhances the chances of him/her accepting the proposed model. The same applies to the case of an analyst developing programs for a third party (businesses as consumers). Trusting the quality of what you buy has always been a precedence. Therefore, a customer wishes to be able to understand and trust the program for which they pay. The simple explanations provided by LIME is a convenient way for an analyst to gain their manager's or customer's trust. What is more, although we advise to think that high accuracy of a model does not always abide by trustworthiness of it, it is not unreasonable to consider a model with a really high level of accuracy (e.g. 95%, or even 99%) to be trustworthy by just looking at this accuracy. Even in this case, LIME can be used in order to faithfully establish this claim to a manager or customer.

A second part of the marketing contribution of LIME to the business field, lies in the fact that, since a local interpretation technique enables us to zoom in into and analyze every single instance independently, the resulting explanations can assist us in enhancing personalized marketing techniques. In this contemporary era, where customer targeting is shifted towards a more personalized strategy, in a way that each customer's specific needs get satisfied, sending the right message to the right person is of supreme importance. By analyzing consumer reviews and interpreting the results using a local interpretation techniques, just like in the example of Section 6.2., the resulting explanations can be considered as indications of the customer's sentiment or opinion towards specific aspects of the product. The review rating itself is undoubtedly an indication of the sentiment of the consumer about the product, but in a sense of a "mean" value. The explanations produced by LIME can give us further and more specific sentiment information about the different aspects of the product that compose the total sentiment or rating of the consumer. For instance, the customer in the example of Figure 7 seems to face issues with streaming "movies", the one in Figure 8 is positive towards the "kindle" fire tablet, and the customer of Figure 10 is facing some "charging" issues, all according to the sign of the weights with which the corresponding words were assigned (and only if the corresponding interpretations are trustworthy). Therefore, a marketeer appears to be in a position where

they are able to comprehend what types of products, or what aspects of a product, each customer is happy or unhappy about. In this way, promotions of the wrong products to the wrong customers could be avoided, and, in the same manner, promotions of similar products, or compatible products, with the ones that the customer is satisfied with or has already purchased, can be prompted. What is more, by understanding the aspects of a product for which consumers are dissatisfied with, the apt signs about recommended ameliorations to a product can be conceived. Finally, the strongest advantages of a product, according to consumers' opinion, can be highlighted as the company's competitive advantage in the upcoming promotions of the product or of a similar product.

7.2. Limitations of the Model

Just like any other model, the models presented in this paper (LSTM and LIME) have some limitations, in the sense that they are not "perfect" or they might not be the optimal tool or solution for some certain cases.

First of all, it is essential to pinpoint that the aim of this report was not the development of the "perfect" recurrent neural network, but the interpretation of it using a local interpretation technique. Therefore, the LSTM network presented in this paper was lacking some substantial elements. For instance, a better hyperparameter tuning should have taken place, so that the optimal number of hidden layers, cells, epochs, batch size, etc. are chosen. In this way, any risk of overfitting would be avoided, and the most accurate results as possible would be obtained. A chance that our model is underfitting is also possible since words such as "trouble" or "impressed" – words which seem to hold a strong positive or negative sentiment – appear as not important in the LIME explanations of Figures 9 and 10. Whether the model is overfitting or underfitting needs to be investigated further and the proper tuning needs to take place. What is more, since LIME indicated that certain stopwords appear to carry a significant effect on the resulting predictions, the possibility of excluding the stopwords out of the model – except for negations – should be considered. The models' predictions should be analyzed both with and without the stopwords as part of the sentences, and, afterwards, the best model should be chosen. However, the decision of not removing the stopwords, in this proposed work, was based on the claim that they are an indispensable part in computing the words' embedding values. Finally, the application of a deep recurrent neural network – or, of any deep artificial neural network, in general – on big data problems, consume excessive computational time and memory. In some cases,

the application of a less complex model might be the suitable solution, especially in cases where data relationships are characterized by less complex relations (e.g. linear) and in problems in which time limitations are present.

LIME is also a model which is characterized by imperfections when it comes to certain aspects. A significant limitation of local interpretation techniques, lies in the fact that since they analyze each single instance individually, obtaining the explanations of all instances, in order to perceive a more general view, would be extremely time consuming. In cases were the amount of data to be analyzed is quite big, a global interpretation technique would be probably preferable. Another important drawback of LIME is that the selection of the family G of interpretable representations (e.g. linear models) might not be the appropriate choice in all cases (Ribeiro, Singh, & Guestrin, 2016). For instance, in the case of nominal variables, continuous variables, or when the input data are characterized by extreme non-linearity, then the choice of a linear, for example, interpretable representation, might fail to capture all the underlying patterns of the data, even at a local scale (Biecek & Burzykowski, 2020), (Ribeiro, Singh, & Guestrin, 2016). Therefore, the resulting explanations will lack fidelity, and, consequently, further and/or better representations need to be developed for certain types of input variables (Ribeiro, Singh, & Guestrin, 2016), (Biecek & Burzykowski, 2020). In our example, we saw that the fit of the linear model that LIME utilizes by default to produce its interpretable representations was probably not the most appropriate fit for all the instances, since, elements which did not support a specific prediction appeared as important. Probably, a better tuning of the model's hyperparameters (e.g. kernel width, distance function to be applied, number of data permutations, etc.), could lead to a better fit and thus more trustworthy explanations. A third limitation of LIME is considered to be the fact that the definition of the neighborhood π_x is still quite vague, as well as the fact that exiguous alterations to the width of the neighborhood σ can cause the resulting explanations to significantly vary (Molnar, 2020), (Biecek & Burzykowski, 2020). A further issue that might occur, is the "instability" of the explanations, meaning that opposed explanations might be obtained, for the same instance, if the data get re-sampled (Molnar, 2020). Finally, as Molnar (2020) claims, the sampling process of the algorithm needs to be enhanced, since important relations between the data might be neglected because of the current distribution (Gaussian) that the algorithm utilizes to generate the new data points.

As Linden et al. (2019) claim, the submodular pick algorithm is also deficient, especially when it comes to text classification problems. More specifically they state that, because of the sparseness of the text data, the formula that the submodular pick algorithm applies to compute the global importance of each feature, neglects the features (words) that rarely appear, while it assigns great importance to the features that appear really often. Calculating the features' global importance by following an average-like approach is also a defective solution, especially in the case when a feature is important in predicting a specific class but insignificant for the other class. By calculating an average importance, we would get a rather mediocre importance value for this feature, although it is significantly important in the prediction of a specific class (Linden, Haned, & Kanoulas, 2019). Therefore, they suggest that the homogeneity of each feature's effect should be considered when calculating their global importance – as it is expressed throughout their occurrence, or not, in different classes – and they propose an alternative global importance measure: the "global homogeneity-weighted importance". These suggestions, as well as other potential adjustments to the submodular pick algorithm, need to be taken into consideration and further research should be conducted.

7.3. The Future Ahead

LIME is one of the most promising techniques for locally interpreting a black-box model. The advantage of the simplicity of the explanations it produces, accompanied by other valuable features, has made its usage quite popular to many science fields, and its performance on image and text data is phenomenal. The fact that many models and the way in which they operate can now be unveiled, in a way understandable to any human being, is a crucial breakthrough for the data science field. Being able to comprehend a model, and thus trust it, is a priceless asset, with numerous beneficial outcomes (both scientific and marketing ones), and LIME, worthily, leads stakeholders towards this way.

However, the algorithm is still in development process, and various adjustments and improvements need to take place in order for its usage to be undoubtful and the explanations produced to be entirely faithful. Finally, when it comes to black-box models, such as artificial neural networks, the usage of an external program, like LIME, for uncovering the model's hidden structures and assessing the validity of its results, might be quite strenuous. The ability of developing self-produced explanations by the black-box model itself, would be a significant step forward of the data science field.

8. References

A. Vellido, P. J. (1999). *Expert Systems with Applications.* Retrieved from https://www-sciencedirect-com.eur.idm.oclc.org/science/article/pii/S0957417499000160

Agarap, A. F., & Grafilon, P. M. (2018). *Statistical Analysis on E-Commerce Reviews, with Sentiment Classification using Bidirectional Recurrent Neural Network.* Retrieved from https://arxiv.org/pdf/1805.03687.pdf

Allaire, J. (2018). *Deep Learning with R* (Manning Publications ed.). Retrieved from https://learning-oreilly-com.eur.idm.oclc.org/library/view/deep-learning-with/9781617295546/

Bagheri, A., Saraee, M., & Jong, F. d. (2013). *Care more about customers: Unsupervised domain-independent aspect detection for sentiment analysis of customer reviews.* Retrieved from https://www-sciencedirect-com.eur.idm.oclc.org/science/article/pii/S0950705113002414

Bastani, O., Kim, C., & Bastani, H. (2019). *Interpreting Blackbox Models via Model Extraction.* Retrieved from https://arxiv.org/pdf/1705.08504.pdf

Biecek, P., & Burzykowski, T. (2020). *Explanatory Model Analysis: Explore, Explain and Examine Predictive Models.* Retrieved from https://pbiecek.github.io/ema/LIME.html

Britz, D. (2015). *Recurrent Neural Network Tutorial, Part 4 – Implementing a GRU/LSTM RNN with Python and Theano.* Retrieved from http://www.wildml.com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano/

Broß, J. (2013). *Aspect-Oriented Sentiment Analysis of Customer Reviews Using Distant Supervision Techniques.* Retrieved from https://refubium.fu-berlin.de/bitstream/handle/fub188/6693/Dissertation_Juergen_Bross.pdf?sequence=1&isAllowed=y

Brownlee, J. (2016). *Crash Course in Recurrent Neural Networks for Deep Learning.* Retrieved from https://machinelearningmastery.com/crash-course-recurrent-neural-networks-deep-learning/

Brownlee, J. (2017). *What Are Word Embeddings for Text?* Retrieved from https://machinelearningmastery.com/what-are-word-embeddings/

Chaudhuri, A., & Ghosh, S. K. (2016). *Sentiment Analysis of Customer Reviews Using Robust Hierarchical Bidirectional Recurrent Neural Network.* Retrieved from https://link-springer-com.eur.idm.oclc.org/chapter/10.1007/978-3-319-33625-1_23

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.* Retrieved from https://arxiv.org/pdf/1412.3555.pdf

Gräbner, D., Zanker, M., Fliedl, G., & Fuchs, M. (2012). *Classification of Customer Reviews based on Sentiment Analysis.* Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.220.8308&rep=rep1&type=pdf

Graves, A., Mohamed, A.-r., & Hinton, G. (2013). *Speech Recognition with Deep Recurrent Neural Networks.* Retrieved from https://arxiv.org/pdf/1303.5778.pdf

Helmini, S. (2019). *All you need to know about RNNs, A beginner's guide into the implementation and data manipulation inside a RNN in TensorFlow.* Retrieved from https://towardsdatascience.com/all-you-need-to-know-about-rnns-e514f0b00c7c

Hochreiter, S., & Schmidhuber, J. (1997). *LONG SHORT-TERM MEMORY.* Retrieved from https://www.bioinf.jku.at/publications/older/2604.pdf

Kaggle. (2019). *Consumer Reviews of Amazon Products, dataset acquired from: https://www.kaggle.com/datafiniti/consumer-reviews-of-amazon-products#1429_1.csv.* Retrieved from https://www.kaggle.com/datafiniti/consumer-reviews-of-amazon-products#1429_1.csv

Kasun, L. L., Zhou, H., Huang, G.-B., & Vong, C. M. (2013). *Representational Learning with Extreme Learning Machine for Big Data.* Retrieved from https://pdfs.semanticscholar.org/8df9/c71f09eb0dabf5adf17bee0f6b36190b52b2.pdf

Khan, A., Baharudin, B., & Khan, K. (2011). *Sentiment Classification from Online Customer Reviews Using Lexical Contextual Sentence Structure.* Retrieved from https://link-springer-com.eur.idm.oclc.org/chapter/10.1007/978-3-642-22170-5_28

Kiritchenko, S., Zhu, X., Cherry, C., & Mohammad, S. M. (2014). *Detecting Aspects and Sentiment in Customer Reviews.* NRC-Canada. Retrieved from https://www.aclweb.org/anthology/S14-2076.pdf

Koehrsen, W. (2018). *Beyond Accuracy: Precision and Recall.* Retrieved from https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c

Linden, I. v., Haned, H., & Kanoulas, E. (2019). *Global Aggregations of Local Explanations for Black Box models.* Retrieved from https://arxiv.org/pdf/1907.03039.pdf

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). *A Critical Review of Recurrent Neural Networks for Sequence Learning.* Retrieved from https://arxiv.org/abs/1506.00019

Liu, P., Joty, S., & Meng, H. (2015). *Fine-grained Opinion Mining with Recurrent Neural Networks and Word Embeddings.* Retrieved from https://www.aclweb.org/anthology/D15-1168.pdf

Mishra, S., Sturm, B. L., & Dixon, S. (2017). *LOCAL INTERPRETABLE MODEL-AGNOSTIC EXPLANATIONS FOR MUSIC CONTENT ANALYSIS.* Retrieved from https://pdfs.semanticscholar.org/8b0a/16b2475798ed1b5f5cffed62a47e1ef30c7d.pdf

Molnar, C. (2020). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable.* Retrieved from https://christophm.github.io/interpretable-ml-book/lime.html

Olden, J. D., & Jackson, D. A. (2002). *Illuminating the ‘‘black box’’: a randomization approach for understanding variable contributions in artificial neural networks.* Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0304380002000649

Palmatier, R., & Sridhar, S. (2017). *Marketing Strategy, Based on First Principles and Data Analytics.*

Peltola, T. (2018). *Local Interpretable Model-agnostic Explanations of Bayesian Predictive Models via Kullback-Leibler Projections.* Retrieved from https://arxiv.org/abs/1810.02678

Phi, M. (2018). *Illustrated Guide to LSTM’s and GRU’s: A step by step explanation.* Retrieved from https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Rai, B. (2019). *Advanced Deep Learning with R: Become an expert at designing, building, and improving advanced neural network models using R.* Retrieved from https://books.google.nl/books?id=jAjFDwAAQBAJ&pg=PA323&lpg=PA323&dq=lime+for+generative+adversarial+neural+network+in+r&source=bl&ots=OQFs1fNN56&sig=ACfU3

U2M2rBh0eh1h648zFgjzVGaFvt9Gg&hl=el&sa=X&ved=2ahUKEwiB99z8q-znAhVEqaQKHXvzDLUQ6AEwAXoECAsQAQ#v=onepag

Reno, N. (2014). *Microsoft Cloud Revenues Leap; Amazon is Still Way Out in Front.* Retrieved from https://www.srgresearch.com/articles/microsoft-cloud-revenues-leap-amazon-still-way-out-front

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). *"Why Should I Trust You?" Explaining the Predictions of Any Classifier.* Retrieved from https://arxiv.org/pdf/1602.04938.pdf

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). *Local Interpretable Model-Agnostic Explanations (LIME): An Introduction.* O'REILLY. Retrieved from https://www-oreilly-com.eur.idm.oclc.org/content/introduction-to-local-interpretable-model-agnostic-explanations-lime/

Ritholtz, B. (2017). *The Big Four of Technology, Life will never be the same after Apple, Facebook, Amazon and Google.* Retrieved from https://www.bloomberg.com/opinion/articles/2017-10-31/the-big-four-of-technology

Rivas, T. (2017). *Ranking The Big Four Tech Stocks: Google Is No. 1, Apple Comes In Last.* Retrieved from https://www.barrons.com/articles/ranking-the-big-four-internet-stocks-google-is-no-1-apple-comes-in-last-1503412102

Schuster, M., & Paliwal, K. K. (1997). *Bidirectional Recurrent Neural Networks.* Retrieved from https://maxwell.ict.griffith.edu.au/spl/publications/papers/ieeesp97_schuster.pdf

Sethi, V. (2019). *Types of Neural Networks (and what each one does!) Explained.* Retrieved from https://towardsdatascience.com/types-of-neural-network-and-what-each-one-does-explained-d9b4c0ed63a1

Shah, J. (2017). *Neural Networks for Beginners: Popular Types and Applications.* Retrieved from https://blog.statsbot.co/neural-networks-for-beginners-d99f2235efca

Shankaranarayana, S. M., & Runje, D. (2019). *ALIME: Autoencoder Based Approach for Local.* Retrieved from https://arxiv.org/pdf/1909.02437.pdf

Sharma, N., Jain, V., & Mishra, A. (2018). *An Analysis Of Convolutional Neural Networks For Image Classification.* Retrieved from https://www.sciencedirect.com/science/article/pii/S1877050918309335

Sousa, I. P., Vellasco, M. M., & Silva, E. C. (2019). *Local Interpretable Model-Agnostic Explanations for Classification of Lymph Node Metastases.* Retrieved from https://www.mdpi.com/1424-8220/19/13/2969

Tamagnini, P., Krause, J., Dasgupta, A., & Bertini, E. (2017). *Interpreting Black-Box Classifiers Using Instance-Level Visual Explanations.* Retrieved from https://dl.acm.org/doi/abs/10.1145/3077257.3077260

Venkatachalam, M. (2019). *Recurrent Neural Networks.* Retrieved from https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce

Wei, C.-P., Chen, Y.-M., Yang, C.-S., & Yang, C. C. (2009). *Understanding what concerns consumers: a semantic approach to product feature extraction from consumer reviews.* Retrieved from https://link-springer-com.eur.idm.oclc.org/article/10.1007/s10257-009-0113-9

Yochanan, S. (2002). *Applying Artificial Neural Networks to Business, Economics and Finance.* The City College of the City University of New York and The University of Pennsylvania.

Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.198.4982&rep=rep1&type=pdf

Zafar, M. R., & Khan, N. M. (2019). *DLIME: A Deterministic Local Interpretable Model-Agnostic Explanations Approach for Computer-Aided Diagnosis Systems.* Retrieved from https://arxiv.org/abs/1906.10263

Zhang, P. G. (2004). *Neural Networks in Business Forecasting.* USA and UK: IRM Press. Retrieved from http://pdfs.semanticscholar.org/fa4b/7853fa8ca2ff524d2b8cd847ef245aa3f0c6.pdf

Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018). *Information Fusion, A survey on deep learning for big data.* Retrieved from https://www-sciencedirect-com.eur.idm.oclc.org/science/article/pii/S1566253517305328

9. Appendix

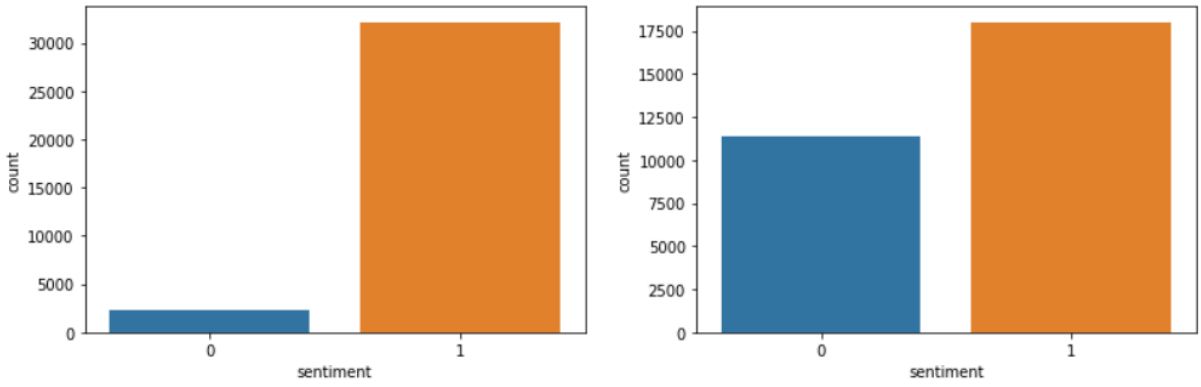

Figure 13: LEFT FIGURE: The imbalanced data set. RIGHT FIGURE: The training data set after the replication of the negative reviews by 8 times. This is clearly a more balanced allocation between positive and negative reviews.

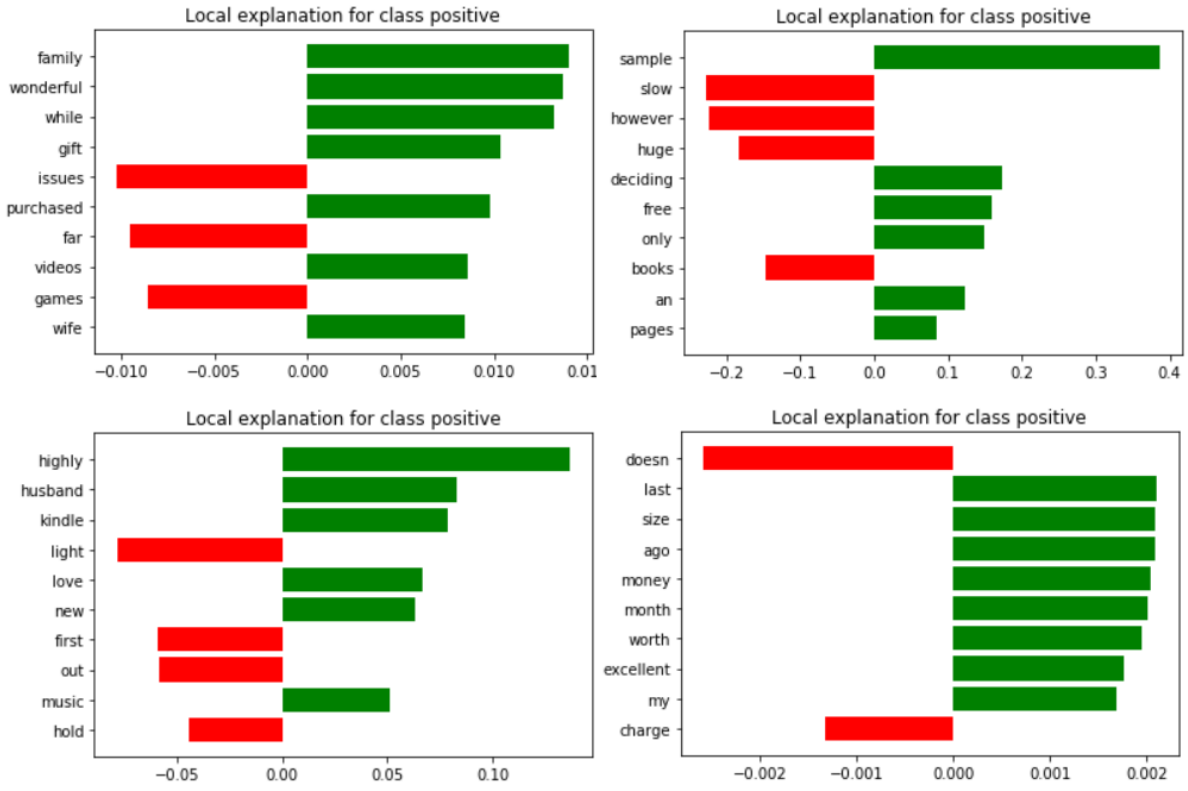

Figure 14: A representative set of explanations selected by the submodular pick algorithm. The features with green (positive) weights support the classification decision (positive or negative), while the features with red (negative) weights contradict it.