ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

Master Thesis Data Science and Marketing Analytics

# Examining different machine learning algorithms to detect fake consumer reviews.

November 1, 2023



| | |
|---|---|
| Author | E. F. Kremer (498394) |
| Supervisor | prof. dr. P.J.F. Groenen |
| Second assessor | prof. dr. M.G. de Jong |

**ABSTRACT**

Online reviews have become an important resource for customers making purchasing decisions. However, the widespread availability of fake reviews, intentionally created to deceive readers, poses a significant challenge. This review fraud undermines the credibility of online reviews, eroding customer trust, fostering unfair competition among companies, and carrying financial consequences. The study employs multiple supervised machine learning classifiers, Naive Bayes, Decision Tree, Support Vector Machine, and K-Nearest Neighbors, to analyze the content of reviews for fake review detection. The detection of fake reviews is crucial for companies, consumers, and the marketplace as a whole. Identifying the most effective classifier can guide companies in enhancing their fake review detection mechanisms, ultimately boosting consumer trust and increasing user engagement and sales. Additionally, this research serves as a benchmark for future studies in fake review detection using machine learning techniques.

## Table of Contents

## 1. Introduction

Customers can express their reviews and opinions on various online platforms. These opinion-sharing platforms enable customers to share their firsthand experiences and opinions, helping prospective customers seeking information on products or services that others have already tested and authorized. Prior studies (Park, Lee, & Han, 2007; Fagerstrøm, Ghinea, & Sydnes, 2016; Fan, Che, & Chen, 2017) have demonstrated that customers increasingly rely on reviews to gather product or service information. Consequently, writing reviews has a big impact on customers' purchasing decisions, making it highly relevant for marketers. As reviews carry significant influence over customers, certain companies have been drawn to malicious practices (Munzel, 2016). Dishonest companies may be tempted to make use of automated software programs or employ writers to create fake online reviews, either to increase the desirability of their products and services or to harm the reputation of competitors. These malicious practices, aimed at influencing customer opinions by posting deceptive content, are referred to as deceptive online communication, and it goes by various names such as 'fake reviews', 'deceptive reviews', 'deceptive opinion spam', 'review spam' or 'review fraud' (Mayzlin et al., 2014). The detection of fake online reviews (from here on, "fake reviews") holds significant importance for companies and marketing due to three primary reasons. Firstly, the impact of fake reviews poses a significant threat to customer trust in reviews, which can potentially lead to a substantial market decline. Reviews are a valuable service within the marketplace, aiding customers in making informed decisions. Additionally, companies benefit from an authentic review environment, as it allows them to collect genuine feedback that can be utilized to enhance their products and services. However, the proliferation of fake reviews on a large scale poses the threat of eroding the credibility of reviews as a reliable source of information. Secondly, fake reviews may lead to unfair competition in which a product's ranking is unfairly boosted or lowered (He et al., 2022). This is due to the fact that reviews are used by online marketplace algorithms to assess how well a product ranks against others in its category. This makes it possible for companies to use fake reviews as a weapon: a positive fake review can boost the product's ranking, while a negative fake review can have a detrimental effect on its ranking. An unethical company may, for example, flood the market with negative reviews about a competitor. By oversaturating the market with these reviews, it could negatively impact the visibility of the attacked company on online raking algorithms (such as those used by platforms like TripAdvisor, Amazon and Yelp). Lastly, the impact of fake reviews go beyond reputational consequences and includes a financial burden. For instance, an one-star increase on an Amazon product can boost sales up to 26 percent, according to the e-commerce consulting firm Pattern

(Maheshwari, 2019). Furthermore, research by Luca (2011) reveals that a reduction of one star in a company's Yelp rating can lead to a significant decline in revenue, ranging from five to nine percent.

Governments have expended serious efforts to ban deceptive practices and penalizing those responsible, with the aim of limiting the prevalence of fake reviews. China, for instance, passed its first “Electronic Commerce Law” in 2018. The law states that companies cannot use fictitious transactions, fake online reviews, or any other means to defraud or mislead customers through false or misleading commercial promotions (CIRS, 2018). Likewise, the Federal Trade Commission (FTC) in the United States has proposed a trade regulation rule to stop companies from using deceptive practices in their product reviews and testimonials. If the new rule is finalized, companies who “buy, sell, and manipulate online reviews” would be subject to penalties of up to $50,000 per violation (Federal Trade Commission, 2023). Nonetheless, despite the proactive effort of legislators to combat fake reviews, implementing legal measures against deceivers is a challenging task, primarily because of the inherent difficulty in identifying fake reviews.

Clearly, the presence of fake reviews hinders the potency of reviews, posing a challenge for both society and the marketplace as it undermines trust and has costly repercussions (Keep & Schneider, 2009). The detection of fake reviews has developed into an active and important research topic. In this study, multiple types of supervised machine learning classifiers are employed to identify fake reviews based on the reviews' content. This study aims to answer the following research question:
*"Which supervised machine learning classifier among Naive Bayes, Decision Tree, Support Vector Machine, and K-Nearest Neighbors is the most effective for detecting fake reviews?"*.

This research questions can provide several valuable insights and benefits. Gaining insights into most effective classifier can guide companies aiming to enhance their fake review detection mechanisms. This, in turn, fortifies the credibility of their reviews, which can lead to increased consumer confidence and, ultimately, higher user engagement and sales. Moreover, our findings serve as a benchmark for future research. Researchers can use this information as a reference point for further advancements in fake review detection through machine learning techniques.

This research is structured as follows: Section 2 discusses relevant literature on the used machine learning algorithms. Section 3 elaborates on the data sets that are used in this research. We discuss the topics in depth in the technical background chapter in Section 4. In Section 5, we explain the methods we applied to answer the research questions. Section 6 shows the results, and Section 7 states the conclusion and discussion.

## 2. Literature

This section presents a review of the relevant literature for our research. Online reviews have become very influential in modern marketing, yet they suffer from the challenge of fake reviews. Fake reviews have different origins, spanning those generated by the companies themselves, rival businesses, or freelance reviewers. Furthermore, this section will shed light on incentivized reviews and how they differ from fake reviews. Fake reviews are intentionally misleading, while incentivized reviews come from individuals enticed by discounts or free products in exchange for a review. With AI-generated fake reviews on the rise, the detection of fake reviews can be difficult. While manual detection is costly and unreliable, natural language processing may offer a potential solution.

### 2.1. Online Reviews

The internet has added new opportunities for customers to exchange information, experiences, and opinions with fellow customers. Online reviews are the modern-day equivalent of word-of-mouth (WOM) marketing. As opposed to traditional WOM communication, whose influence is often limited to a local social network, its digital counterpart, electronic word-of-mouth (eWOM) (e.g. online reviews), can have a significant influence outside of the local community (Schinler & Bickart, 2005). An online review is an online record made by an individual, who is usually not associated with the company, about their view or opinion of a product or service (Chowdhary, Pandit, 2018). It presents a way to learn about customer preferences, product quality as well as product's shortcomings.

Notably, individuals tend to place greater trust in fellow customers than in marketing-related sources such as digital advertising (Nielson, 2013). Roughly eight-in-ten Americans state that they consult online reviews before making a purchase (Smith & Anderson, 2016). Therefore, reviews have evolved into a critical factor influencing customers' purchasing decisions, directly affecting the sales and reputation of vendors (Barbado et al., 2019). Online reviews serve as a mechanism for market regulation by highlighting goods and services of subpar quality (Malbon, 2013). They contribute to the moderation of bad seller behavior and raises market standards and efficiency. As a result, online reviews have emerged as a fundamental element of the marketing strategy, with companies becoming more conscious of their impact (Chen & Xie, 2008). However, due to how influential online reviews are, it creates both opportunities and incentives for dishonest companies for manipulating customers decisions through fake reviews (Ahmed et al., 2018).

## 2.2. Fake Reviews

So far, there has been no universally accepted definition of fake reviews. Ott et al. define fake reviews as "*fictitious reviews that have been deliberately written to sound authentic, in order to deceive the reader*" (Ott, Cardie, and Hancock, 2012, p. 201). Similarly, Lee et al. (2016), define a fake review as one that is generated or written without having experience with the product or service being reviewed. Fake reviews are the result of manipulating marketplace information by posing as a customer (Boush, Friestad and Wright, 2019). Due to the potential impact reviews have on customers' buying behavior, many vendors, retailers, and platforms are tempted to make use of fake reviews (Wu, Ngai, Wu, & Wu, 2020). Fake reviews may originate from various origins, including the company itself, rival companies, or freelance reviewers. The desired effect of fake reviews can either be to increase the attractiveness of a company's own goods and services, or to harm a competitors' reputation (Salminen et al., 2022). With monetary or reputational gain as the end goal. The major issue with this rising practice of this masked marketing lies within the fact that it forms a threat to customers' ability to make better informed purchasing decisions (Lappas, 2012). It goes without saying that fake reviews lead to an unfair playing field between companies and disillusioned customers.

Companies can purchase fake reviews, which are openly traded on the internet. Dishonest companies make use of Facebook and Telegram groups to recruit fake reviewers. These groups bring in thousands of people willing to write a 5-star review in exchange for a free product or a refund. He et al. (2022), investigated 23 Facebook groups dedicated to the exchange of fake reviews for Trustpilot, Google, and Amazon. These groups collectively had over 360,000 members among them. On average 568 fake review requests are posted per day per group (He, Hollenbeck, & Proserpio, 2022). The majority of people in these groups are average consumers who are looking to earn a little extra money or to receive goods for free. However, as the market for fake reviews expands quickly, some now choose to make a living off of it. People make anywhere from $12 to several hundreds of dollars per review, according to a small survey of Oak and Shafiq (2021).

## 2.3. Incentivized Reviews

Incentivized reviews are a concept related to fake reviews. Incentivized reviews refer to reviews gained through a marketing campaign or by offering customers a discounted or free product in return for a review (Petrescu et al., 2018; costa et al., 2019). However, there are key differences: incentivized reviews are often linked to an 'real person' (e.g. an influencer) whereas fake

reviews are frequently published anonymously or pseudonymously. Given that the incentivized reviews are published under their own names, the plausibility of incentivized reviews may be stronger than that of fake reviews. An influencer typically refrains from publishing misleading reviews because they place a great value on their reputation in order to uphold the trust of their audience (kaabachi et al. 2017). Furthermore, incentivized reviews do not necessitate a positive review, while compensation for fake reviews typically requires a five-star rating review. Additionally, the majority of the authors of incentivized reviews have experience with the product or service, enabling for a genuine review. Nevertheless, it is important to note that incentivized reviews have the potential to be upward biased (Salminen, Kandpal, Kamel, Jung, & Jansen, 2022). In contrast, fake reviews frequently lack experience with the product or service, making it impossible for them to accurately reflect a true (dis)like of the product. Lastly, incentivized reviews generally include a disclaimer in the review itself stating that the product was received for free or at a discount in exchange for the review.

### 2.4. Artificially Created Reviews

Due to technological developments in text generation, particularly in machine learning (ML) and Natural Language Processing (NLP), fake reviews are now produced not only by humans but also by computers. Fake reviews generated by Artificial Intelligence (AI) systems are trained on real reviews and are therefore practically impossible to distinguish apart. In addition, AI can generate fake reviews in seconds. According to the study of Yao et al. (2017), artificially created reviews could evade human detection and even outperform the reviews written by humans in terms of perceived usefulness (Yao, Viswanath, Cryan, Zheng, & Zhao, 2017). The key differentiator is scale, computer-generated fake reviews can be produced on a far larger scale than human-generated fake reviews. AI has made it possible for nearly anyone to create thousands of fake reviews that appear to have been written by a real person.

### 2.5. Detection Methods

Due to the volume of reviews that are posted online and the high credibility of fake reviews, it frequently is challenging to detect fake reviews. There are no specific words that can help to distinguish fake reviews from genuine reviews (Crawdord, Khoshgoftaar, Prusa, Richter, Al Najada, 2015). Nevertheless, it is essential for both businesses and customers to be able to detect fake reviews in order to remove them. Fake review detection can be done manually or automatically, however doing it manually is generally more expensive, time-consuming, and

inaccurate than doing it automatically. The challenge in manually detecting fake reviews lies in the large amount information. The human capacity to analyze and understand data is not enough to find the fake reviews. Ott et al. (2011) made use of three human judges to determine, individually, whether or not a hotel review on TripAdvisor.com was fake. The human judges have achieved limited performance with accuracies, they were only right about half the time. Given that there are two options to choose from (fake or not fake), the random chance of choosing the correct label is 1/2 = 50%, suggesting that human intuition is not much better than chance (Hovy, 2016; Ott et al., 2011; Plotkina et al., 2020; Salminen et al., 2022). The three judges did not even agree on which reviews they thought were fake, reinforcing the conclusion that they were doing no better than chance. It implies that that either fake reviews are impossible to be detected by humans or humans lack the knowledge to identify them. Furthermore, human suffer from a truth bias, assuming that what they are reading is true until they find evidence to the contrary (Vrij, 2008). Due to a significant truth bias, humans are more likely to classify a review as genuine than fake.

## 3. Data

The amount of public accessible datasets for the examination of fake reviews is limited (Wu, Ngai, Wu, & Wu, 2020). In the study by Ott et al. (2011), a dataset was generated that features 800 fake and 800 genuine reviews. Nevertheless, this dataset has two significant drawbacks. First, the total number of reviews in the dataset (n=1600) is insufficient to train ML classifiers to detect fake reviews at scale. Second, when constructing the dataset, the researcher excluded reviews with ratings below the maximum of five stars. However, the deletion of reviews with ratings below five stars may not be proper, as fake reviews are not exclusively positive; they can also be negative. A dataset created by Sandulescu and Ester (2015) contained 900 genuine and fake reviews, all of which were awarded four- and five-star ratings. However, since only positive reviews were included in this dataset, it came at the expense of detecting negative fake reviews. Moreover, the dataset is not openly accessible, which prevents replication and further advancement.

The dataset used in this study is the dataset from Salminen et al. (2022). They created a dataset based on the publicly available Amazon Review Data (2018) dataset. Salminen et al., trained OpenAI's Generative Pre-trained Transformer 2 (GPT-2) model on samples of the Top-10 product categories of Amazon (based on review count). A total of 40,000 samples were extracted from these product categories for model training, and 10,000 samples were extracted for model testing. With roughly the same number of samples for each product category (approximate 2500 per product category for model training and approximately 625 per product category for model validation), see Figure 1. The created dataset consists of 20,216 fake reviews generated by GPT-2 and 20,216 genuine reviews that were presumed to be written by humans. The genuine reviews are original samples from the Amazon Review Data (2018) dataset. In this study, fake reviews are defined as fictitious reviews intentionally written, with the intent to deceive the reader, as per the definition provided by Ott, Cardie, and Hancock (2012, p. 201). There could potentially be an unknown number of fake reviews in the genuine review dataset. Regrettably, it is impossible for us to definitively confirm the veracity of any review within the dataset. We use the 'genuine' reviews, with the assumption that a small percentage (less than 5%) of them might be fake, with the understanding that this minimal proportion is unlikely to significantly affect our ML models. Hence, there are 40,432 reviews in total, this is a sufficient number of samples for a text classification task.

The dataset contains 4 features including *category, rating, label* and *text.* The feature *category* refers to the top ten product categories of Amazon with the most product reviews (Toys & Games; Tools & Home Improvements; Sport & Outdoors; Pet Supplies; Movies & TV; Kindle Store; Home & Kitchen; Electronics; Clothing, Shoes & Jewelry; Books). These categories make up 88,4% of Amazon's total product reviews, adequately capturing the dataset. Figure 1 shows that each product category has a roughly equal number of reviews, with approximately 2,000 reviews per product category.

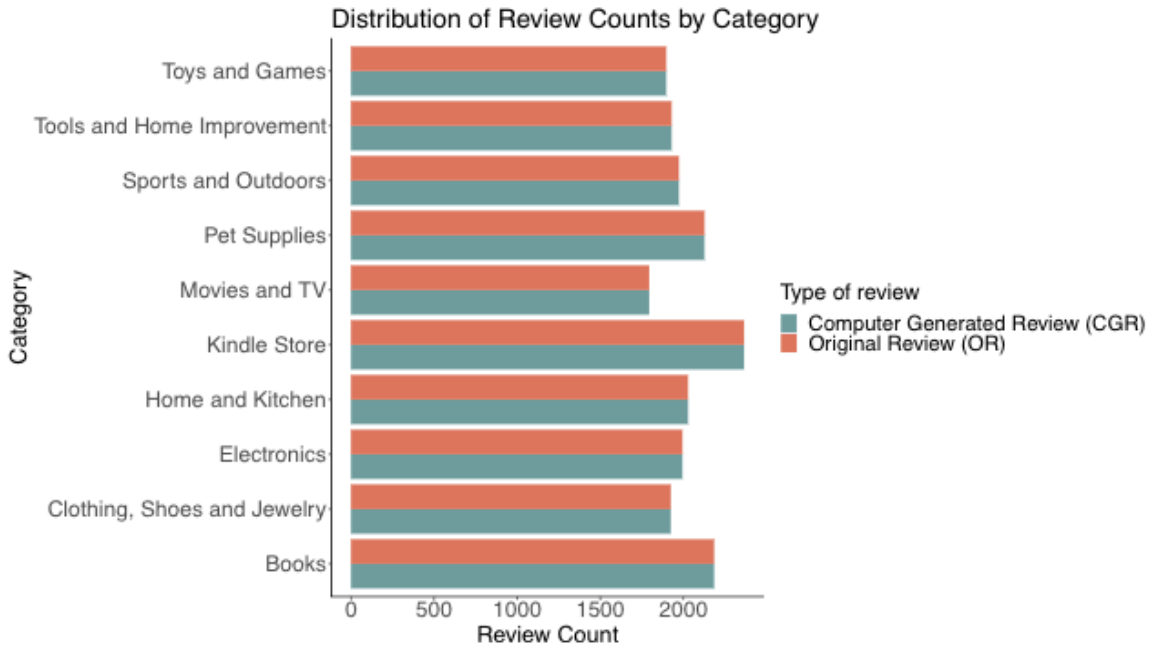


*Figure 1: Distribution of review counts by category*

The feature *rating* refers to the rating level of the review. There are nearly the same amount of reviews for both classes for each rating level, from one star to five stars. Figure 2 illustrates a notable disparity in the number of 5-star reviews compared to the much lower number of 1, 2, 3, and 4-star reviews. Different rating levels must be taken into consideration because fake reviews can be both positive and negative.

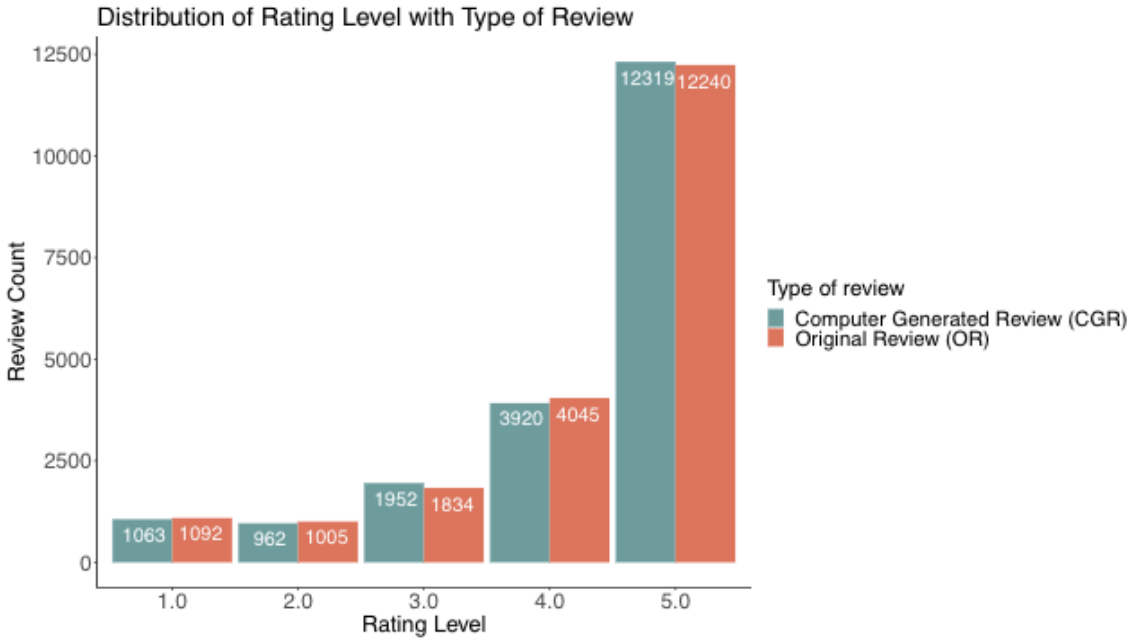


*Figure 2: Distribution of rating level with class labels*

The feature *label* indicates whether the review is a computer-generated review (CGR) or an original review (OG). The data is perfectly balanced and contain equal numbers of the classes (2,0216), so there's no need for handling imbalanced classification problems. The feature *text* displays the text of the review. Table 1 shows an excerpt of a genuine and fake review in our dataset in the Pet Supplies category.

| Genuine Review | Fake Review |
|---|---|
| I am very happy with these remote collars. As a professional dog training I like the fact that the vibration part has several levels as well as the static. At this point Im very happy with these collars. | I could not have been happier. The pups were not annoyed at all by their collars. I kept an eye on them so they were neither to loose or too tight. And MADE IN THE USA!!!!! |

*Table 1: Sample genuine (OR) and fake (CGR) 5-star reviews in the Pet Supplies category*

We created an extra feature called *textlength,* to assess its potential contribution to the identification of fake reviews. This feature measures the length of a review by counting the number of characters it contains.

Figure 3 visualizes the distribution of review lengths for both fake and genuine reviews. There is no clear distinction between fake and genuine reviews based on review length. However, it appears that fake reviews tend to be shorter in length compared to genuine reviews.

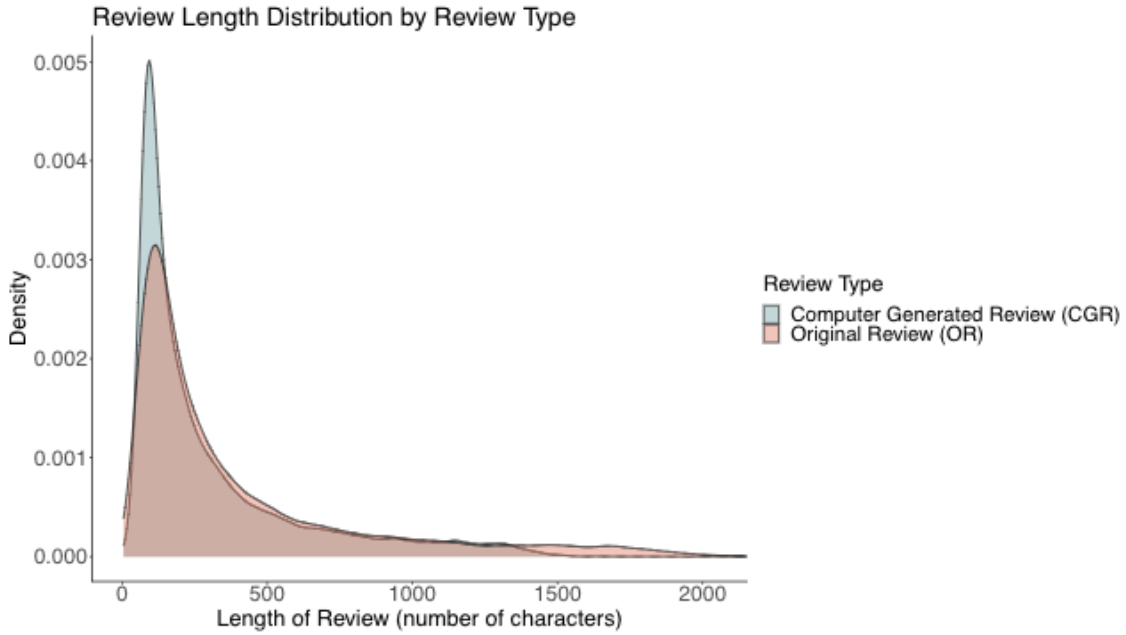


*Figure 3: Distribution of review lengths with class labels*

# 4. Technical Background

In this chapter, we explain the theories that support our research. We begin by data pre-processing, move on to the essential concept of Term Frequency and Inverse Document Frequency (TF-IDF), and then delve into a technical summary of the supervised machine learning algorithms, including Naïve Bayes, Decision Trees, Support Vector Machines, and k-Nearest Neighbors.

## 4.1. Data Pre-Processing

Text is commonly referred to as unstructured data. Text typically lacks labels or categories and is often not organized in a structured way, making it unstructured data. Text mining is the process of transforming unstructured text into a structured format. Text mining can accomplishes this through the use of Natural Language Processing (NLP). NLP is a component of text mining that performs a linguistic analysis that helps a machine to processes and interpret text. In order for a machine learning model to understand the unstructured review text data, text preprocessing is a critical step in NLP. The data preprocessing steps includes (1) tokenization, (2) lower casing, (3) stop words removal (4) punctuation removal (5) stemming, and (6) common & rare words removal.

Tokenization is the first step in the NLP pipeline, it breaks text into smaller units called tokens. Tokens can be either words, characters, or subwords. Tokenization ensures that each token in a sentence stand on their own. Most of the preprocessing steps and modeling happens at the token level. Tokenization, for instance, will separate the review *"A little pricey for what it is"* into the tokens *"A", "little", "pricey", "for", "what", "it", "is"*.

In the next step we lowercase the tokens. Although words like *"pricey", "Pricey"* and *"PRICEY"* have the same meaning, models will treat them differently. The idea is to convert the review text into the same lowercase format, so that words like *"pricey", "Pricey"* and *"PRICEY"* are converted to "pricey" and treated the same way.

Stop words, words that frequently recur in text, include words like "a", "the", "is", "was", "this", etc. These words do not add any meaningful information, and we do not need them to distinguish fake reviews from genuine reviews. As the frequency of stop words are high, removing them from the review text data results in a significantly smaller dataset. Take the review *"A little pricey for what it is*" as an example. The review will show up as *"little pricey"* after eliminating stop words.

Punctuation marks, such as commas, question marks, exclamation marks and periods, are frequently viewed in NLP as noise or irrelevant information. Eliminating punctuation can

make the review text data simpler and makes it simpler for the model to identify important features that are relevant for the classification task.

Stemming is the process of stripping the final few letters from a word to obtain their stem. By converting words to their stems, NLP can treat different inflections of words as a single entity, reducing the complexity of the text data. For stemming, there are several different algorithms. We applied the Porter stemmer, which is the most widely used algorithm, it uses a set of heuristics to strip common suffixes from words (Porter, 1980). The words *"playing", "played",* and *"plays"*, for example, can all be reduced down to the common word stem *"play"*.

The most common words are removed because the most scoring systems give more weight to the detection of these common words, which can overshadow the importance of less common word. The most rare words are removed because they are unlikely to contribute meaningfully due to their infrequent occurrence.

### 4.2. Term Frequency–Inverse Document Frequency

Text documents must be converted to a format that classification algorithms can understand. Most classification algorithms require input in the form of vectors or matrices. For this reason, documents are represented as a vector and the corpus (the collection of documents) as a matrix. The most frequently used form of text transformation is known as the Vector Space Model (VSM), in which the documents are represented by vectors of words (Raghavan & Wong, 1986; Salton, Wong, & Yang, 1975). The size of the vector is equal to the size of the vocabulary (i.e., the number of unique terms (words) in the corpus).

Document $j$ can be represented as vector:

$$X_j = \left(x_j^1 \;\; x_j^2 \;\; \dots \;\; x_j^M\right),$$

where $M$ is the size of the vocabulary, and the element $x_j^i$ is the weight of term $i$ in document $j$. This weight is defined as the product of Term Frequency (TF) and Inverse Document Frequency (IDF) for term $i$ in document $j$:

$$x_j^i = \text{TF - IDF}(i,j) = \text{TF}(i,j) \times \text{IDF}(i).$$

TF - IDF is statistical metric used to measure how important a term is to a document in a collection of documents. It is calculated as the product of TF and the IDF. TF measures how frequently a term $i$ appears in a document $j$ $\left(\text{TF}(i,j)\right)$. IDF examines the rarity of a term appears

across the corpus. It is the logarithm of the total number of documents in the corpus divided by the number of document where term $i$ appears. The formula for IDF is:

$$\mathrm{IDF}(i) = \log\left(\frac{N}{df(i)}\right),$$

where $N$ is the size of the corpus and $df(i)$ stand for the document frequency of term $i$ (the number of documents containing term $i$). Terms unique to a small percentage of documents receive higher importance values than words common across all documents

By multiplying TF and IDF, the TF-IDF value can be obtained. The commonality of a term within a document measured by TF is balanced by the relative rarity of the term in the corpus measured by IDF. TF - IDF makes rare terms more prominent and effectively ignores common terms. The TF - IDF score reflects the importance of term $i$ for document $j$ in the corpus. The importance of a term is high when it occurs a lot in a document and rarely in the corpus. The result of applying this TF - IDF transformation to all the documents, is a Document-Term Matrix (DTM). $X_j$ represents the columns in the matrix. Each $X_j$ is the vector representation of document $j$ in the corpus. The documents are represented by the rows and its entries represent to the TF - IDF weights of the terms in that document.

### 4.3. Supervised Machine Learning

With the use of machine learning (ML), which is a type of Artificial Intelligence (AI), computers can learn and make decisions without being explicitly programmed. It is based on the idea that computers can identify patterns, develop understanding, make decisions, evaluate their confidence, and learn from it all without receiving much assistance from humans. A ML algorithm needs training data to learn from. Training data will vary depending on whether you are working with supervised or unsupervised machine learning. Unsupervised learning uses unlabeled data, which means that there is no target assigned to the data. The algorithm has to act on the data without guidance; they are left to their own to discover and present patterns and insights in the data. Supervised machine learning algorithms uses labeled data, where the label is the target we are interested in predicting, implying that you have the values of the inputs and outputs. The idea behind the labels is to train algorithms to recognize patterns. In this manner, the input of new, unlabeled, data will result in a prediction of the unknown output.

### 4.3.1. Naïve Bayes

The Naïve Bayes (NB) classifier is a probabilistic supervised machine learning algorithm, which is used for classification tasks, like text classification. NB is based on Bayes' Theorem with the assumption of independency among predictors. Bayes Theorem finds the probability of an event ($y$) occurring given the probability of another event ($X$) that has already occurred ($P(y|X)$):

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}.$$

The variable $y$ is the target variable, variable $X$ represents the features, which is given as: $X = (x_1,\ x_2, \dots,\ x_n)$.

By substituting for $X$ and using the chain rule we get:

$$P(y|x_1, \dots, x_n) = \frac{P(x_1|y)P(x_2|y)\dots P(x_n|y)P(y)}{P(x_1)P(x_2)\dots P(x_n)}.$$

It is assumed that the features are conditionally independent. That is, the presence of one feature does not affect the other. It also assumes that all features are given equal weight; all features contribute equally to the outcome. Since these assumptions are rarely possible in real-life data, the classifier is described as naïve. Despite these simplified assumptions, the Naïve Bayes classifiers can deliver remarkably strong performance in many real-life applications such as text classification (Rish, 2001). To classify a data point, Naïve Bayes calculates the likelihood probability of each class label given the features of the datapoint. It then selects the class with the highest probability as the predicted class.

### 4.3.2. Decision Tree

A Decision Tree (DT) is a non-parametric supervised learning algorithm which is used for both classification and regression tasks. The objective is to learn the algorithm simple decision rules, in order to build a flowchart-like tree structure that will lead to the correct classification. In the tree, the internal node represents a feature, the branch represents a decision rule, and each leaf node represents the outcome.

To classify a data point, the decision process starts at the root note and follows the branches and internal notes, where it evaluates the values of different features, and ends at a leaf node, which presents the final classification for the given data point. The decision rules are based on statements equal to "if-then-else" conditions, it checks if the conditions is true and if it is, it proceeds to the next node attached to that decision. This sequential evaluation process continues until a final decision or outcome is determined. The tree is built in a top-down manner, starting with the root node, which is the most important feature for splitting the data. The tree construction process continues recursively, with each node selecting the most informative feature to split the data into subsets. The choice of when to stop growing the tree is crucial. Typically, real-world data often contain many features, leading to a large number of splits in the Decision Tree, ultimately yielding a large and complex tree structure. Building such extensive tree take time to build and can often result in overfitting. The choice of when to stop growing the tree, also known as pruning, is therefore crucial. A common strategy for controlling the growth of decision trees is setting a maximum depth of the tree, it limits the number of nodes in the tree. The tree stops growing once it reaches this specified depth. This is a straightforward way to prevent the tree from becoming too deep and complex. Another common strategy is setting a minimum number of data points for a node to be split further. If a node has fewer data points than the specified threshold, it would not be split, and the tree stops growing at that branch.

### 4.3.3. Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm mostly used in classification problems but is also suitable for regression problems. The goal of SVM is to identify a hyperplane in a p-dimensional space (where p is the number of features) that best divides the data points into different classes. "Best" is defined as the hyperplane with the largest separation, or margin, between the closest points of different classes. A wider margin indicates better classification performance. The data points that are closest to the hyperplane are called the support vectors. The hyperplane serves as decision boundary for classifying data points, attributing data points on each side of the hyperplane to distinct classes. To determine to which class a data point belongs, SVM examines its position relative to the hyperplane. If the data point falls on one side of the hyperplane, it is classified into one class, and if it falls on the other side, it is classified into the other class. In practice, it is challenging to find a hyperplane that perfectly separates the data since it is not always feasible to draw a line that neatly divides the classes due to noise or overlapping data points, i.e. the data is linearly inseparable. Soft margin

classifier is a technique that may be used to enhance this. With the soft margin classifier, the constraint of maximizing the margin of the hyperplane is loosened, this allows a few data points to get misclassified. It introduces a penalty parameter, hyperparameter C, for data points that fall on the wrong side of the hyperplane. A large penalty parameter places more emphasis on minimization of misclassification of the data points, resulting in a narrower margin. Conversely, a small penalty parameter prioritizes maximizing the margin at the expense of misclassifying some of the data points (Misra, 2019). So, by tuning the penalty parameter, you can control the balance between maximizing the margin and minimizing the misclassification. Another technique for handling linearly inseparable data involves utilizing a kernel trick. The kernel trick involves training a linear SVM model to learn a nonlinear function in a high-dimensional feature space while controlling the system's capacity with a parameter that does not depend on the dimensionality of the feature space. Simply explained, it converts low-dimensional input space into higher-dimensional space where a hyperplane can more effectively separate the classes. In other words, it transforms a non-separable problem to a separable problem. Popular kernels include polynomial and radial basis function (RBF) kernels.

### 4.3.4. K-Nearest Neighbors

The K-Nearest Neighbors (KNN) algorithm is a non-parametric, supervised machine learning algorithm used to solve classification and regression problems. It is based on the idea that the observations most similar to a particular data point are those that are closest to it in the data set. Generally, KNN classifiers use Euclidean as distance metric to measure similarity, the smaller the distance, the more similar. Let $X_1 = (x_1^1 \ x_1^2 \ \dots \ x_1^M)$ and $X_2 = (x_2^1 \ x_2^2 \ \dots \ x_2^M)$ be two vectors, where $M$ is the dimension of each vector then, the Euclidean distance formula is given by the equation:

$$d(X_1, X_2) = \sqrt{\sum_{i=1}^{M}(x_i^1 - x_i^2)^2}$$

The distances are arranged in ascending order, ensuring that the closest neighbors appear at the beginning, followed by those progressively farther away. By choosing parameter K it can be specified how many neighboring observations will be used in the algorithm. It allows us to classify unseen data points based on the values of the existing data points that are closest to them. The decision rule puts a data point into a particular class if the class has the majority vote among the K nearest neighbors. If the difference between the number of points belonging to the two competing classes (i.e., two classes with the majority vote) among the k nearest neighbors

is at least one then the point will be assigned to that class. Simply said, KNN seeks to identify the class to which a data point belongs by examining the data points around it.

## 5. Methodology

In this section, we present the methodology employed in this study to develop and compare the ML models. The proposed model shown in Figure 4, consists of 5 stages to identify the best classification model for fake review detection. The details of these stages are described as follows:

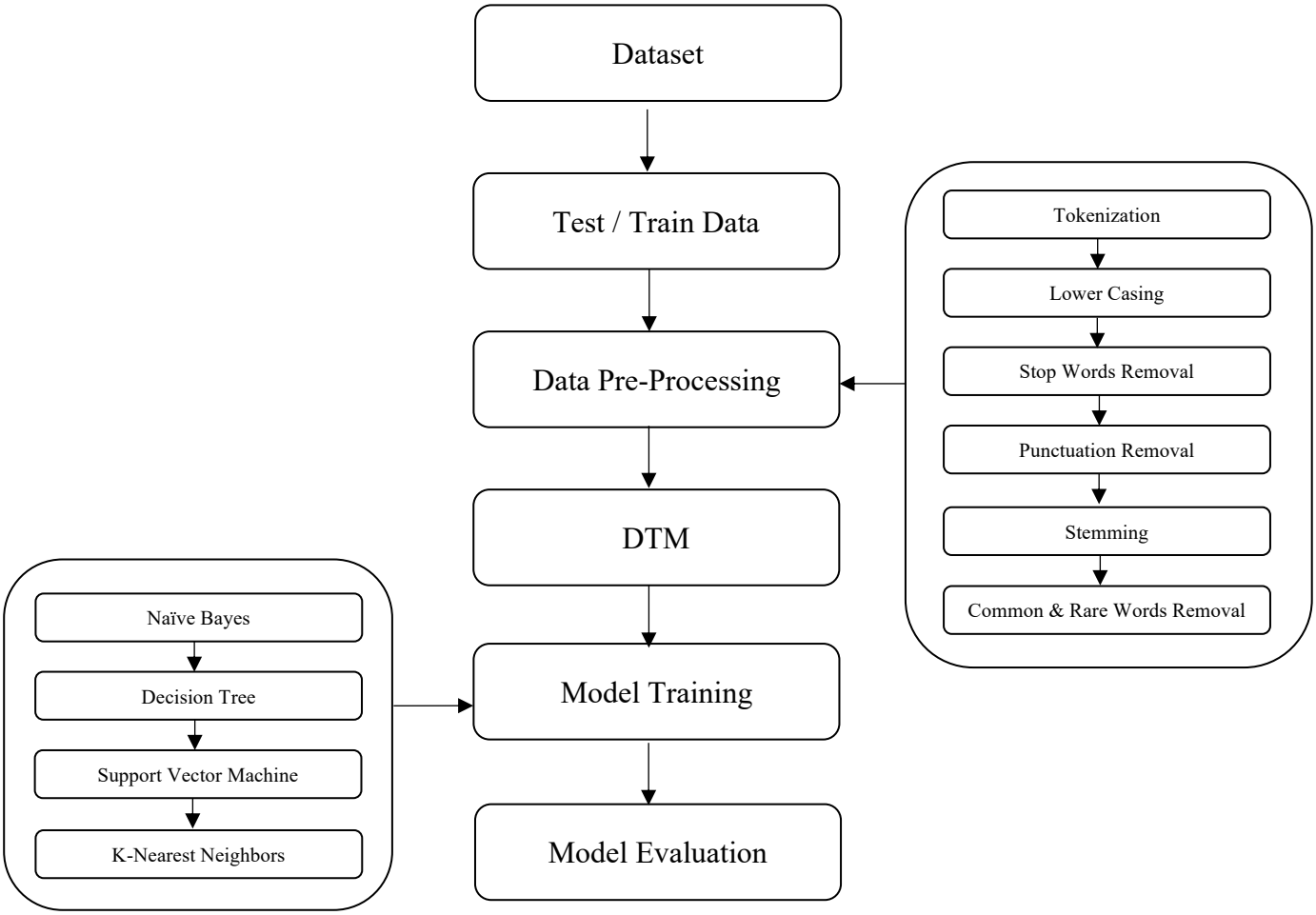


*Figure 4: Proposed Model*

The first step in our proposed model is cleaning the data, to get rid of invaluable information. There are no missing values in the data, all features are character vectors. We transform the feature *label* data to a factor with two levels: "CGR" and "OG". This makes it easier for algorithms to process because the majority of ML algorithms are built to work with numerical inputs.

A 70/30 split was utilized to train and test the ML models, which implies that 70% of the dataset (n = 28,304) was randomly selected to train the model and the remaining 30% (n = 12,128) was held out for evaluation purposes. The test set therefore includes samples that the model was not exposed to during model training. The train and test set have both 5 features (*label, category, text, textlength* and *rating*). The train set will be fit on the ML models.

The pre-processing steps were taken to clean the data in the *text* feature for both the training set and the test set. Initially, all the reviews underwent word-level tokenization, where each token corresponded to an individual word within a review. These individual words were subsequently converted to lowercase, and both stop words and punctuation were removed. Following that, the words were stemmed using the Porter stemmer to derive a word's stems. In the final preprocessing step, words that appear in more than 95% of the reviews or less than 5% of them were removed. To prevent 'data leakage', the train and test sets are pre-processed independently. By applying preprocessing techniques to the entire data set, there is a risk that information from the test set can 'leak' into the training data. We must ensure that words in the test set do not become part of the vocabulary of the train model. It will give the model a 'heads-up' about the unseen data, resulting in an inaccurate evaluation and unreliable predictions. We ensured that preprocessing steps are consistently applied to both the training and testing sets.

Following the preprocessing phase, we convert the train and test text data into a numerical format by constructing a DTM. The DTM represents the word frequencies of each word in a review, each feature corresponds to a unique word in the corpus. The training DTM comprises 25,702 features, while the test DTM comprises 16,272 features. To apply TF - IDF weighting to the DTM, we must be sure that TF - IDF is applied in a consistent way, because the IDF component is computed based on the entire corpus. We fit the TF - IDF vectorizer on the corpus of the training data, which involves creating a vocabulary from the words in the training reviews and calculating the TF - IDF values for each word in the training data. The values quantify the importance of each word in the context of the entire training corpus. The trained IDF vectorizer is then applied to calculate the TF - IDF values of the testing corpus. By doing so, we ensure that the testing data are represented in the same vector space as the training data. The vectorizer uses the previously learned vocabulary to convert the new reviews into TF - IDF vectors. This transformation ensures that the testing set have the same number of features as the training set, which is 25,702 features. To these DTMs, the feature *label* from the original dataset was added, which was used as the dependent variable in the ML models. In the training DTM the additional

features, *rating, category* and *textlength,* were also included, to be able to calculate the feature importance of these features. Therefore, we use the varImp function within the caret package in R, which calculates the increase in the prediction error resulting from permuting feature values. If such permuting does not change the model error, the corresponding feature is considered as unimportant. The variable importance helps identifying which features have the most influence on a model's performance. The higher the variable importance score, the more significant the feature is in making accurate classifications.

With these training and testing DTM matrix, the models are trained with a 5-fold cross validation to assess the model performance and tune the hyperparameters. A 5-fold cross validation is a technique in which an original sample is randomly split into five equal-sized subsamples. The training is performed on four of these subsamples, while the remaining subsample serves as the testing set. This process is repeated five times to ensure that each subsample serves as the testing set once.

The right metrics for evaluating fake review detection models, which is a binary classification problem, are crucial. Relying solely on accuracy as the performance measure may not be sufficient. We aim to minimizing the misclassification of genuine reviews as fake, as this could damage a seller's reputation, and minimizing the misclassification of fake reviews as genuine, which could result in a poor customer experience. As a result, we use the performance metrics listed Table 2 for our evaluation of the ML model's performance. These metrics together offer a more nuanced understanding of the model's performance in fake review detection.

In binary classification, one class is termed positive and the other is termed negative. The positive class are the fake reviews and the negative class are the genuine reviews. 'True Positives' refers to correctly classified the positive class (fake reviews), while 'False Positives' refers to incorrectly classified the positive class (fake reviews). ‘True Negatives’ refers to correctly classified the negative class (genuine reviews), and 'False Negatives' refers to incorrectly classified the negative class (genuine reviews).

| Performance Metrics | Description |
|---|---|
| Accuracy Score | The ratio between the number of correctly classified samples and the overall number of samples.<br>$$\text{Accuracy Score} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total number of instances}}$$<br>It answers the question: "When the model says that a review is genuine or fake, what is the probability that it is correct?" |
| Recall Score | The ratio between the number of positive samples correctly classified as positive to the total number of actual positive samples.<br>$$\text{Recall Score} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$<br>It answers the question: "Of all the fake reviews in the review data set, what fraction did the model detect? |
| Specificity Score | The ratio between the number of negative samples correctly classified as negative to the total number of actual negative samples.<br>$$\text{Specificity Score} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}$$<br>It answers the question: "Of all the genuine reviews in the review data set, what fraction did the model detect? |
| Precision Score | The ratio of correctly classified positive samples to the total number of samples classified as positive.<br>$$\text{Precision Score} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$<br>It answers the question: "Of all the reviews classified as fake, how many reviews were correctly classified as fake?" |

*Table 2: Performance metrics*

## 6. Analysis and Results

In this section, we evaluate the performance of Naïve Bayes, K-Nearest Neighbors, SVM, and Decision Tree, in identifying fake reviews. Each classifier is accompanied by a confusion matrix and key performance metrics, providing valuable insights into their performance. Prior to this evaluation, we calculate the feature importance of the relevant features.

Figure 5 presents a visual representation of the feature importance for the top 24 features. The varImp function utilized scales the variable importance up to 100. Notably, the variable importance scores for the additional features (*rating, category* and *textlength*) are relatively low and do not make it into the top 24 rankings. As a result, we made the decision to omit these features from the model, opting to exclusively use the DTM matrix for our ML models.

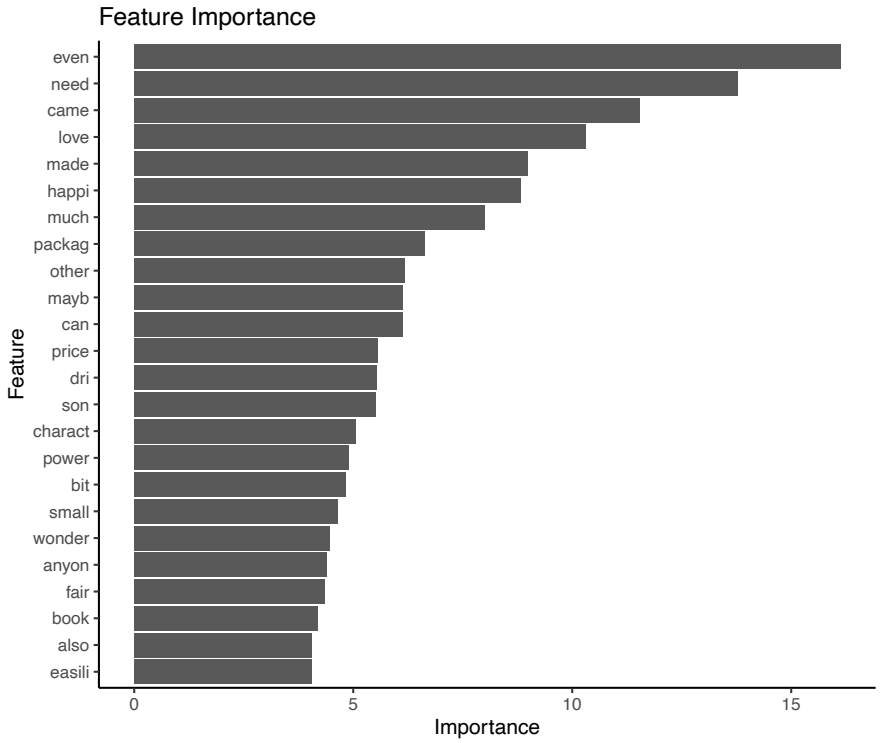


*Figure 5: Feature importance*

The confusion matrix from the testing with Naïve Bayes classifier is given in Table 3. From the confusion matrix, we conclude that out of 12,128 test reviews, 9,037 were correctly classified. Of the genuine reviews, 1,754 were incorrectly classified as fake and 1328 fake reviews were incorrectly classified as genuine. As shown in Table 4 the accuracy, recall, specificity score,

and precision score that we obtained with Naïve Bayes classifier are 74.51%, 80.59%, 66.76%, 75.87%, respectively.

| | | Actual | |
|---|---|---|---|
| | | CGR | OR |
| Predicted | CGR | 5514 | 1754 |
| | OR | 1328 | 3523 |

*Table 3: Confusion matrix Naïve Bayes*

| Performance Metrics | NB | KNN | SVM | DT |
|---|---|---|---|---|
| Accuracy | 0.7451 | 0.5532 | 0.7843 | 0.7818 |
| Recall | 0.8059 | 0.5322 | 0.8217 | 0.8446 |
| Specificity Score | 0.6676 | 0.5737 | 0.7327 | 0.6964 |
| Precision Score | 0.7887 | 0.5491 | 0.8094 | 0.7910 |

*Table 4: Performance metrics for NB, KNN, SVM and DT*

The confusion matrix from the testing with KNN classifier is given in Table 5. From the confusion matrix, we conclude that out of 12,128 test reviews, 6,710 were correctly classified. Of the genuine reviews, 2,617 were incorrectly classified as fake and 2,801 fake reviews were incorrectly classified as genuine. As shown in Table 4 the accuracy, recall, specificity score, and precision score that we obtained with KNN classifier are 55.33%, 53.22%, 57.37%, 54.91% respectively.

| | | Actual | |
|---|---|---|---|
| | | CGR | OR |
| Predicted | CGR | 3187 | 2617 |
| | OR | 2801 | 3523 |

*Table 5: Confusion matrix KNN*

The confusion matrix from the testing with SVM classifier is given in Table 6. From the confusion matrix, we conclude that out of 12,128 test reviews, 9,512 were correctly classified. Of the genuine reviews, 1,361 were incorrectly classified as fake and 1255 fake reviews were incorrectly classified as genuine. As shown in Table 4 the accuracy, recall, specificity score, and precision score that we obtained with SVM classifier are 78.43%. 82.17%, 73.27%, 80.94%, respectively.

| | | Actual | |
|---|---|---|---|
| | | CGR | OR |
| Predicted | CGR | 5782 | 1361 |
| | OR | 1255 | 3730 |

*Table 6: Confusion matrix SVM*

The confusion matrix from the testing with Decision Tree classifier is given in Table 7. From the confusion matrix, we conclude that out of 12,128 test reviews, 9,482 were correctly classified. Of the genuine reviews, 1,560 were incorrectly classified as fake and 1,086 fake reviews were incorrectly classified as genuine. As shown in Table 4 the accuracy, recall, specificity score, and precision score that we obtained with Decision Tree classifier are 78.18%. 84.46%, 69.64%, 79.10%, respectively.

| | | Actual | |
|---|---|---|---|
| | | CGR | OR |
| Predicted | CGR | 5903 | 1560 |
| | OR | 1086 | 3579 |

*Table 7: Confusion matrix Decision Tree*

## 7. Conclusion and Discussion

In conclusion, this thesis employed four machine learning algorithms, including Naive Bayes (NB), k-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Decision Tree (DT), for the task of detecting fake Amazon reviews based on review text. The performance results demonstrates that SVM outperforms the other machine learning models in terms of accuracy, specificity score and precision score, indicating its efficacy in correctly identifying fake reviews. However, it is noteworthy that the DT classifier does not deviate substantially from SVM, and even outperforms SVM in terms of recall. A higher recall indicates that the DT model is better at capturing and correctly classifying a greater proportion of the actual fake reviews in the dataset, minimizing the number of false negatives (i.e., fake reviews being incorrectly labeled as genuine).

SVM stands out as a strong performer, in various metrics, while the DT classifier proves to be a competitive alternative. The choice between these models should be informed by the specific goals of the application. In a scenario where the focus is on preventing fake reviews, prioritizing precision score can be crucial. This means being stricter in classifying a review as fake to reduce the risk of falsely accusing genuine reviews, which can have reputational consequences for companies. In this case, the company would be willing to accept a lower recall rate, in which some fake reviews might go undetected, to ensure that the reviews identified as fake are indeed fake.

Although, the results we got are quite promising, this study is not without its limitations. First, the experiments were performed on only Amazon reviews, which may restrict the generalizability of the findings. Future research should consider reviews from other e-commerce websites to enhance generalizability. Secondly, as previously acknowledged, the authenticity of the reviews labeled as 'genuine' in our dataset cannot be definitively confirmed. It is impossible for us to definitively confirm the veracity of any review within the dataset. Another limitation is that the training was exclusively conducted using reviews in the English language because the dataset was in this language. Lastly, further research could explore the potential benefits of integrating additional features, such as the reviewer's name and the number of reviews they have written, to potentially enhance the performance of the machine learning models.

# 8. Bibliography


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

Chowdhary, N. S., & Pandit, A. A. (2018). Fake review detection using classification. *Int. J. Comput. Appl*, 16-21.

CIRS. (2018, September 29). *China's First E-Commerce Law Published*. Retrieved September 2023, from CIRS: https://www.cirs-group.com/en/cosmetics/china-s-first-e-commerce-law-published

Fagerstrøm, A., Ghinea, G., & Sydnes, L. (2016). Understanding the impact of online reviews on customer choice: A probability discounting approach. *Psychology & Marketing*, 125–134.

Fan, Z.-P., Che, Y.-J., & Chen, Z.-Y. (2017). Product sales forecasting using online reviews and historical sales data: A method combining the Bass model and sentiment analysis . *Journal of Business Research*, 90-100.

Federal Trade Commission. (2023, Juni 30). *Federal Trade Commission Announces Proposed Rule Banning Fake Reviews and Testimonials*. Retrieved September 2023, from Federal Trade Commission: https://www.ftc.gov/news-events/news/press-releases/2023/06/federal-trade-commission-announces-proposed-rule-banning-fake-reviews-testimonials

He, S., Hollenbeck, B., & Proserpio, D. (2022). The Market for Fake Reviews. *Marketing Science*, 896 - 921.

Hill, S. (2022, November 2). *Inside the Underground Market for Fake Amazon Reviews*. Retrieved from WIRED: https://www.wired.com/story/fake-amazon-reviews-underground-market/

Keep, W., & Schneider, G. (2009). Deception and defection from ethical norms in market relationships: a general analytic framework . *Business Ethics, the Environment & Responsibility*, 64-80.

Lee, S.-Y., Qiu, L., & Andrew, W. (2018). Sentiment Manipulation in Online Platforms: An Analysis of Movie Tweets. *Production and Operations Management*, 385-595.

Maheshwari, S. (2019, November 29). When Is a Star Not Always a Star? When It's an Online Review. *The New York Times*, p. 1.

Malbon, J. (2013). Taking Fake Online Consumer Reviews Seriously. *Springer*, 139-157.

Mayzlin, D., Dover, Y., & Chevalier, J. (2014). Promotional Reviews: An Empirical Investigation of Online Review Manipulation. *AMERICAN ECONOMIC REVIEW*, 2421-2455.

Misra, R. (2019, mei 1). *Support Vector Machines — Soft Margin Formulation and Kernel Trick.* Retrieved from Towards Data Science: https://towardsdatascience.com/support-vector-machines-soft-margin-formulation-and-kernel-trick-4c9729dc8efe

Munzel, A. (2016). Assisting consumers in detecting fake reviews: The role of identity information disclosure and consensus. *Journal of Retailing and Consumer Services*, 96-108.

Oak, R., & Shafiq, Z. (2021). The Fault in the Stars: Understanding Underground Incentivized Review Services. *ArXiv preprint arXiv:2102.04217*.

Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011). Finding Deceptive Opinion Spam by Any Stretch of the Imagination. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 309–319.

Park, D., Lee, J., & Han, I. (2007). The Effect of On-Line Consumer Reviews on Consumer Purchasing Intention: The Moderating Role of Involvement. *International Journal of Electronic Commerce*, 125-148.

Plotkina, D., Munzel, A., & Pallud , J. (2020). Illusions of truth—Experimental insights into human and algorithmic detections of fake online reviews . *Journal of Business Research*, 511-523.

Porter, M. F. (1980). An Algorithm for Suffix Stripping. *Program: electronic library and information systems*, 130–137.

Rish, I. (2001). An Empirical Study of the Naïve Bayes Classifier. *IJCAI 2001 workshop on empirical methods in artificial intelligence*, 41-46.

Salminen, J., Kandpal, C., Kamel, A., Jung, S.-g., & Jansen, B. J. (2022). Creating and detecting fake reviews of online products. *Journal of Retailing and Consumer Services*.

Sandulescu, V., & Ester, M. (2015). Detecting singleton review spammers using semantic similarity. *Proceedings of the 24th International Conference on World Wide Web*, 971-976.

Schinler, R., & Bickart, B. (2005). Published Word of Mouth: Referable, Consumer-Generated Information on the Internet . *Online consumer psychology: Understanding and influencing consumer behavior in the virtual world*, 35-61.

Smith, A., & Anderson, M. (2016). Online Reviews and Ratings. *Pew Research Center*.

Thakur, R. (2018). Customer engagement and online reviews. *Journal of Retailing and Consumer Services*, 48-59.

Wang, L., Chu, F., & Xie, W. (2007). Accurate cancer classification using expressions of very few genes. *Trans Comput Biol Bioinforma*, 40–53.

Wu, Y., Ngai, E., Wu, P., & Wu, C. (2020). Fake online reviews: Literature review, synthesis, and directions for future research. *Decision Support Systems*.

Yao, Y., Viswanath, B., Cryan, J., Zheng, H., & Zhao, B. Y. (2017). Automated Crowdturfing Attacks and Defenses in Online Review Systems. *In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 1143–1158.

Yoo, K.-H., & Gretzel, U. (2009). Comparison of deceptive and truthful travel reviews. *Information and Communication Technologies in Tourism*, 37-47.