Masters thesis

Using Sentiment Analysis on online product reviews for determining fairness

Zabek, A.

___

Using Sentiment Analysis on online product reviews for determining fairness

A thesis submitted to Middlesex University in partial fulfilment of the requirements for the degree of MSc by Research

Aneta Elzbieta Zabek

M00583551

School of Science and Technology

Middlesex University London

December 2022

School of Science and Technology

Student Name: Aneta Elzbieta Zabek
Student ID Number: M00583551
MSc by Research

I hereby confirm that the work presented here in this report and in all other associated material is wholly my own work.

Signature: Aneta Elzbieta Zabek
Date: 13/12/2022

Abstract

Product reviews became one of the most relevant ways customers have to make up their mind about buying specific products. The relevance of these reviews tempts companies to either use them to attack their rivals or to oversell their products by providing misguiding information that does not fit with the real product's characteristics. Identifying this unfair situation is complicated but, at the same time, crucial to guarantee the reliability on the customer's choice. In this work, we aim to simplify unlawful reviews by providing a simile mechanism. Our hypothesis claims that sentiment analysis can help to red flag unfair reviews and, consequently, simplify this difficult process. For that, we measure the correlation between unfairness and sentiments to check how much emotions are manipulated to guide shopping tendencies.

On the one hand, having access to meaningful information is, in fact, essential during the decision-making process and, on the other hand, observation of unfair data can prevent its negative impact on businesses and consumer choices, therefore this project focuses on exploring and experimenting how to detect unfair online reviews through Sentiment Analysis using Machine Learning Techniques. The experiments focus on the discovery of unfairness in online product feedback through the process of establishing the accuracy of sentiment classification algorithms aims to detect existing unfairness towards the products.

Keywords

- Sentiment Analysis
- Machine Learning Techniques
- Fairness
- Natural Language Processing
- Unfairness measurement

Table of Contents

Tables

Acknowledgement

This experiment and research was constantly supported by the supervisory team available for discussions about new ideas, solutions to challenges and analysis of issues and problems encountered during the work through all the development process.

Dr Kelly Androutsopoulos from Middlesex University and Dr Héctor Menéndez from King's College London made available all the time their theoretical and technical expertise for which I am grateful.

Introduction

In the digital era, more traditional methods like helplines and comment boxes, which allowed customers to leave feedback about products and services, have been in large part replaced by online user reviews. Nowadays these customer reviews are publicly available on online shopping sites and are easily reachable by potential future customers.

Also, because of the improvements in data storage technology and the development of the E-Commerce environment, collections of data in form of online feedback become quicker and larger. Therefore, the advantage of having access to tangible and qualitative product information allows users to make relevant conclusions more efficiently. A high percentage of buyers choose to read feedbacks about their upcoming purchases more often, and 84% of them confirm that the reviews are crucial during the decision-making process (Bloem, 2017). The judgment about shopping is influenced by positive, neutral, and negative feedbacks of people that previously bought and had experience with specific products. For that reason, the importance of online reviews is essential because buyers trust them and therefore reviews are considered decisive for the success or failure of a business or brand.

But even though customers' behaviours are strongly affected by online reviews, they have raised many doubts about their reliability. The honesty of online reviews has been in fact questioned because of the existence of unfair or false feedbacks (Woollacott, 2017). Posted online fake positive reviews have mainly the aim to unfairly promote products and, simultaneously, false-negative reviews have the objective to discredit competitors. Misleading feedbacks become in this way dangerous, unethically influencing user decisions. We view these misleading feedbacks as being unfair to a product.

Posted reviews can be categorized as positive, neutral or negative depending on the satisfaction of each customer's experience about the product or service received. A review (positive, neutral or negative) can be true or false, and the false one could be in turn categorized as intentional or non-intentional depending on the goal of the reviewer.

Both the intentional and unintentional types of reviews are unfair to the product, being fake reviews a subset of these. Fake reviews are malicious and intentionally fraudulent, written with disruptive purposes and not aiming to truly describe the product. The definition of unfairness is therefore connected with the behaviours of reviewers that post both online not true intentional and unintentional reviews.

Aims

This project aims to address the following questions about unfair reviews:

a. Determining what fairness means in terms of online reviews. i.e. What does fairness mean with respect to reviews? When is a review considered unfair? E.g. is the unfairness towards the products?

b. Detecting unfairness in online reviews. How can we detect unfairness in reviews? Determining how unfairness can be measured?

- To answer these questions, we can use Opinion Mining by applying Machine Learning Methods, exploring the use of Sentiment Analysis Algorithms, and studying the challenges and limitations encountered during Sentiment Analysis classifications. The starting point will be the study of the results of the work described in the paper "*Unfair Reviews Detection on Amazon Reviews using Sentiment Analysis with Supervised Learning Techniques"* (Elmurngi and Gherbi, 2018).

c. When analysing product reviews what is the relation between unfairness/fairness and sentiments (positive, neutral and negative customer's feedbacks)?

The scope is to detect unfair product reviews posted by purpose or genuinely, to discover the most accurate Sentiment Analysis algorithm adopted for the experiment, and the comparison of findings with discoveries of previous research in the field.

Fairness in machine learning is a hard topic to analyse and to solve. The given customer reviews should be considered fair with respect to the context and the unfairness can be detected in different forms. Unfairness can in fact exists towards the products when the online feedback is fraudulent and posted unethically by those users acting with the aim of discrediting the competition by posting false information.

But exists different types of unfairness than fake reviews. Unfairness can be observed when, for example, the star rating does not match what is described in the review and the star rating is low, but the review is positive, and opposite, the star rating is high, but the review is negative. Unfairness can also occur when the review focuses on something outside of the company's control e.g., a review about the delivery should not affect the rating of the product. Descriptions of unrealistic consumer expectations and contradictory feedbacks are additional examples of possible unfair reviews.

The fairness of online-posted reviews could be also affected by the identification of irrelevant reviews. These reviews are neither an opinion nor any sort of feedback. These online posts could be, for example, random text or questions not providing any information about the product but additionally could exist also reviews focused only on the product's brand but not on the specific customer's purchase, and therefore being as well unfair (Bing Liu, 2012).

Because the project covers the topic of fairness in online customer reviews, the main benefit of the research will be accessing meaningful information which is essential during the decision-making process as the advantage of having access to trustworthy product information, allows users to make relevant conclusions more efficiently and in a well-planned way.

Background

This work applies Machine Learning to create the learning models for both fairness and sentiment analysis. Machine learning uses statistical methods to identify patterns in data and, at the same time, explores the study and construction of algorithms that can learn from a set of data and make predictions about them, inductively building a model based on samples known as training data. Machine Learning refers therefore to the ability of a machine to accurately understand new examples or tasks, which it has never seen before, after having experienced a set of learning data.

Tasks performed by Machine Learning can be typically classified into two main categories which are: supervised learning and unsupervised learning (Quattrone, 2021). In supervised learning, the model uses labelled data that guides its training, are given examples in the form of possible inputs and their respective desired outputs and the goal is to extract a general rule that associates the input with the correct output, while in unsupervised training it learns blindly identifying the patterns without the information of the desired outputs. There exist various applications for Machine Learning nowadays and among these Sentiment Analysis (SA), also called opinion mining, is used in many sectors: from politics to stock markets, from marketing to communication, from sports to medical and natural sciences, from social media analysis to even the evaluation of consumer preferences which is analysed in this research (Medhat, Hassan and Korashy, 2014).

To be more precise, Sentiment Analysis is an application of Natural Language Processing (NLP), which is a subfield of artificial intelligence and has the objective to make computers understand the Natural Language along with its semantics and therefore to deal with human language and understand the unstructured text, finding information from it. Using Machine Learning methods and Natural Language Processing, it is possible to retrieve information from a document and classify the textual data as positive, negative, or neutral. Sentiment Analysis studies opinions, evaluations, and thoughts that are presented in the form of text, but it is a challenging application because unlike humans that can interpret the tone of a text,

computers are not intuitive, and algorithms need to work on features generated from language data (Khan et al., 2016).

There exist many different machine learning algorithms which are used for sentiment classification, each one with pros and cons. The most used algorithms are (Larose 2014):

1. *Linear Regression* (Quattrone, 2021): LR is an algorithm based on supervised learning and is one of the most used techniques in Machine Learning. Its task is to predict the dependant variable value of *y (the class)* knowing an independent variable *x (the feature vector)*. It is a statistical method that is used for predictive analysis and is used especially for forecasting.

2. *Naïve Bayes (*Elmurngi and Gherbi, 2018)*:* Naïve Bayes classifiers are a collection of classification algorithms*.* A Bayesian model is a statistical and conditional probability model. It is based on Bayes' Theorem which is a simple mathematical formula used for calculating conditional probabilities.

3. *Support Vector Machines* (Quattrone, 2021): SVM is a supervised machine learning algorithm used especially in classification problems. It is a discriminative classifier formally defined by a separating hyperplane that differentiates two classes.

It is also important to mention that, in addition to Sentiment Analysis, in machine learning and natural language processing there is another category of algorithms that have been designed to analyse large language data, called Topic Modelling (Nikolenko et al., 2016).

Although these algorithms classify data according to a specific criterion, there are algorithms such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003) that can actually find topics within textual data. LDA is a generative probability model used to classify text in a document to a particular topic based on the words in it and producing a topic distribution. It is known to be fast and intuitive, but it also needs human interpretation and lots of fine-tuning (Moro et al., 2015).

Identifying the presence of false reviews is an arduous task and it is complex to distinguish them from real ones (Jindal et al., 2007). To solve this challenge, the

use of machine learning algorithms is a possible solution, applying Sentiment Analysis processing that deals with building systems for identifying and extracting opinions from the text. Sentiment Analysis is one of the many applications for machine learning.

However, machine learning has also its limits. For instance, these algorithms are biased because of a lack of suitable data or wrong labelling of data when the approach is to label some of the correct answers as valid when no fully satisfactory algorithm is available (Domingos, 2012). When this bias relates to societies, it is refer as a fairness bias (Mehrabi et al., 2019). Experts in this area of study investigate often when algorithms are unfair and their decisions lead to discriminatory decisions where, for example, a specific group of individuals is mistreated.

According to a recent survey on fairness in machine learning (Mehrabi et al., 2019), there has yet to be defined a unique definition of fairness (10 different definitions have been described). These different definitions can be classified as types of individual fairness, group fairness, and subgroup fairness. Unfairness is investigated especially concerning social discrimination, which reflects on documents and biases word embedding models. This can influence negatively future built models (Papakyriakopoulos, Serrano, Hegelich, and Marco, 2020).

Detection of fairness is a challenge for machine learning as it is a notion that can have many different implications and hidden meanings and depends on the choice of words and type of language used in a specific text or document but is not easy to detect. Identification of hidden meanings of data is a difficult task to be performed by computers, which have little room for subjective interpretation (Cheeks and Gaffar, 2017). This difficulty is due to the ambiguity of human language, which allows more than one unique interpretation of sentences especially because of their subjectivity and the possibility of using multiple styles when expressing and reporting an event or opinion.

Formal data and informal data are therefore the principal areas considered when it is necessary to categorize the automation of data processing. Formal data is expected and anticipated where the output depends on the specific input data without space for interpretation and, on the other side, natural languages are an

example of informal data (Cheeks and Gaffar, 2017) and this research focuses on unfairness detection in informal data as a result of personal interpretations of events formalized in online feedback about products.

Literature review

Consumer preferences are often expressed in the form of online product feedback and detection of fake and unfair reviews has been a topic of interest for the last years, although it started to be studied in 2007 focusing initially on the analysis of review spamming (Barbado, Araque and Iglesias, 2019).

With the advent of Big Data, the volume of the available information about customer satisfaction has become huge and data sources are heterogeneous, therefore also the research of hidden patterns in data becomes pivotal for commercial benefits.

For that reason, several pieces of research have been conducted analysing different datasets to understand customer needs and product sentiment in customer feedback.

In (Singla, Randhawa and Jain, 2017), authors conducted sentiment analysis classifying reviews, from a unique dataset of over 400,000 reviews about mobile phones, as positive and negative using three different classification models: Naïve Bayes, SVM, and Decision Tree. The outcome of the research, from using the Support Vector Machine algorithm, resulted to be the most precise with an accuracy of 81.75%.

Also, the authors in (Elmurngi and Gherbi, 2017), confirmed the SVM algorithm to be the best one after the measurements of their experiments. They applied sentiment analysis using machine learning techniques aiming to detect fake reviews and using two different datasets of movie reviews. In the mentioned experimental approach, data analysis was carried out applying five algorithms for sentiment classification including Naïve Bayes, Support Vector Machine, K-Nearest Neighbors, KStar, and Decision Tree. At the end of the paper, the authors proposed

to use a Statistical Analysis System and R for the detection of false reviews as future work.

The following year, the same authors in (Elmurngi and Gherbi, 2018) applied sentiment analysis in Weka using supervised learning techniques in another publication about unfair detection on Amazon reviews. This time, the authors used the four algorithms Naïve Bayes, Decision Tree, Logistic Regression and Support Vector Machine, and three Amazon datasets. In this new work, has been implemented also the LR algorithm, which has not been used in the previous research and it resulted to be the most accurate not only during the detection processes of fair and unfair positive, negative, and neutral reviews but also in the process of text classification.

In this project, is interesting to explore the use of different algorithms for applying sentiment classification with machine learning techniques for detecting unfair reviews. It is interesting to compare and evaluate, as starting point, the performance and accuracy of the two algorithms, SVM and LR, by performing experiments using the same environment and similar datasets.

In addition to experiments developed so far, recently a group of researchers from the Department of Telematic Engineering Systems of Universidad Politecnica de Madrid focused their attention on the fact that detecting false reviews based only on textual features can be challenging. Thus, in the paper *A Framework for Fake Review Detection in Online Consumer Electronics Retailers,* the proposed framework is based on both review-centric and user-centric features (Barbado, Araque and Iglesias, 2019). As conclusion, this emerging area also needs to be analysed considering nonverbal characteristics to improve the classification model. In addition to textual features and attribute selection process during text analysis, behaviours of users when writing reviews also need to be taken into consideration.

As explained in (Liu, 2012), abnormal behavioural patterns of reviewers that write potential unfair and fake reviews could be detected analysing also meta-data about the review. It is possible in fact to focus the attention on features of data like the time when the review was posted and also the host IP address and MAC address of the reviewer's computer and the user ID. The analysis of such information may lead therefore to the observation of suspicious and not genuine behaviours like,

for example, the fact that the same reviewer writes only positive or only negative posts to promote or discriminate a specific brand or business, although we did not analyse this aspect in the current experiment.

Recent research papers over the last few years are focused on unfairness using sentiment analysis algorithms. These often involve also social media data or online news and news articles but discovering bias/unfairness of the last one is assumed as a particularly elaborated process because needs to consider political, economic, and social problems (SV and Geetha, 2019).

The authors in (SV and Geetha, 2019) conducted an experiment to determine news biasedness collecting data from 20 news websites across the UK, the USA, and India with the aim of producing credible information. For the training data, they used a dataset of 3265 news sentences, and they carried out comparisons between the classifier and lightweight neural network architecture with Logistic Regression, Gradient Tree Boosting, SVM, and Naïve Bayes applying machine learning techniques. The authors designed an algorithm for estimating the biasedness of news and measured the accuracy level in comparison with the previous algorithm.

An example of data-driven analysis through text analysis has been produced also by a group of researchers in 2018, at the International Conference on Machine Learning and Applications. Authors of *Bias Evaluation of Professors' Reviews* performed an experiment to automatically discover topics of discussion. They analysed students' reviews investigating the relationship between numerical and text features and the overall rating of the review (Antonie et al., 2018). The main achievement was topic modelling and regression analysis, which have been used for an extensive analysis of the online reviews using the LDA algorithm and this approach would be interesting when thinking also about the aim to discover unfairness in online product reviews, which involves aspects that are not directly related with the product itself and where it is necessary to find multiple topics in the text review.

All the information about the mentioned papers, which are useful to better understand how the sentiment analysis using supervised learning techniques has been applied to data like reviews or text in the form of articles and social media information, are summarized in Table 1.

Title	Authors	Dataset	Algorithms used	Venue of publication	Available at:
Sentiment analysis of customer product reviews using machine learning (Singla, Randhawa and Jain, 2017)	Singla, Z., Randhawa, S. and Jain	Over 400,000 reviews about mobile phones	Naïve Bayes, SVM and Decision Tree	2017 International Conference on Intelligent Computing and Control (I2C2)	https://ieeexplore.ieee.org/document/8321910/authors#authors
Detecting Fake Reviews through Sentiment Analysis Using Machine Learning Techniques (Elmurngi and Gherbi, 2017)	Elmurngi, E. and Gherbi, A	Two different datasets of movie reviews	Naïve Bayes, Support Vector Machine, K-Nearest Neighbors, KStar and Decision Tree	DATA ANALYTICS 2017: The Sixth International Conference on Data Analytics	https://www.researchgate.net/publication/325973731_Detecting_Fake_Reviews_through_Sentiment_Analysis_Using_Machine_Learning_Techniques
Unfair Reviews Detection on Amazon Reviews using Sentiment Analysis with Supervised Learning Techniques (Elmurngi and Gherbi, 2018)	Elmurngi, E. and Gherbi	Three different datasets of products reviews	Naïve Bayes, Decision Tree, Logistic Regression and Support Vector Machine	Journal of Computer Science Volume 14 No. 5, 2018	https://www.researchgate.net/publication/325736087_Unfair_reviews_detection_on_Amazon_reviews_using_sentiment_analysis_with_supervised_learning_techniques
Determination of news biasedness using content sentiment analysis algorithm (SV and Geetha, 2019)	SV, S. and Geetha, A	Training data of 3265 news sentences	Logistic Regression, Gradient Tree Boosting, SVM and Naive Bayes	The Indonesian Journal of Electrical Engineering and Computer Science (IJEECS) Volume 16 No. 2, 2019	https://ijeecs.iaescore.com/index.php/IJEECS/article/view/18242

Bias Evaluation of Professors' Reviews (Antonie et al., 2018)	Antonie, L., Foxcroft, J., Grewa, G., Narayanan, N., Plesca, M. and Ramirez, R.	Students' online reviews	LDA	2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)	https://ieeexplore.ieee.org/document/8614121

Table 1. Summary of literature papers

Methodology

The scope of the project is to find the answer to how can we detect unfairness in reviews and how to measure what that unfairness means. Based on looking at the literature review, the paper "*Unfair Reviews Detection on Amazon Reviews using Sentiment Analysis with Supervised Learning Techniques"* (Elmurngi and Gherbi, 2018) was considered as a good starting point for the research. We have in fact taken a similar approach into account.

In the mentioned paper, for a given dataset in the classifier, the outcome can be a fair/true negative review if the document is labelled as negative and at the same time is also classified as negative. But it is counted as unfair/false negative review if instead the document is classified as positive review. Reviews are therefore considered fair negative in the testing data when the sentences are correctly predicted by the classification model like negative ones, otherwise are considered as unfair negative (Elmurngi and Gherbi, 2018). The same procedure is adopted also for neutral and positive reviews counted as fair or unfair ones.

The research is based therefore on studying and analysing Machine Learning techniques applied to Sentiment Analysis (SA). We explore and study Sentiment analysis algorithms for the classification of online product reviews like positive, neutral, and negative ones and additionally detection of unfair and fair reviews. Detection of false reviews is guided also by the definition of fairness in terms of online user feedbacks to detect and measure unfairness affecting text posted by reviewers.

We divided the experiment in two main studies that will be called in this paper respectively Study 1 and Study 2.

For both Studies we followed the same methodological approach with the difference that in Study 1 the data labeling is about the review sentiment that can be positive, negative or neutral whereas in Study 2 the data is labeled as Fair Review or Unfair review. In both studies we used the same sample of data.

Study 1	Study 2

• Amazon reviews collection
• Data cleaning
• Data preprocessing
  • Stopword removal
  • Punctuation marks removal
  • Tokenization
  • Stemming of words
• Data labeling of **review sentiment**
  • K-medoids technique
• Sentiment classification algorithm
  • Discovering of the most accurate Sentiment Analysis algorithm
• Detection processes:
  • Fair Outcomes
  • Unfair Outcomes
• Evaluation
• Conclusion
	Amazon reviews collection
	• Data cleaning
	• Data preprocessing
	  • Stopword removal
	  • Punctuation marks removal
	  • Tokenization
	  • Stemming of words
	• Data labeling **(fair/unfair review)**
	  • K-medoids technique
	• Sentiment classification algorithm:
	  • Discovering of the most accurate Sentiment Analysis algorithm
	• Detection processes:
	  • Fair Outcomes
	  • Unfair Outcomes
	• Evaluation
	• Conclusion

Table 2. Steps in the methodology of Study 1 and Study 2

We retrieved the data used for the experiment from the Amazon Customer Reviews Dataset (Amazon Customer Reviews Dataset, 2020), which is a public source of information for academic researchers in the fields of Machine Learning and

Natural Language Processing. The project does not involve the participation of people but instead uses existing anonymous records as samples of customer evaluations and opinions for analysis of these reviews.

The dataset contains the customer review texts with accompanying metadata. It is a collection of reviews written in the Amazon.com marketplace and associated metadata from 1995 until 2015 concerning product categories. Amazon or its content providers grant a limited, non-exclusive, non-transferable, non-sublicensable, revocable license to access and use the Reviews Library for the purposes of academic research, and in the used dataset, information about customers personal data are not available (i.e. name, surname, username, address).

Information about the metadata of the datasets is provided in Figure 1.

```
DATA COLUMNS:
marketplace       - 2 letter country code of the marketplace where the review was written.
customer_id       - Random identifier that can be used to aggregate reviews written by a single author.
review_id         - The unique ID of the review.
product_id        - The unique Product ID the review pertains to. In the multilingual dataset the reviews
                    for the same product in different countries can be grouped by the same product_id.
product_parent    - Random identifier that can be used to aggregate reviews for the same product.
product_title     - Title of the product.
product_category  - Broad product category that can be used to group reviews
                    (also used to group the dataset into coherent parts).
star_rating       - The 1-5 star rating of the review.
helpful_votes     - Number of helpful votes.
total_votes       - Number of total votes the review received.
vine              - Review was written as part of the Vine program.
verified_purchase - The review is on a verified purchase.
review_headline   - The title of the review.
review_body       - The review text.
review_date       - The date the review was written.
```

Figure 1. Attributes of the dataset tables

The most important metadata used for the purpose of the experiment was the ‘review_body’ as starting point for the Sentiment Analysis, it represented the review text. A useful one during the manual labelling was also the ‘start_rating’: the 1-5 star rating of the review. Was interesting in fact to compare the body of the review and the star rating to check their correspondence in terms of correlation high rating/positive review or low rating/negative review.

We collected the dataset depending on the customer review product category that can be used to group the feedbacks. The specific product category of customers reviews is about jewerly.

We divided the data into training and test datasets and after the data collection phase, we also started the cleaning and pre-processing steps.

The data cleaning process involved the detection of inaccurate and corrupted records in the dataset and duplicates. Part of the available data can be incomplete, irrelevant, or not accurate, therefore the data-cleaning phase is essential to make data consistent, otherwise, it could lead the research to false conclusions. The data can be affected, for example, by user entry errors, different data dictionary definitions, the existence of blank rows causing confusion, or errors caused by the corruption of the storage device or the data transmission.

Then once the dataset was clean and ready to use, the next step was data pre-processing, which is very important in text mining when supervised learning techniques are used (Elmurngi and Gherbi, 2018). This process focuses on:

- Normalizing words
- Removing punctuation marks
- Removing stop words
- Tokenizing sentences
- Vectorizing text

We prepared the infrastructure for the sentiment analysis and we based the research on the comparison of SA algorithms and the measurement of their accuracy. We used a confusion matrix and a classification report as a possible approach to return a representation of the statistical classification accuracy.

The project was carried out therefore as an analysis based on experimental results using existing data and analysing it through Machine Learning and Sentiment Analysis algorithms.

This research aims to find the best SA algorithm comparing labelled data as positive and equal to label 1 in the csv file, neutral equal to label 0 or negative equal to label -1 with the predictions obtained using the classifier models. There are in fact six possible outcome defined as Fair Positive Reviews, Unfair Positive Reviews, Fair Neutral Reviews, Unfair Neutral Reviews, Fair Negative Reviews and Unfair

Negative Reviews. For each classifier we calculated the number of reviews belonging to each of the six mentioned categories.

In Study 1 we followed some guidelines while labelling each review as positive, negative or neutral.

Data labeled as:

- **Positive** :
    - Review has a positive tone
    - It expresses overall satisfaction with a product
    - Includes recommendations for others
    - Highlights the strengths and positive aspects of a product
- **Negative**:
    - Review has a negative tone
    - It expresses dissatisfaction of the customer with a product and a negative experience
    - Poor product quality which doesn't meet a customer expectations
- **Neutral:**
    - Review has a neutral tone
    - The reviewer expresses neither positive nor negative feelings

K-medoids technique

To achieve this, we first labeled datapoints manually as positive, neutral and negative to allow supervised training to find the pattern in data. To do so we used the K-medoids technique which is a partition clustering algorithm related to the K-means algorithm. It expects as input a set of n objects and a number k which determines how many clusters are wanted as output where a medoid can be defined as an element of a cluster whose mean dissimilarity to all objects in the cluster is minimal, so it will be the most central point of a given set of points.

```
kmedoids = KMedoids (n_clusters=100, init= 'build').fit(X)
```

Figure 5. Line of code setting the number of medoids=100

Once we labeled k medoids, each one being part of a separate cluster, was possible then to label the rest of the data knowing each datapoint to which cluster belongs to and therefore we labeled all the reviews either as positive, neutral or negative sentiment (Arora et al., 2016).

We used the clustering technique to explore data to identify patterns especially because of the large number of reviews to label and we have chosen K-Medoids over K-Mens because it is considered to be better when measuring the execution time. It also brings to minimum the sum of heterogeneity of the data objects analysed (Arora et al., 2016).

In the second part of the research we used the clustering technique not only to label the reviews with the correspondent sentiment but we did separately the labelling also as unfair review or fair review following the interpretation of fairness and unfairness described in the section Aims. A product review can be considered unfair when for example there are described unrealistic consumer expectations, focus on something outside the company's control, when the review is fake, it does not provide any useful information about the product, or the star rating does not reflect the sentiment of the review text.

In Study 2 data was then labeled as;

Unfair :

- Unrealistic consumer expectation
- Contradictory feedbacks
- Questions and random text not without information about the product
- Star rating low and review positive, and opposite, star rating high and review negative
- Focus on something outside the company's control

- **Fair**:
  - The star rating reflects the sentiment of the review
  - Focus on product and anything the company has control over

Using therefore the same test data we retrieved predictions not only about sentiments on datapoints but also about the interpretation of fairness using different classifiers models.

The last research question can be considered as: When analysing product reviews there exists a correlation between unfairness and sentiments (positive, neutral and negative customer's feedbacks)?

To answer that question the implementation of the Pearson coefficient correlation used in Python can be a possible solution to measure the linear association between variables, in this case positive reviews and unfairness and negative reviews and fairness as we expect that there is not a 1-to-1 correlation between positive and fair reviews, and negative and unfair reviews. Is possible that a positive review is unfair when for example describes positively another product or contrary a negative review can be fair when describe a product that is not good enough.

With the Pearson correlation coefficient also called Pearson Product-Moment Correlation Coefficient is measured the linear associations between variables and the results can vary between +1 and -1 (Saeed, 2022). When the value obtained is close to +1 that means that exists a strong positive correlation between variables but on the other side if it is close to -1 the correlation is negatively strong. Value +1 represents a total positive correlation but -1 a total negative correlation. Value equal to zero represents that there is not correlation at all between variables and finally values around +0.6 and -0.6 refers respectively to a moderate positive and moderate negative correlation (Saeed, 2022).

This research focuses therefore on a possible methodology to measure unfairness and the evaluation of the research outcomes, giving the opportunity of improving not only hard skills but also methodological problem solving and critical thinking.

Implementation

To achieve the goal of the experiment and to answer to the research's questions we used Machine Learning Methods exploring the use of Sentiment Analysis Algorithms.

In order to do that, we implemented the experiment using Python programming language and using the web-based interactive development environment Jupyter notebook.

For the analysis of the data and for its manipulation we used the open-source Panda Library for Python language, which is known to be a simple, rapid and commonly used tool for machine learning experiments when involving data science. This is in fact a library which allows to work easily and efficiently with big amounts of data. But other libraries used during the research development are also Numpy, Nltk and Sklearn.

Numpy has been created in 2005 by Travis Oliphant and is especially used when working with multidimensional and large arrays and it is a Python library offering mathematical functions for data science where arrays are widely applied but when working with Natural Language Porcessing (NLP) Nltk is considered to be the most popular lilbrary with a large choice of algorithms and Sklearn is an open-source machine learning library for predictive analysis of data.

Applying Machine Learning Methods, the algorithms that we used to do predictions on the given dataset about sentiments and fairness are:

1. Logistic Regression
2. GaussianNB
3. Support Vector Machines
4. Decision Tree Classifier
5. Linear Discriminant Analysis
6. K Neighbors Classifier
7. Random Forest Classifier
8. Gradient Boosting Classifier
9. Ada Boost Classifier
10. NNet Classifier

For each algorithm we produced predictions, accuracy estimation, confusion matrix and classification report with statistical information about the data like precision, recall, f1-score and support, macro average and weighted average. Additionally, we calculated the sum of the fair positive reviews, fair negative reviews, fair neutral reviews and then unfair positive, negative and neutral by comparing the result of the labelling done manually and the calculated predictions by the above-mentioned algorithms.

In both the studies, Study 1 and Study 2, setting up the Machine Learning classifiers for Sentiment Analysis, we used the libraries in table 3.

ML algorithms for Sentiment Analysis (Study 1) and Fairness (Study 2)	Imported libraries
Logistic Regression	`from sklearn.linear_model import LogisticRegression`
GaussianNB	`from sklearn.naive_bayes import GaussianNB`
Support Vector Machines	`from sklearn.svm import SVC`
Decision Tree Classifier	`from sklearn.tree import DecisionTreeClassifier`
Linear Discriminant Analysis	`from sklearn.discriminant_analysis import LinearDiscriminantAnalysis`
K Neighbors Classifier	`from sklearn.neighbors import KNeighborsClassifier`
Random Forest Classifier	`from sklearn.ensemble import RandomForestClassifier`
Gradient Boosting Classifier	`from sklearn.ensemble import GradientBoostingClassifier`
Ada Boost Classifier	`from sklearn.ensemble import AdaBoostClassifier`
NNet Classifier	`from sklearn.neural_network import MLPClassifier`

Table 3. Machine Learning algorithms used for Sentiment Analysis and Firness in Study1 and Study 2

Dataset

For the achievement of the experimental goals we produced the implementation in Python programming language and the first step during the implementation of the work was the retrieving of data. From the original Amazon Customer Reviews Dataset about Jewelry we decided to use a sample of 5000 data points saved in the format of a '.csv' file when the original amount of data was equal to 1,752,932

reviews. We explored the data and checked its data format leaving all the metadata as in their original version without the deletion of any of them. But we deleted 3 reviews from the 5000 dataset because of the presence of empty fields.

```
df = df[df.astype(str)['words'] != '[]']
```

Fig.15 Line of code deleting empty fields in the column 'words'

Finally, the training dataset was of 4997 customer feedbacks and for testing data we used 999 datapoints.

When starting the experiment, we used a very limited amount of reviews when proceeding with the testing, which was equal to only 100 reviews, but going further with the work also the dataset was increased adding 900 reviews to the previous 100 with the decision of having 1000 reviews in the test phase. As final result, the testing dataset has 999 reviews because was necessary to remove one as was a duplicate, checking the IDs Review.

For training and testing dataset we selected random samples of reviews making sure that the two datasets are different and that data in the testing dataset was not included in the training dataset when Training Dataset > Test Dataset.

In the tables below are specified the amounts of positive, neutral and negative reviews in Study 1 and Fair reviews and Unfair reviews in Study 2, both in the testing dataset with a total of 999 reviews.

Label	**N. of reviews**
Positive	766
Neutral	82
Negative	151

Table 9. Results of manual labeling of testing dataset in Study 1

Label	**N. of reviews**
Fair	925
Unfair	74

Table 10. Results of manual labeling of testing dataset in Study 2

In the next figure is showed the distribution of manual labels in the training dataset in Study 1 and Study 2. When labeling the training dataset we started with the Medoids and then we were able to label the rest of the data.

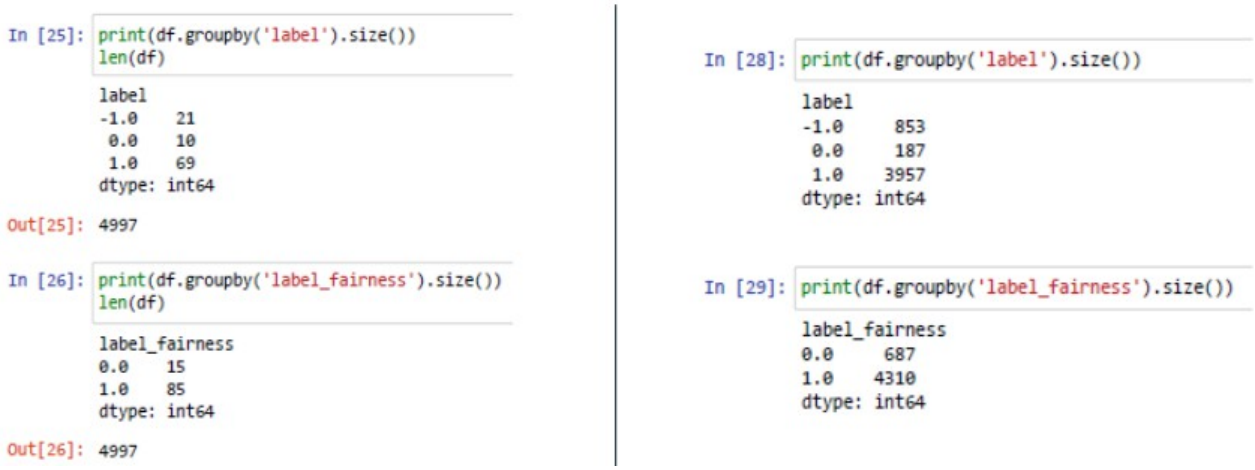

Figure 6. Manual labels of medoids and entire training datset in Study 1 and Study 2

Data-preprocessing

Using the Nltk library we found out the possible stop words present in the text of the reviews 'review_body'.

Therefore, we added in the '.csv' file a new column called 'words'. For each row (data point) we saved the words of each review removing punctuation marks and stop words.

```
def identify_tokens(row):
    review = row["review_body"]
    tokens = nltk.word_tokenize(review)
    # taken only words (not punctuation)
    token_words = [w for w in tokens if w.isalpha() and w not in stopwords]
    return token_words

df["words"] = df.apply(identify_tokens, axis=1)
```

Figure 7. Code removing punctuation marks and stop words

```
from nltk.corpus import stopwords

stopwords= stopwords.words("english")
print(stopwords)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yo
urs', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's",
'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "tha
t'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do',
'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'b
y', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 't
o', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'ther
e', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no',
'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'sho
uld', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn',
"didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn',
"mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'were
n', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

Figure 8. Examples of stop words

Additionally, to what was mentioned we also had to delete the rows where the list of cleaned words was empty as '[]' resulting this in the final amount of datapoints of 4997 reviews.

Making still use of Nltk Python library we retrieved the column 'stemmed_words' with input words, This because in the English language, but not only, depending by the context analysed, the words can be expressed in different forms. Stemming is in fact a method for the normalization of the words which allows to avoid interpreting a single word expressed in multiple forms with more than one explanation and meaning. The word inputs necessary for the research are therefore the ones truncated using only stem words. An example of Stemming can be the words "working", "worked" and "works" where the stem word will be only "work" reducing noises and possible misunderstanding about the interpretation. But using the Porter's Stemmer Algorithm there are also limitations because the stem input words can be not real, although the advantages are more than the limitation because the error rate is very low and the outputs are the most precise when compared to other algorithms like for example Lovin's Stemmer, Krovetz Stemmer, Xerox Stemmer, Dawson's Stemmer, or Snowball Stemmer.

```
from nltk.stem import PorterStemmer
stemming = PorterStemmer()

def stem_list(row):
    my_list = row['words']
    stemmed_list = [stemming.stem(word) for word in my_list]
    return (stemmed_list)

df['stemmed_words'] = df.apply(stem_list, axis=1)
```

Figure 9. Porter's Stemmer Algorithm

The following step for the preparation of the saved data is the introduction of the TfidfVectorizer using the Sklearn Python library. The Term Frequency - Inverse Document Frequency (TF-IDF) is useful to detect how relevant are in a specific document the words applying calculation of the score when retrieving information. Term Frequency is measured when a word is counted more times than other terms and therefore is considered to be more important compared to other, but on the other side, for the Inverse Document Frequency if a word appears very frequently in many documents it can be considered not very important.

During the implementation of the clustering algorithm K-Medoid, tf-idf matrix was also used as data to create the 100 central medoids splitting the 4997 datapoints into 100 clusters to allow easier and more efficient manual labelling. Once knowing the exact sentiment of each medoid we implemented a function to label the rest of the data. We labelled each datapoint positive, neutral or negative depending on the cluster of appartenance and checking the sentiment of the central medoid.

We adopted the same technique when labelling manually fair and unfair review but, in this case, instead of having three classes like in the sentiment analysis there are only two classes fair and unfair.

Once achieved the goal about finding the accuracy of SA algorithms and having made the predictions, the last part of the implementation involved the research about an existing correlation between sentiments and fairness. To answer the research question, we used the Support Vector Machines algorithm and we predicted the probabilities for fairness and unfairness labels and separately for sentiments of the reviews (positive, neutral and negative). In case of fairness and unfairness we

obtained for each datapoint two values which sum is 1 and for sentiments has been obtained three values as well having as sum 1.

When we calculated the predictions for probabilities, they have been grouped in five different lists. We created a list with all the positive probability predictions, a list for all the negative ones and a list for the neutral ones. Similarly, we created a list with the probability predictions for unfairness and fairness separately using the indexing to access the needed data in the multidimensional arrays were data was stored.

Finally, we found the Pearson correlation between the sentiment probability prediction and the one for fairness or unfairness.

Result Analysis and evaluation

The experiment was focused on studying and analysing Machine Learning techniques applied to Sentiment Analysis (SA) using ten different algorithms.

After the predictions made to find out if a review is classified as positive, neutral or negative, as result turned out that the most accurate algorithm is the Support Vectors Machines (SVM) with an accuracy of 0.7817817817817818 and with the following accuracy values for the other examined algorithms:

- Logistic Regression Accuracy: 0.7537537537537538
- GaussianNB Accuracy: 0.35035035035035034
- Decision Tree Classifier Accuracy: 0.7337337337337337
- Linear Discriminant Analysis Accuracy: 0.6316316316316316
- K Neighbors Classifier Accuracy: 0.7527527527527528
- Random Forest Classifier Accuracy: 0.7667667667667668
- Gradient Boosting Classifier Accuracy: 0.7767767767767768
- Ada Boost Classifier Accuracy: 0.5765765765765766
- NNet Classifier Accuracy: 0.7547547547547547

GaussianNB Algorithm results to be the most inaccurate one with the accuracy value of only 0.35035035035035034 where we retrieved the accuracy with the below calculation:

Accuracy = FPR+FNR+FNeR/ tot. Reviews

After the implementation of Sentiment Analysis algorithms for retrieving unfair and fair reviews we observed that with the most accurate SVM algorithm are counted the following reviews:

- Number of Fair Positive Reviews : 742
- Number of Unfair Positive Reviews : 183
- Number of Fair Negative Reviews : 38
- Number of Unfair Negative Reviews : 27
- Number of Fair Neutral Reviews : 1
- Number of Unfair Neutral Reviews : 8

The sum of the above reviews is equal to 999 which is the testing dataset used to perform the predictions.

```
[[ 38   2 111]
 [  9   1  72]
 [ 18   6 742]]
```

Figure 10. Example of confusion-matrix for the most accurate SVM algorithm in Study 1

We applied the calculation of the Number of Fair Positive Reviews, Unfair Positive Reviews, Fair Negative Reviews, Unfair Negative Reviews, Fair Neutral Reviews and Unfair Neutral Reviews using all the ten algorithms involved in the research.

When the review is labelled as positive equal to 1 in the csv file and the prediction is positive too, then it is counted as fair positive one and it is correctly predicted otherwise if the prediction is negative or neutral then the review is counted as unfair positive one. The same approach is used for negative reviews in testing data and if they are predicted as negative then can be considered fair but if not, they are considered as unfair negative if predicted differently.

In the testing data of 999 reviews we did the labelling manually also for fairness and unfairness separately for each review text and then we did predictions for fairness labelled as 1 and unfairness labelled as 0. In terms of accuracy these are the results for each algorithm (same algorithms used for sentiment analysis):

- Logistic Regression Accuracy: 0.8578578578578578
- GaussianNB Accuracy: 0.4914914914914915
- Support Vector Machines Accuracy: 0.9029029029029029
- Decision Tree Classifier Accuracy: 0.8448448448448449
- Linear Discriminant Analysis Accuracy: 0.7277277277277278
- K Neighbors Classifier Accuracy: 0.8478478478478478
- Random Forest Classifier Accuracy: 0.8758758758758759
- Gradient Boosting Classifier Accuracy: 0.8998998998998999
- Ada Boost Classifier Accuracy: 0.8948948948948949
- NNet Classifier Accuracy: 0.8658658658658659

Also, in this case the SVM algorithm is the most accurate with the accuracy value equal to 90%. Although for both predictions of sentiments and for fairness/unfarness it is the most accurate, the value visibly changes, being in fact lower for sentiment analysis predictions and equal to 78%in contraposition to the 90%.

In the below tables are presented the results of Study 1 and Study 2.

ML algorithm for SA	**Accuracy in Study 1**	**Accuracy in Study 2**
Logistic Regression	0.75	0.86
GaussianNB	0.35	0.49
Support Vector Machines	0.78	0.9
Decision Tree Classifier	0.73	0.84
Linear Discriminant Analysis	0.63	0.73
K Neighbors Classifier	0.75	0.85
Random Forest Classifier	0.77	0.88
Gradient Boosting Classifier	0.78	0.9
Ada Boost Classifier	0.58	0.89

NNet Classifier	0.75	0.87

Table 4. ML accuracy results in Study 1 and Study 2

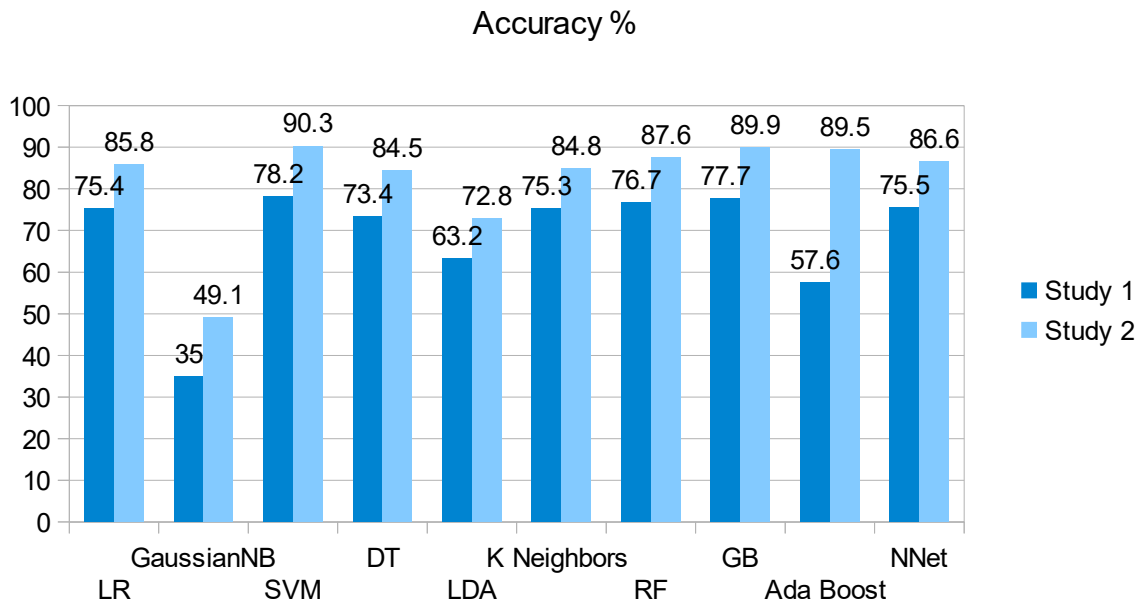

Figure 2. Comparison of accuracy of different classifiers in Study1 and Study2

Algorithm	**True Positive Reviews**	**False Positive Reviews**	**True Negative Reviews**	**False Negative Reviews**	**True Neutral Reviews**	**False Neutral Reviews**
Logistic Regression	689	129	66	99	5	18
GaussianNB	283	105	37	163	30	381
Support Vector Machines	742	183	38	27	1	8
Decision Tree Classifier	663	127	65	110	5	29
Linear Discriminant Analysis	556	107	63	176	12	85
K Neighbors Classifier	699	161	51	81	2	5

Random Forest Classifier	709	146	56	78	1	9
Gradient Boosting Classifier	725	161	46	35	2	27
Ada Boost Classifier	499	115	73	292	2	18
NNet Classifier	691	144	61	86	1	16

Table 5. Predictions on testing dataset in Study 1

Algorithm	**True Positive Reviews %**	**False Positive Reviews %**	**True Negative Reviews %**	**False Negative Reviews %**	**True Neutral Reviews %**	**False Neutral Reviews %**
Logistic Regression	68.9	12.9	6.6	9.9	0.5	1.8
GaussianNB	28.3	10.5	3.7	16.3	3	38.1
Support Vector Machines	74.2	18.3	3.8	2.7	0.1	0.8
Decision Tree Classifier	66.3	12.7	6.5	11	0.5	2.9
Linear Discriminant Analysis	55.6	10.7	6.3	17.6	1.2	8.5
K Neighbors Classifier	69.9	16.1	5.1	8.1	0.2	0.5
Random Forest Classifier	70.9	14.6	5.6	7.8	0.1	0.9
Gradient Boosting Classifier	72.5	16.1	4.6	3.5	0.2	2.7
Ada Boost Classifier	49.9	11.5	7.3	29.2	0.2	1.8
NNet Classifier	69.1	14.4	6.1	8.6	0.1	1.6

Table 6. Results of ML algorithms in Study 1 as %

Algorithm	True Fair Reviews	False Fair Reviews	True Unfair Reviews	False Unfair Reviews
Logistic Regression	830	47	27	95
GaussianNB	451	34	40	474
Support Vector Machines	897	69	5	28
Decision Tree Classifier	817	47	27	108
Linear Discriminant Analysis	700	47	27	225
K Neighbors Classifier	837	64	10	88
Random Forest Classifier	849	48	26	76
Gradient Boosting Classifier	894	69	5	31
Ada Boost Classifier	888	68	6	37
NNet Classifier	845	54	20	

Table 7. Predictions on testing dataset in Study 2

Algorithm	True Fair Reviews %	False Fair Reviews%	True Unfair Reviews %	False Unfair Reviews %
Logistic Regression	83	4.7	2.7	9.5
GaussianNB	45.1	3.4	4	47.4
Support Vector Machines	89.7	6.9	0.5	2.8
Decision Tree Classifier	81.7	4.7	2.7	10.8
Linear Discriminant Analysis	70	4.7	2.7	22.5
K Neighbors Classifier	83.7	6.4	1	8.8
Random Forest Classifier	84.9	4.8	2.6	7.6
Gradient Boosting Classifier	89.4	6.9	0.5	3.1
Ada Boost	88.8	6.8	0.6	3.7

Classifier				
NNet Classifier	84.5	5.4	2	8

Table 8. Results of ML algorithms in Study 2 as %

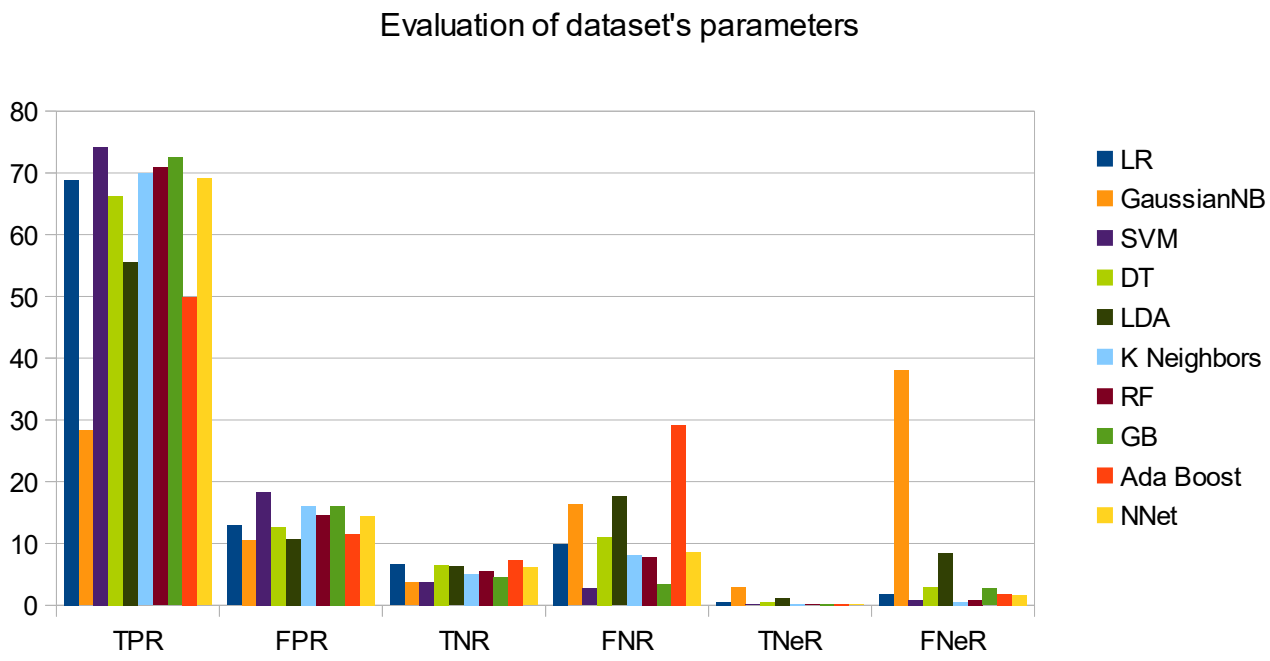

Figure 3. Graph about the evaluation of dataset parameters in Study 1

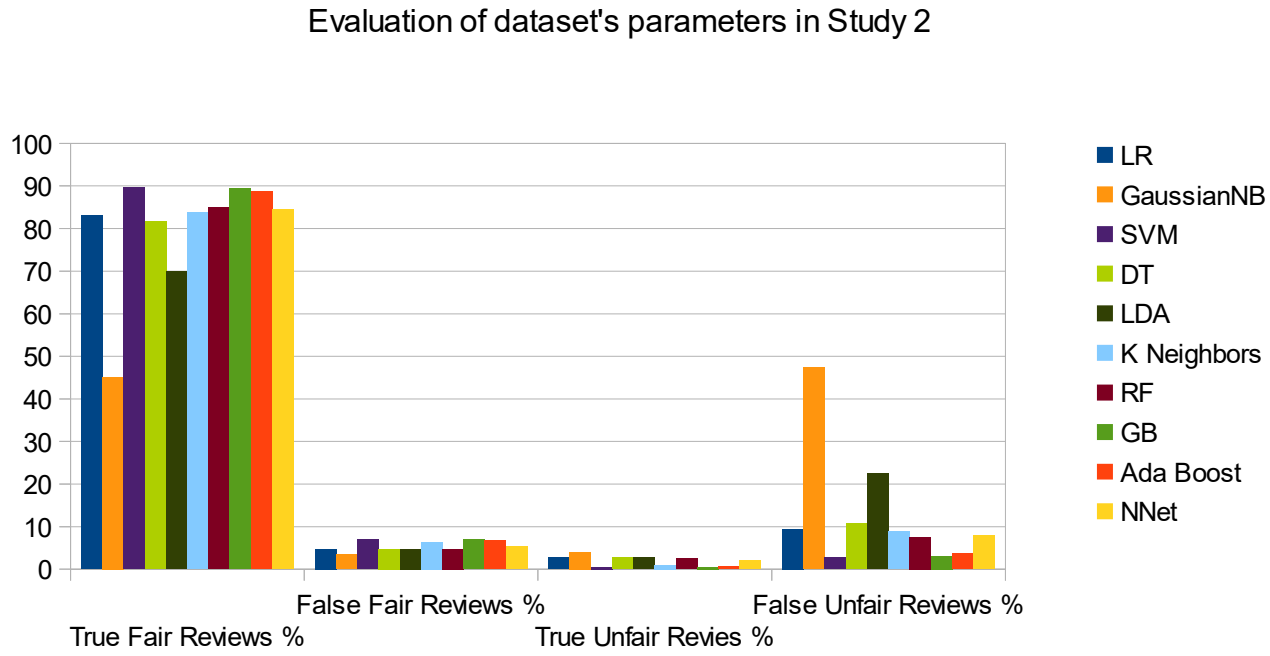

Figure 4. Graph about the evaluation of dataset parameters in Study 2

In the above graph, for clarity, TPR is the abbreviation for True Positive Reviews, FPR for False Positive Reviews, TNR for True Negative Reviews, FNR for False Negative Reviews, TNeR for True Neutral Reviews and last FNeR is the abbreviation for False Neutral Reviews.

The results of the last part of the experiment are about the correlation between fairness/unfairness and sentiments.

Predicting first the probabilities for fairness labels and separately for sentiment labels and using then the Pearson correlation, we then calculated the coefficient between:

- the probability predictions for unfairness and probability predictions of the positive sentiment
- the probability predictions for fairness and probability predictions of the negative sentiment
- the probability predictions for fairness and probability predictions of the neutral sentiment

One method to find the Pearson Correlation coefficient is the use of *corrcoef()* from NumPy.

The results of this part of the experiment are the following:

- Pearsons correlation coefficient between positive sentiment and unfairness: -0.487
- Pearsons correlation coefficient between neutral sentiment and fairness: 0.077
- Pearsons correlation coefficient between negative sentiment and fairness: -0.558

```
[[ 1.         -0.48700918]
 [-0.48700918  1.        ]]
```

Figure 12. Example of the output correlation matrix between positive sentiment and unfairness

```
[[ 1.         -0.55798609]
 [-0.55798609  1.        ]]
```

Figure 13. Example of the output correlation matrix between negative sentiment and fairness

Measuring the correlation coefficient we produced as output also the correlation matrix for a visualization of the results where the correlation coefficient result is equal to one when is calculated between a variable and itself.

Interpreting the results about the Pearson correlation coefficient it is visible that between positive sentiments and unfairness there is a moderate negative correlation. Also, between negative sentiments and fairness the correlation is moderate negative and this because when the value result is around -0.6 it is not considered as a strong correlation. But between neutral sentiments and fairness there is no correlation because the value is close to zero being equal to 0.077(Saeed, 2022).

The strength of a linear association between two variables is calculated using the Pearson correlation coefficient which checks the association between two variables. When the result is a value bigger than 0 then the association between variables is positive, so when one value increases then also the second one do the same. On the other case, when the value is less then 0 that is meaning that there is a negative correaltion and when the value of one variable increases then the other one decreases.

This research was based on a similar approach used in the work described in the paper "*Unfair Reviews Detection on Amazon Reviews using Sentiment Analysis with Supervised Learning Techniques"* (Elmurngi and Gherbi, 2018). The amount of data used in the two experiments varies significantly but the idea to find the answers to the research questions is similar. The results of this experiment and the ones of the mentioned paper as starting point are not the same.

Researchers Elmurngi and Gherbi from Department of Software and IT Engineering from the École de Technologie Supérieure in Montreal showed in their

experiment results that the Logistic Regression algorithm (LR) is the best one compared to Support Vector Machine (SVM), Naïve Bayes (NB) and Decision Tree (DT-J48) for sentiment classification because in the described work had the best accuracy when tested on three different datasets with outcome equal to 81% of accuracy for the first dataset, 80% for the second dataset and 60% for the third dataset.

Differently, in this research the highest accuracy is the one of Support Vector Machines (SVM) algorithm.

In the paper "*Unfair Reviews Detection on Amazon Reviews using Sentiment Analysis with Supervised Learning Techniques"* (Elmurngi and Gherbi, 2018) has not been studied and calculated any correlation (Pearson coefficient correlation) between sentiment analysis results and fairness/unfarness as intended in this work proceeding with Study 1 and Study 2, therefore is not possible to do a comparison between the two research considering the linear associations between variables. But on the other hand was possible to check the difference in results about the best Machine Learning technique applied to sentiment analysis for unfair reviews detection on Amazon Reviews already mentioned above.

The possible reasons the results of our experiment differ from the ones presented in previous work (Elmurngi and Gherbi, 2018) could be attribuited to several factors.

Reaserchers Elmurngi and Gherbi in their work used three different datasets:

- clothing, shoes and jewelry dataset
- baby reviews dataset
- pet supplies dataset

In our research we used only one dataset about jewelry also retrieved from Amazon but without being sure that it is the same used by Elmurngi and Gherbi. The datasets of the previous work is infact not available for inspection and even if the category of our reviews is the same of one of the three used in the paper "*Unfair Reviews Detection on Amazon Reviews using Sentiment Analysis with Supervised Learning Techniques"* there is high probability that the sample is not the

same. We chose the sample randomly using the amount of 4997 reviews for training dataset and a separate dataset of 999 reviews for testing.

In their paper the reaserchers not only used three different datasets but chose also a notable bigger amount of reviews than we did in our experiment.

In the next table is showed the number of reviews used for each dataset category:

Table 1: Number of reviews and ratings of dataset

Dataset	Reviews	Ratings
Clothing, Shoes and Jewelry	278,677	278,677 (1to5 scores)
Baby	160,792	160,792 (1to5 scores)
Pet Supplies	157,836	157,836 (1to5 scores)

Figure 14. Number of reviews and ratings of dataset in the paper "*Unfair Reviews Detection on Amazon Reviews using Sentiment Analysis with Supervised Learning Techniques" (Elmurngi and Gherbi, 2018)*

Additional possible reason of the difference in the results in the two works is the fact that the authors used Weka 3.8 tool and we implemented the experiment using Python programming using Jupyter notebook. But the main impact on the outcomes is given by the manual labeling process and we do not have any information on how was done by the authors Elmurngi and Gherbi the manual labeling of te reviews as positive, negative and neutral while we used the K-Medoids technique.

In the author's paper are not mentioned the guidelines used during the labeling of the reviews and these can be different from the ones that we used in our research. Moreover human mistakes can have also an impact on the labeling and then on the results of the ML predictions.

In our research, deciding which ML algorithm for sentiment analysis is the best, we focused on the analysis of the accuracy of each one choosing the algorithm with the highest value but on the contrary the researchers, in the paper used as starting point, to evaluate the performance of the classifiers, implemented accuracy and also precision and recall as a performance measure. In our implementation there are outputs about the classification report where there are results about precision, recall and f1-score but we did not analysed deeply these parameters and focused only on accuracy of the classfiers.

	precision	recall	f1-score	support
-1	0.58	0.25	0.35	151
0	0.11	0.01	0.02	82
1	0.80	0.97	0.88	766
accuracy			0.78	999
macro avg	0.50	0.41	0.42	999
weighted avg	0.71	0.78	0.73	999

Figure 11. Example of classification report output for the most accurate SVM algorithm in Study 1

After further research and studies we understood that only accuracy is not always a good performance measure. The performance of prediction models to evaluate sentiment classification can be determined using different evaluation measures without excluding each other, to be more accurate.

Like described by the authors Rebecca Moussa and Federica Sarra in the paper "*On the use of evaluation measures for defect prediction studies"* (Moussa and Sarro, 2022), when doing predictions the results can be often biased by the presence of data imbalance which is one of the main limitations of accuracy. If the frequency of one class is higher then the others, then also the result of the accuracy could be higher also if the model does not predict the class with the lower frequency correctly. This could be the case when the number of positive reviews is much higher then the number of labeled reviews as negatiove ones.

Authors analysing 111 works published in 2020-2022 discovered that the majority of them, more than half, for the performance measure of the prediction classifiers do not use more than one evaluation measure which, in presence of imbalanced data, does not give realistic results about the model performance.

In general, when using only one evaluation model at a time, a prediction classifier can be considered the best only until will not be used a different one because the results will change and the classifier instead of being considered the best one, like it was until then, can become even the worst one according to the new results detected with the other evaluation model. Accuracy can therefore be an

insufficient evaluation model when deciding which is the best prediction classifier because of the weakness when is analysed imbalanced data.

Limitations

Knowing the results of the experiment and research conducted it is necessary for us also to include possible work limitations as existing threats to its validity. This especially because of the manual labelling process of the data.

Once we set the criteria for the labelling, it could still be affected by bias especially when we did it manually. Human errors are common during this kind of practice and can produce biased results. Additionally different researchers could interpret differently the meaning of a text review and therefore decide to label it in different way both regarding the sentiment of the text which not necessary for everyone can be always positive, negative, or neutral and also the regarding the fairness of a review. A researcher could consider the text fair but another researcher working on the same project could label it as unfair.

When data is labelled manually it can be partially or totally disrupted by inaccuracy, part of the data could even be labelled and part of it not having this a negative impact on the results.

Another limitation can be the amount used during the experiment. Bigger is the dataset and more meaningful are the outcomes of their study.

Conclusion and future work

Detecting fairness and unfairness in online product reviews is a very challenging topic but at the same time important and interesting to solve, especially because of the growing amount of data available about different products on the market which have impact on the behaviour and choice of customers. Buyers more often do online research about the opinion of others and look for feedbacks before pursuing a product. Information available have an impact on potential consumers

therefore, the decision-making process is influenced by fair and unfair reviews available online.

In this work we conducted a research and experiment using Machine Learning Methods, exploring the use of Sentiment Analysis Algorithms, and studying the challenges and limitations encountered during Sentiment Analysis classifications with the aim to find if exists a correlation between fairness/unfairness and sentiments. We also implemented a method to count the amount of Fair and Unfair reviews in a testing dataset considering the result of the manual labelling of data and the prediction by multiple classifiers on the same data.

We used ten different algorithms for Sentiment Analysis: Logistic Regression, GaussianNB, Support Vector Machines, Decision Tree Classifier, Linear Discriminant Analysis, K Neighbors Classifier, Random Forest Classifier, Gradient Boosting Classifier, Ada Boost Classifier and NNet Classifier.

During the work we also explored the K-medoids clustering algorithm used during the labelling phase of data to make it easier and more efficient. It was one of the possible challenges encountered during the experiment as information available about clustering algorithms are mainly focused on the algorithm K- Means.

For future work to improve the experiment would be useful use a larger amount of data. Additionally we could use Topic Modelling, which is a type of statistical model for discovering the abstract "topics" that occur in a collection of documents for this goal. In this way, it would be possible to identify if some aspects not directly related to the product are included in the review and can unfairly influence the readers.

References

Antonie, L., Foxcroft, J., Grewa, G., Narayanan, N., Plesca, M. and Ramirez, R., 2018. Bias Evaluation of Professors' Reviews. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). [online] IEEE. Available at: <https://ieeexplore.ieee.org/document/8614121> [Accessed 20 February 2021].

Barbado, R., Araque, O. and Iglesias, C., 2019. A framework for fake review detection in online consumer electronics retailers. *Information Processing & Management*, 56(4), pp.1234-1244.

Bloem, C., 2017. 84 Percent Of People Trust Online Reviews As Much As Friends. Here's How To Manage What They See. [online] Inc.com. Available at: <https://www.inc.com/craig-bloem/84-percent-of-people-trust-online-reviews-as-much-.html#:~:text=Research%20shows%20that%2091%20percent,one%20and%20six%20online%20reviews.)> [Accessed 8 November 2020].

Cheeks, L. and Gaffar, A., 2017. A social influence model for exploring double subjectivity through news frames in online news. In: 2017 Intelligent Systems Conference (IntelliSys). [online] IEEE. Available at: <https://ieeexplore.ieee.org/document/8324285> [Accessed 20 February 2021].

Chouldechova, Alexandra & Roth, Aaron. (2018). *The Frontiers of Fairness in Machine Learning.*

Elmurngi, E. and Gherbi, A., 2017. Detecting Fake Reviews through Sentiment Analysis Using Machine Learning Techniques. In: *DATA ANALYTICS 2017: The Sixth International Conference on Data Analytics*. [online] Available at: <https://www.researchgate.net/publication/325973731_Detecting_Fake_Reviews_through_Sentiment_Analysis_Using_Machine_Learning_Techniques> [Accessed 10 October 2020].

Elmurngi, E. and Gherbi, A., 2018. Unfair Reviews Detection on Amazon Reviews using Sentiment Analysis with Supervised Learning Techniques. *Journal of Computer Science*, [online] 14(5), pp.714-726. Available at: <https://www.researchgate.net/publication/325736087_Unfair_reviews_detection_on_Amazon_reviews_using_sentiment_analysis_with_supervised_learning_techniques> [Accessed 10 October 2020].

Liu, B., 2012. Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*, 5(1), pp.1-167.

Medhat, W., Hassan, A. and Korashy, H., 2014. Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal, 5(4), pp.1093-1113.

Mehrabi, Ninareh & Morstatter, Fred & Saxena, Nripsuta & Lerman, Kristina & Galstyan, Aram. (2019). *A Survey on Bias and Fairness in Machine Learning*.

Quattrone, G., 2021. *Supervised learning.*

Quattrone, G., 2021. *Natural Language Processing in R (Part 1).*

Quattrone, G., 2021. *Topic modelling.*

S3.amazonaws.com. 2020. Amazon Customer Reviews Dataset. [online] Available at: <https://s3.amazonaws.com/amazon-reviews-pds/readme.html> [Accessed 5 November 2020].

Singla, Z., Randhawa, S. and Jain, S., 2017. Sentiment analysis of customer product reviews using machine learning. In: *2017 International Conference on Intelligent Computing and Control (I2C2)*. [online] IEEE. Available at: <https://ieeexplore.ieee.org/document/8321910/authors#authors> [Accessed 10 November 2020].

SV, S. and Geetha, A., 2019. Determination of news biasedness using content sentiment analysis algorithm. Indonesian Journal of Electrical Engineering and Computer Science, 16(2), pp.882-889.

Woollacott, E., 2017. Amazon's Fake Review Problem Is Now Worse Than Ever, Study Suggests. Forbes, [online] Available at: <https://www.forbes.com/sites/emmawoollacott/2017/09/09/exclusive-amazons-fake-review-problem-is-now-worse-than-ever/?sh=979a63c7c0f1> [Accessed 1 November 2020].

Nikolenko, S.I., Koltcov, S. and Koltsova, O. (2016) “Topic modelling for Qualitative Studies,” *Journal of Information Science*, 43(1), pp. 88–102. Available at: https://doi.org/10.1177/0165551515617393.

Khan, M.T. *et al.* (2016) “Sentiment analysis and the complex natural language,” *Complex Adaptive Systems Modeling*, 4(1). Available at: https://doi.org/10.1186/s40294-016-0016-9.

Larose, D.T., Larose, C.D.: Discovering knowledge in data: an introduction to data mining, vol. 4. John Wiley & Sons (2014).

Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research 3(Jan), 993–1022 (2003).

Moro, S., Cortez, P., Rita, P.: Business intelligence in banking: A literature analysis from 2002 to 2013 using text mining and latent dirichlet allocation. Expert Systems with Applications 42(3), 1314–1324 (2015).

Jindal, N., Liu, B.: Review spam detection. In: Proceedings of the 16th internaional conference on World Wide Web. pp. 1189–1190 (2007).

Domingos, P.: A few useful things to know about machine learning. Communications of the ACM 55(10), 78–87 (2012)

Saeed, M. (2022) *Calculating pearson correlation coefficient in python with Numpy*, *Stack Abuse*. Stack Abuse. Available at: https://stackabuse.com/calculating-pearson-correlation-coefficient-in-python-with-numpy/ (Accessed: December 13, 2022).

Arora, P., Deepali and Varshney, S. (2016) “Analysis of K-means and K-medoids algorithm for Big Data,” *Procedia Computer Science*, 78, pp. 507–512. Available at: https://doi.org/10.1016/j.procs.2016.02.095.

Moussa, R. and Sarro, F. (2022a) ‘On the use of evaluation measures for defect prediction studies’, *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis* [Preprint]. doi:10.1145/3533767.3534405.