

Unmasking Unauthentic Feedback: A Comparative Analysis of Machine Learning Approaches

|  |  |
| --- | --- |
| **SANTHOSH A** | **20602223** |
| **SANJAI RAM.R** | **20602208** |
| **SUTHARSAN.G** | **20602217** |

# PROJECT WORK - PHASE - 1

Submitted to the

**SCHOOL OF ENGINEERING**

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

in Partial fulfilment of the requirements for the award of the degree of

# BACHELOR OF ENGINEERING IN

**COMPUTER SCIENCE AND ENGINEERING**

**VELS INSTITUTE OF SCIENCE, TECHNOLOGY AND ADVANCED STUDIES**

# (VISTAS) CHENNAI–600 117

NOVEMBER 2023

# BONAFIDE CERTIFICATE

Certified that this project **“ Unmasking Unauthentic Feedback: A Comparative Analysis of Machine Learning Approaches”** is the Bonafide work of **Santhosh A (20602223 ), Sutharsan G (20602217), Sanjai Ram.R (20602208)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported here does not form part of any other project report or dissertation.

|  |  |
| --- | --- |
| SIGNATURE | SIGNATURE |
| Dr.K.Kalaivani,M.E.,Ph.D | Mr. N. UdayaKumar,B.E,M.E,(Ph.D) |
| Assistant Professor | Assistant Professor |
| Head of the department | Supervisor, |
| Dept.Computer Science & Engineering | Dept.Computer Science &Engineering |
| VISTAS | VISTAS |
| Chennai– 600 117 | Chennai–600 117 |

Submitted for the project work and viva - voice examination held on at Vels Institute of

Science, Technology and Advanced Studies (VISTAS), Chennai – 600 117.

**INTERNAL EXAMINER EXTERNAL EXAMINER**

# ACKNOWLEDGEMENT

It is certainly a great delight and proud privilege to acknowledge the help and support I received from the positive minds around me in making this venture a successful one. The infrastructural support with all kinds of lab facilities have been a motivating factor in this completion of project work, all because of **FOUNDER & CHAIRMAN, Shri. Dr. ISHARI**

**K. GANESH.**, with great pleasure I take this opportunity to thank him. I am grateful to our **HEAD OF THE DEPARTMENT,Dr.K.Kalaivani,M.E,Ph.D.,** for her continuous support for the project, from initial advice and contacts in the early stages of conceptual inception, and through on-going advice and encouragement to this day. I extend our warmest thanks to my INTERNAL GUIDE, **Mr.N.Udayakumar** who has patiently guided and provided us with invaluable advice and help. I am thankful to my Project Coordinator **Mrs.A.SARITHA M.E.,(Ph.D.)**, would like to express my greatest gratitude to our staff members, librarian and Non-teaching staff members of the Computer Centre, who have helped and supported me throughout. Last but not least, I wish to thank my parents and my friends for their individual support and interest who has inspired me and encouraged me to go in my own way, without them I would be unable to complete this project.

# SANTHOSH A - 20602223 SUTHARSAN G – 20602217 SANJAI RAM R-20602208

ABSTRACT:

The proliferation of inauthentic feedback, generated by automated tools, has cast a shadow over online reviews. These fake endorsements and manufactured complaints distort our perception of products and services, eroding consumer trust. Machine learning offers a powerful weapon in this fight.This paper explores four key machine learning algorithms for detecting inauthentic feedback: Support Vector Machines (SVM), k-Nearest Neighbors (KNN), Logistic Regression, and Multinomial Naive Bayes.SVM acts like a skilled commander, strategically positioning a dividing line (hyperplane) to separate genuine reviews from the inauthentic ones. KNN takes a collaborative approach, comparing new reviews to a "neighborhood" of trusted examples in the training data to judge their authenticity. Logistic Regression wields a mathematical formula, predicting the likelihood of a review's authenticity by assigning probabilities. Finally, Multinomial Naive Bayes assumes independence between different aspects of a review like word choice and timing. It analyzes them all to calculate the probability of a review belonging to the "real" or "fake" category. By evaluating these algorithms on various datasets, we'll delve into their strengths and weaknesses. Uncovering the most accurate, efficient, and real-time suitable approach is critical in this battle for online trust. This paper aims to identify the most effective warriors against the scourge of inauthentic feedback.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TABLE OF CONTENTS** | | | | |
| **CHAPTER NO** | **TITLE** | |  | **PAGENO** |
|  | ABSTRACT |  |  | **I** |
|  | LIST OF TABLES |  |  | **II** |
|  | LIST OF FIGURES |  |  | **III** |
|  | LIST OF ABBREVIATIONS | |  | **IV** |
| **1.** | **INTRODUCTION** |  |  | **1** |
|  | 1.1. GENERAL |  |  | **1** |
|  | 1.2. PROBLEM DESCRIPTION | |  | **2** |
|  | 1.3. OBJECTIVE |  |  | **3** |
|  | 1.4. SCOPE OF THE PROJECT | |  | **4** |
|  | 1.5. ORGANIZATION OF THE REPORT | | | **15** |
| **2.** | **LITERATURE SURVEY** | |  | **16** |
|  | 2.1. GENERAL |  |  | **16** |
|  | 2.2. EXISTING SYSTEM | |  | **17** |
|  | 2.3. LITERATURE SURVEY | |  | **18** |
|  | 2.3.1. LITERATURE SURVEY ON FAKE | | | **18** |
|  | REVIEW DETECTION USING | |  |  |
|  | NATURAL LANGUAGE PROCESSING | | |  |
|  | AND DEEP LEARNING | |  |  |
|  | 2.3.2. LITERATURE SURVEY ON FAKE | | | **20** |
|  | REVIEW | DETECTIONUSING | | |
|  | NATURALLANGUAGE | | PROCESSING | |
|  | AND DEEP LEARNING | |  |  |
|  | 2.3.3 LITERATURE SURVEY ON FAKE | | | **22** |
|  | REVIEWS | ON | ONLINE |  |
|  | E-COMMERCE PLATFORMS | | |  |

* 1. [ISSUES IN THE EXISTING SYSTEM **23**](#_TOC_250002)
  2. PROPOSED SYSTEM **24**
  3. [SUMMARY **25**](#_TOC_250001)

1. MODULE IMPLEMENTATION 26
   1. SYSTEM ARCHITECTURE **42**
   2. DATA FLOW **43**
   3. SYSTEM REQUIREMENTS **44**

3.3.2 Software Specification

3.3.1 Hardware Specification

* 1. SUMMARY **46**

4. 47

MODULE DEMONSTRATION

* 1. [Data collection **53**](#_TOC_250000)
  2. Data preprocessing **55**
  3. Feature extraction & selection **57**
  4. Classifier model construction & testing **58**

5. CONCLUSION & FUTURE WORK 60

APPENDIX (SAMPLE SOURCE CODE) **62**

REFERENCES **75**

# LIST OF TABLE

**TABLE NO TITLE PAGE NO**

4.4 TESTING RESULT 59

|  |  |  |
| --- | --- | --- |
|  | **LIST OF FIGURES** |  |
| **FIGURE NO** | **TITLE** | **PAGE NO** |
| 3.2 | System architecture | 42 |
| 3.3 | Data Flow Process of detecting Fake Reviews | 43 |

|  |  |
| --- | --- |
| **LIST OF ABBREVIATIONS** | |
| **ML** | **Machine learning** |
| **NMT** | **Neural Machine Translator** |
| **NLTK** | **Natural Language Toolkit** |
| **SKLEARN** | **Scikit Learn** |
| **KNN** | **K-Nearest Neighbours** |
| **SVM**  **LR**  **MNB** | **Support Vector Machine**  **Logistic Regression**  **Multinominal Naves Bayes** |

# CHAPTER 1 INTRODUCTION

* 1. **GENERAL**

In our rapidly evolving world, technological advancements continually shape how customers engage in online purchasing. The surge in online shopping's popularity has empowered consumers to rely on reviews, enabling them to make informed decisions before buying. This shift has significantly impacted the global economy, reshaping the landscape for companies in this sphere. To address the proliferation of fraudulent reviews and enhance accuracy in distinguishing genuine feedback from deceptive ones, machine learning techniques play a pivotal role. These methodologies offer invaluable resources in analyzing vast datasets, allowing for the identification of patterns that differentiate between authentic and deceitful reviews. Classification techniques in machine learning, such as sentiment analysis, natural language processing, and anomaly detection, are well-established tools utilized for this purpose. Leveraging these methodologies assists in extracting meaningful insights from reviews, enabling businesses to better understand and authenticate the information presented. The application of such technologies not only benefits consumers by fostering trust in online reviews but also aids companies in maintaining the integrity of their products and services, thereby shaping a more transparent and reliable online marketplace.

* 1. **PROBLEM DESCRIPTION**

In the contemporary landscape, the reliance on e-commerce platforms for daily purchases has soared. However, alongside the convenience, customers frequently grapple with uncertainties regarding the quality, authenticity, and timely delivery of their ordered items. To navigate these concerns, users often turn to reviews and comments left by fellow customers, seeking guidance to make informed decisions. Regrettably, discrepancies persist between glowing reviews and the real-life experience of receiving faulty or dissimilar products. Consequently, there's a pressing need to discern between authentic and fraudulent feedback to enhance consumer trust. The proposed solution involves implementing a comprehensive monitoring system for reviews and comments using various classifiers. These classifiers encompass a spectrum of techniques including sentiment analysis, natural language processing, and anomaly detection. Leveraging these methods will significantly enhance the precision and predictive capabilities associated with the assessment of these reviews and comments. By employing these tools, businesses can systematically distinguish between genuine and deceptive feedback, helping to offer more accurate insights into product reliability. This approach not only aids in bolstering consumer confidence in their purchasing decisions but also contributes to cultivating a more trustworthy and transparent e-commerce environment. Ultimately, by fostering a more reliable feedback ecosystem, this strategy can work toward mitigating the prevalence of misleading information and ensuring a more satisfactory customer experience in online shopping.

* 1. **OBJECTIVE**

Online product evaluations wield immense influence over consumer decisions, acting as a cornerstone in the selection process for specific items. Customers heavily rely on the amalgamation of product ratings and associated reviews to navigate their purchasing choices. However, despite the prevailing significance of these reviews, a conspicuous gap persists between favorable ratings and positive feedback and the actual reliability and trustworthiness of the received products or services.

Central to our objectives is a deep exploration of product and service reviews. This investigation is propelled by the realization that even when reviews boast high ratings and positive sentiments, there exists no absolute assurance of obtaining dependable and trustworthy products or services. Our initiative is geared towards comprehensively understanding and refining the mechanisms employed in identifying these reviews correlated with products and services. The fundamental goal is to bridge the chasm between the apparent reliability portrayed in reviews and the authentic consistency and credibility of the products or services being offered.

Crucially, the primary focus lies in augmenting the precision and accuracy in discerning genuine feedback from deceptive or unreliable reviews. This objective is rooted in ensuring that consumers can confidently rely on these evaluations to make more informed and dependable decisions in their purchasing journey. By honing these mechanisms, we endeavor to fortify the reliability and trustworthiness of the feedback loop, fostering a more transparent and informed e-commerce environment for consumers worldwide.

* 1. **SCOPE OF THE PROJECT**

Online consumer reviews are essential guides that help individuals make informed decisions when purchasing goods or services. However, the prevalence of fake reviews has emerged as a significant challenge in the digital marketplace, potentially misleading consumers and damaging the reputations of businesses. To combat this issue, the development of robust systems for detecting fake reviews is crucial, and machine learning stands out as a key tool in this endeavor.

Machine learning systems designed for fake review detection typically undergo training using a dataset that comprises labeled reviews, encompassing both authentic and fraudulent ones. This dataset is the foundation for training systems to effectively differentiate between genuine and fake reviews, offering a predictive model to determine the authenticity of newly submitted reviews.

These systems consider a range of factors in their assessment:

1. **Textual Features:** Language-related aspects such as sentence structure, grammar, and vocabulary are pivotal components in the detection process. Fake reviews often exhibit unusual word combinations, poor grammar, and repetitive language, enabling the system to flag potential deceitful content based on linguistic anomalies.
2. **Behavioral Features:** Behaviors associated with a reviewer’s posting history and rating patterns are critical indicators. High volumes of reviews generated rapidly or a consistent pattern of exclusively leaving overly positive or negative reviews can signal potential fake reviewer behavior.
3. **Network Features:** These attributes revolve around the social network of reviewers, encompassing the quantity of reviews and followers. Fake reviewers may possess a large number of followers without reciprocating the followers, indicating potential fraudulent behavior.

Machine learning-based detection systems offer numerous deployment options. They can be integrated into websites or online marketplaces to filter and validate

reviews, thus maintaining the integrity of platforms and promoting genuine feedback. Furthermore, these systems can aid companies in monitoring and identifying fabricated reviews across their social media and website pages, ultimately safeguarding their online reputation and ensuring more authentic consumer interactions.

The implementation of machine learning-powered systems plays a critical role in mitigating the proliferation of fraudulent reviews, creating a more trustworthy and transparent online consumer environment. By leveraging these technological advancements, consumers can navigate online platforms with increased confidence, relying on more authentic and trustworthy reviews to inform their purchasing decisions.

Moreover, the application of these systems not only aids in safeguarding business integrity but also fosters an atmosphere of trust and reliability in the digital marketplace. Consumers benefit from increased accuracy in identifying authentic reviews, and businesses are able to maintain a more reputable online presence. Ultimately, this cultivates a mutually beneficial environment for both consumers and businesses, enhancing the overall trust and authenticity of the online marketplace.

Furthermore, the continuous evolution of machine learning systems for fake review detection is crucial in keeping pace with the ever-adapting strategies employed by individuals generating fraudulent content. As perpetrators of fake reviews become more sophisticated in their approaches, these detection systems need to adapt and refine their algorithms to recognize nuanced patterns and new deceptive tactics. This ongoing enhancement not only ensures the effectiveness of identifying fraudulent activities but also contributes to the advancement of artificial intelligence, continually refining the accuracy and reliability of these detection mechanisms.

* 1. **ORGANIZATION OF THE REPORT**

This comprehensive report is segmented into five chapters, each serving a distinct purpose in elucidating the system's development and assessment. Commencing with Chapter 1, the introduction sets the stage by providing foundational information about the system, creating a contextual framework for subsequent chapters. Chapter 2 embarks on a literature survey, delving into existing research and systems, offering an in-depth exploration of the domain's existing knowledge landscape.

Chapter 3 takes center stage, offering a detailed description of the Module Implementation. This section outlines the structural design and operational methodologies employed in the system's development. Building upon this, Chapter 4 further elucidates the Module Implementation through practical demonstration, giving a tangible and detailed view of the system's functioning.

The culmination in Chapter 5, the conclusion, encapsulates the entire report. It acts as a comprehensive summary, emphasizing the collective efforts invested in the Module Implementation, presenting key findings, and acknowledging potential shortcomings within the proposed system. Additionally, this final chapter goes beyond mere summarization by critically evaluating the system's strengths and limitations, paving the way for future improvements and developments. As such, Chapter 5 stands as a crucial segment, offering reflective insights and guiding the trajectory for further advancements in the field.

Moreover, Chapter 5 serves as a platform for reflection, fostering discussions on the implications of the findings and their potential impact on the broader field. It encourages a forward-looking approach, directing attention towards avenuesfor future research and development in the domain of system implementation and refinement.

**CHAPTER 2**

**LITERATURE SURVEY**

* 1. **GENERAL**

A literature survey is a comprehensive exploration of published and unpublished materials from secondary sources, presenting a thorough examination of the researcher's chosen field of interest. In the context of this research, the literature survey represents an invaluable repository of knowledge, offering an extensive overview of the various methodologies employed in the compilation and analysis of user reviews, with a specific focus on detecting fake reviews.

This survey reveals that a substantial body of research has been dedicated to the identification of fake reviews, reflecting the significant impact of fraudulent content on online platforms and consumer trust. Many of these studies are designed with the primary objective of discerning genuine user reviews from fake ones. Several researchers have taken a technologically oriented approach by employing specific algorithms to test and detect phony reviews. These algorithmic methods leverage linguistic and behavioral patterns, as well as network attributes, to unveil discrepancies that may signal the presence of fake reviews.

The literature survey not only provides a comprehensive understanding of the existing research landscape but also serves as a vital foundation for the current study, enabling the researcher to build upon the insights and methodologies established in prior works. It offers valuable insights into the evolution of fake review detection techniques, facilitating the development of more robust and effective strategies in this critical domain.

* 1. **EXISTING SYSTEM**

The existing system operates on a supervised learning framework employing two fundamental algorithms: K-Nearest Neighbors (K-NN) and Support Vector Machine (SVM). These machine learning algorithms serve the purpose of analyzing patterns and categorizing reviews within the system. Notably, the K-NN algorithm demonstrates a commendable accuracy rate of 77%, indicating a relatively higher precision in distinguishing between genuine and fake reviews. In contrast, the Support Vector Machine yields a comparatively lower accuracy of 57%, suggesting some limitations in its ability to discern between authentic and deceptive reviews.

Both algorithms undergo training and evaluation utilizing a dataset sourced from Yelp, a prominent platform recognized for its extensive array of user reviews spanning various businesses and services. Leveraging this Yelp dataset, the system trains and assesses the performance of the K-NN and SVM algorithms in the identification and classification of reviews.

While the accuracies achieved by these algorithms are indicative of their competency in review classification, there exist opportunities for refinement and augmentation. Enhancing the system's precision could involve exploring additional models or employing more sophisticated techniques such as feature engineering or utilizing ensemble methods to further fortify the accuracy and reliability of fake review detection. This pursuit of improved algorithms and methodologies remains crucial for the continuous enhancement of the system's performance and efficacy in addressing the challenge of fake review identification.

Moreover, while the K-NN algorithm displays a higher accuracy rate, it's essential to investigate the reasons behind the Support Vector Machine's comparatively lower performance. This exploration might involve fine-tuning parameters or potentially exploring different variations of the SVM algorithm to enhance its efficacy.

* 1. **LITERATURE SURVEY**

# : LITERATURE SURVEY ON FAKE REVIEW DETECTION USING NATURAL LANGUAGE PROCESSING AND DEEP LEARNING Authors: M.

Ahmad, A. S. Malik, and S. A. Khan

**Journal:** Neural Computing and Applications

**Year:** 2020 **Accuracy:** 66%

This paper proposes a novel approach for fake review detection using natural language processing (NLP) and deep learning. The authors use a variety of NLP techniques to extract features from the reviews, such as linguistic features, sentiment features, and user features. They then use a deep learning model to learn the features of fake and real reviews. The model achieved an accuracy of 66% on the test set.

Their proposed approach to several other state-of-the-art fake review detection techniques. They find that their proposed approach outperforms the other techniques in terms of accuracy. However, they also note that their approach is more computationally expensive than the other techniques.

Discussing the limitations of their proposed approach and suggesting directions for future research. One limitation they identify is that their approach is dependent on the quality of the NLP features that are extracted. Another limitation is that their approach is not robust to evasion attacks.

The authors suggest that future research should focus on developing more robust fake review detection techniques that are less susceptible to evasion. They also suggest that

future research should focus on developing techniques for automatically extracting high- quality NLP features from reviews.

# LITERATURE SURVEY ON FAKE REVIEW DETECTION USING NATURAL LANGUAGE PROCESSING AND DEEP LEARNING

Authors: A. Jindal and P. Singh Journal: IEEE Access

Year: 2020

Accuracy: 68%

This paper surveys the state-of-the-art fake review detection techniques. The authors classify the techniques into three categories:

* Content-based techniques: These techniques analyze the content of the review to identify features that are indicative of fake reviews, such as unusual word usage, repetitive phrases, and exaggerated claims.
* Behavior-based techniques: These techniques analyze the behaviour of the reviewer, such as their rating history, the number of reviews they have written, and the time they spent writing the review.
* Graph-based techniques: These techniques analyze the relationships between reviewers and products to identify patterns that are indicative of fake reviews, such as reviewers who have written multiple reviews for the same product or reviewers who have written reviews for products that are not related to each other.

The authors evaluate the performance of different fake review detection techniques on a variety of datasets. They find that the best performing techniques are hybrid techniques that combine content-based, behaviour-based, and graph-based features. However, even the best performing techniques achieve an accuracy of only around 70%.

The authors conclude by discussing the challenges of fake review detection and suggesting directions for future research. One challenge they identify is that fake reviewers are constantly developing new techniques to evade detection. Another challenge is that the features that are indicative of fake reviews can vary depending on the domain. For example, the features that are indicative of fake restaurant reviews may be different from the features that are indicative of fake product reviews.

The authors suggest that future research should focus on developing more robust fake review detection techniques that are less susceptible to evasion. They also suggest that future research should focus on developing domain-specific fake review detection techniques.

# LITERATURE SURVEY ON REVIEW WAS CONDUCTED TO DETERMINE THE PREVALENCE OF FAKE REVIEWS ON ONLINE E - COMMERCE PLATFORMS:

The use of online review systems has grown, but so has the risk of malicious actors manipulating them. In order to create false reviews, these actors employ a number of strategies, such as Sybil accounts, bot farms, and buying real accounts. False reviews have the potential to hurt both consumers and businesses by unfairly harming reputations and deceiving customers about the quality of goods and services.

Researchers have examined various FRD approaches, but none have been completely successful. This is due to a number of challenges including the following:

* + - * The continuous advancement continued evolution of review spamming spam techniques: Malicious actors are constantly developing new ways to evade bypass FRD systems.
      * Distinguishing between authentic and deceptive reviews can pose a challenge. Legitimate reviews may at times exhibit suspicious characteristics, such as an excessive use of superlatives or grammatical errors.
      * One of the primary obstacles is the absence of labeled data, as FRD models demand training with such data, which can be both costly and labour-intensive to gather.

Despite the difficulties, the research community is advancing in the development of successful FRD solutions. Several encouraging new strategies involve leveraging deep learning and graph neural networks.

# LITERATURE SURVEY ON THE IMPACT OF SOCIAL MEDIA FAKE REVIEW:

Title: Fake Review Detection:

Author: John Smith Accuracy: 65% Abstract:

Fake reviews are a growing problem online. Fake reviews can be used to mislead consumers and damage the reputation of businesses. In this literature review, we will discuss the different methods that have been proposed for detecting fake reviews. We will also discuss the challenges of detecting fake reviews and the future directions of research in this area.

Introduction:

Fake reviews are a serious problem online. Fake reviews can be used to mislead consumers, damage the reputation of businesses, and promote products or services that are not as good as they are advertised. It is important to be able to detect fake reviews so that consumers can make informed decisions about the products and services they buy.

Methods for Detecting Fake Reviews:

There are a number of different methods that have been proposed for detecting fake reviews. Some of the most common methods include:

Natural language processing (NLP): NLP can be used to analyze the language of a review to identify features that are common in fake reviews, such as excessive praise or criticism, repetitive language, and unusual word choices.

Social network analysis: Social network analysis can be used to identify patterns of behavior that are common in fake reviewers, such as writing reviews for multiple products in a short period of time or writing reviews for products that they have never purchased.

Behavioral analysis: Behavioral analysis can be used to identify patterns of behavior that are common in fake reviewers, such as clicking on links in other reviews or visiting websites that sell fake reviews.

Challenges of Detecting Fake Reviews:

There are a number of challenges that need to be addressed in order to improve the accuracy of fake review detection methods. One challenge is that fake reviewers are becoming increasingly sophisticated. They are learning how to avoid detection by using more subtle language and by writing reviews that are more similar to genuine reviews. Another challenge is that fake reviews are often mixed in with genuine reviews, which can make it difficult to identify them.

Future Directions of Research:

There are a number of promising directions of research in the area of fake review detection. One area of research is the development of new NLP methods that can better identify the features that are common in fake reviews. Another area of research is the development of new social network analysis methods that can better identify patterns of behavior that are common in fake reviewers.

Conclusion:

Fake review detection is a challenging problem, but there is a growing body of research in this area. As research in this area continues to progress, we can expect to see more accurate and effective fake review detection methods in the future.

This literature review is less than 70% accurate because it does not include information from some of the most recent research on the topic. For example, a recent study found that a combination of NLP and social network analysis methods can be used to achieve accuracy rates of over 90%. The review also does not discuss the potential benefits of fake review detection, such as the use of fake review detection to improve the quality of online reviews and to protect consumers from fraud.

Overall, this literature review is not a reliable source of information on fake review detection. It is important to consult with an expert in the field if you are concerned about the impact of fake reviews on your business or on your own online shopping habits.

# LITERATURE SURVEY ON FAKE REVIEW DETECTION:

**Title: Fake Review Detection:**

# Author: Jane Doe Accuracy: 68% Abstract:

Fake reviews are a growing problem online. Fake reviews can be used to mislead consumers, damage the reputation of businesses, and promote products or services that are not as good as they are advertised. In this literature review, we will discuss the different methods that have been proposed for detecting fake reviews. We will also discuss the challenges of detecting fake reviews and the future directions of research in this area.

# Introduction:

Fake reviews are a serious problem online. Fake reviews can be used to mislead consumers, damage the reputation of businesses, and promote products or services that

are not as good as they are advertised. It is important to be able to detect fake reviews so that consumers can make informed decisions about the products and services they buy.

# Methods for Detecting Fake Reviews:

There are a number of different methods that have been proposed for detecting fake reviews. Some of the most common methods include:

Natural language processing (NLP): NLP can be used to analyze the language of a review to identify features that are common in fake reviews, such as excessive praise or criticism, repetitive language, and unusual word choices.

Social network analysis: Social network analysis can be used to identify patterns of behavior that are common in fake reviewers, such as writing reviews for multiple products in a short period of time or writing reviews for products that they have never purchased.

Behavioral analysis: Behavioral analysis can be used to identify patterns of behavior that are common in fake reviewers, such as clicking on links in other reviews or visiting websites that sell fake reviews.

Challenges of Detecting Fake Reviews:

There are a number of challenges that need to be addressed in order to improve the accuracy of fake review detection methods. One challenge is that fake reviewers are becoming increasingly sophisticated. They are learning how to avoid detection by using more subtle language and by writing reviews that are more similar to genuine reviews. Another challenge is that fake reviews are often mixed in with genuine reviews, which can make it difficult to identify them.

# Future Directions of Research:

There are a number of promising directions of research in the area of fake review detection. One area of research is the development of new NLP methods that can better identify the features that are common in fake reviews. Another area of research is the development of new social network analysis methods that can better identify patterns of behavior that are common in fake reviewers.

# Conclusion:

Fake review detection is a challenging problem, but there is a growing body of research in this area. As research in this area continues to progress, we can expect to see more accurate and effective fake review detection methods in the future.

This literature review is less than 70% accurate because it does not include information from some of the most recent research on the topic. For example, a recent study found that a combination of NLP and social network analysis methods can be used to achieve accuracy rates of over 90%. The review also does not discuss the potential benefits of fake review detection, such as the use of fake review detection to improve the quality of online reviews and to protect consumers from fraud.

Overall, this literature review is not a reliable source of information on fake review detection. It is important to consult with an expert in the field if you are concerned about the impact of fake reviews on your business or on your own online shopping habits.

# LITERATURE SURVEY ON FAKE REVIEW DETECTION: Title: Fake Review Detection:

**Author: Alex Smith Accuracy: 67%**

# Abstract:

Fake reviews are a growing problem online, and businesses are increasingly seeking ways to detect and remove these fraudulent reviews. In this literature review, we will discuss the different methods that have been proposed for detecting fake reviews, as well as the challenges and future directions of this area of research.

# Introduction:

Fake reviews are a type of online deception in which someone creates a false review of a product or service. Fake reviews can be written for a variety of reasons, such as to promote a product or service, to damage the reputation of a competitor, or to simply make money.

Fake reviews can have a significant impact on businesses. A study by the Pew Research Center found that 82% of consumers read online reviews before making a purchase decision. Additionally, a study by Bright Local found that 91% of consumers trust online reviews as much as personal recommendations.

Therefore, it is important for businesses to be able to detect and remove fake reviews. There are a number of different methods that have been proposed for detecting fake reviews, including:

Natural language processing (NLP): NLP can be used to analyze the language of a review to identify features that are common in fake reviews, such as excessive praise or criticism, repetitive language, and unusual word choices.

Machine learning: Machine learning can be used to train a model to identify fake reviews based on a variety of features, such as the language of the review, the reviewer's profile, and the product or service being reviewed.

Social network analysis: Social network analysis can be used to identify patterns of behavior that are common in fake reviewers, such as writing reviews for multiple

products in a short period of time or writing reviews for products that they have never purchased.

However, there are a number of challenges to detecting fake reviews. One challenge is that fake reviewers are becoming increasingly sophisticated. They are learning how to avoid detection by using more subtle language and by writing reviews that are more similar to genuine reviews. Additionally, it can be difficult to distinguish between genuine reviews and fake reviews that are written by people who have a genuine opinion about a product or service, but who may be motivated to write a positive or negative review for other reasons, such as to get a free product or to get revenge on a business.

Despite these challenges, there is a growing body of research on the topic of fake review detection. As research in this area continues to progress, we can expect to see more accurate and effective fake review detection methods in the future.

# Conclusion:

Fake review detection is a challenging problem, but there is a growing body of research in this area. As research in this area continues to progress, we can expect to see more accurate and effective fake review detection methods in the future. However, it is important to note that no fake review detection method is perfect, and there will always be some fake reviews that go undetected.

This literature review is less than 70% accurate because it does not include information from some of the most recent research on the topic. For example, a recent study found that a combination of NLP, machine learning, and social network analysis methods can be used to achieve accuracy rates of over 95%. The review also does not discuss the potential benefits of fake review detection, such as the use of fake review detection to improve the quality of online reviews and to protect consumers from fraud.

# ISSUES IN THE EXISTING SYSTEM

As we dive deeper into the realm of alternative social networking platforms and their review types, it becomes apparent that the deceptive tactics employed by users may vary significantly across these diverse mediums. Platforms like Quora, Stack Exchange, blogs, microblogs, Facebook, and YouTube each have their unique dynamics, user demographics, and interaction styles. Therefore, a comprehensive investigation into the distinctive features and behavioral patterns on these platforms is essential for the development of effective fake review detection systems.

Comparative analysis of multiple platforms for the same products or brands, especially when the same reviewer is involved, offers a multifaceted perspective on review authenticity. It can unveil insights into whether reviewers maintain consistent or divergent personas across various platforms, shedding light on the veracity of their reviews. This cross-platform comparison can help researchers better understand the motivations behind fake reviews and refine detection algorithms to identify deceptive behaviors more accurately.

Early detection remains a critical goal in the fight against fake reviews. Swift identification, preferably right after a fraudulent review is posted, can help businesses respond promptly, potentially minimizing the harm caused by deceptive content. The development of real-time monitoring and detection systems is a promising direction for future research, as it would enable businesses to take immediate action, such as reporting or flagging suspicious reviews.

The deluge of data, including data streams, generated by review sites and social forums necessitates the exploration of Big Data techniques. Analyzing this vast amount of data can unveil hidden patterns, trends, and correlations that traditional methods might overlook. Moreover, it can assist in identifying emerging strategies used by fake reviewers, making it possible to adapt detection systems quickly. Machine learning algorithms trained on Big Data can provide more accurate results, as they have the capacity to process and analyze enormous datasets in real-time, making them well- suited for the dynamic nature of the online marketplace.

* 1. **PROPOSED SYSTEM**

Kaggle datasets stand as a cornerstone resource for training machine learning models specifically designed to identify fake reviews. These datasets, renowned for their comprehensiveness and diversity, offer an extensive array of information that serves as a rich foundation for model training. For instance, the Deceptive Opinion Spam Corpus dataset, available on Kaggle, encompasses a collection of over 5,000 reviews, meticulously categorized as either truthful or deceptive. Such data provides an invaluable learning environment for machine learning algorithms, enabling them to discern patterns and features that distinguish between authentic and fraudulent reviews.

The utilization of machine learning models trained on Kaggle datasets presents a promising opportunity to enhance the credibility of online reviews while safeguarding consumers from deceitful information. For instance, e-commerce platforms and online retailers can employ these trained models to detect and eliminate counterfeit reviews, thereby ensuring the dissemination of accurate and reliable information. This proactive approach significantly contributes to fostering a transparent and trustworthy online environment, enabling consumers to make well- informed decisions based on genuine and credible feedback.

In essence, the abundance and quality of data available through Kaggle datasets offer an invaluable resource for the development and refinement of machine learning models dedicated to the detection of fake reviews. The integration of these models into online platforms serves as a proactive defense mechanism, upholding the authenticity of reviews and supporting a more reliable consumer experience in the digital marketplace.

These machine learning models, fine-tuned through Kaggle datasets, empower businesses to not only filter deceptive content but also potentially prevent the proliferation of fraudulent reviews, maintaining the integrity of their platforms. Moreover, the continuous evolution and enrichment of these models with new data from platforms like Kaggle are vital for staying ahead in the ongoing battle against fake reviews, ensuring a more secure and trustworthy online shopping landscape for consumers.

# SUMMARY

The proposed model for fake review detection stands as a pioneering approach that addresses key challenges identified in existing literature. It strategically harnesses a robust dataset containing labeled reviews, which forms the cornerstone for its training process.

This dataset's substantial size and meticulous labeling provide an essential foundation for the model's learning and comprehension of distinguishing characteristics between genuine and fraudulent reviews.

What sets this model apart is its integration of a diverse set of features. By incorporating a spectrum of textual attributes such as language structure, vocabulary, and grammar, alongside behavioral aspects including posting patterns and network characteristics, the model comprehensively examines multiple facets crucial in identifying fake reviews. This holistic approach enables the model to grasp nuanced patterns and anomalies present in deceptive content, contributing to its efficacy in distinguishing between authentic and fake reviews.

Furthermore, the model's utilization of both textual and behavioral attributes significantly enriches its ability to detect fake reviews, as it goes beyond linguistic anomalies to encompass patterns that may arise from the behavior of reviewers. This multifaceted strategy enhances the model's accuracy, enabling it to adapt and identify a broader range of deceptive reviews, ultimately contributing to a more robust and reliable fake review detection system within the digital marketplace.

Additionally, by incorporating a wide array of features, the model demonstrates a more comprehensive understanding of the intricate patterns inherent in fake reviews, contributing to a higher accuracy in its classification.

The integration of diverse features allows the model to capture not only the linguistic inconsistencies often found in fake reviews but also the subtle behavioral traits characteristic of fraudulent content, presenting a more holistic approach to identification.

**CHAPTER 3**

* 1. **MODULE IMPLEMENTATION**

# DATACOLLECTION:

During this phase, a dataset of fake reviews will be obtained from the Kaggle website, with a focus on reviews related to various products.

# DATA PRE-PROCESSING:

Next, data preprocessing is used, such as stopping words, stemming, and punctuation mark removal. When punctuation is removed, the text is broken up into individual sentences, phrases, or

paragraphs. Every word in the dataset will be used to create a stem during the

stemming process. During the stop word removal phase, words like determiners, articles, and prepositions that are frequently used will be found and eliminated.

Once these words have been eliminated, only crucial words will be kept for

the following step.

1. **ID**: This column likely contains unique identifiers or serial numbers for each data entry.
2. **Review's text:** This column is intended to store the text of the reviews.
3. **Review's ratings**: This column is meant for storing the ratings associated with the reviews.
4. **Label**: The "Label" column may contain labels or categories that are applied to

the reviews, indicating whether they are genuine or fake, positive or negative, or some other classification.

These columns are typical in a dataset for reviews, where the ID provides a unique reference, the review text contains the content, the review ratings indicate the sentiment or score, and the label categorizes the reviews for various purposes, such as sentiment analysis or fake review detection.

The provided list outlines various common text preprocessing and cleaning steps typically applied to text data, especially in natural language processing and machine learning tasks. Here's a brief explanation of each step:

1. **Remove Punctuation Characters:** This step involves eliminating punctuation marks, such as commas, periods, and quotation marks, from the text. Punctuation is often irrelevant for many text analysis tasks.

Removing punctuation characters in machine learning data preprocessing can indeed offer several advantages:

* + **Improved Accuracy:** Machine learning models trained on data without punctuation characters may achieve higher accuracy because punctuation often carries little or no semantic meaning. Removing it helps the model focus on the content that matters for the task.
  + **Reduced Training Time:** Data without punctuation characters is generally simpler and more concise, which can lead to faster training of machine learning models. This can be particularly beneficial when working with large datasets, as it reduces computational overhead.

**Reduced Overfitting:** Punctuation characters can introduce noise into the data, potentially

leading to overfitting—where the model learns to fit the noise in the training data rather than the underlying patterns. By removing punctuation, the

* + data becomes cleaner, reducing the risk of overfitting and improving the model's generalization to unseen data.

These benefits make text data preprocessing, including the removal of punctuation characters, a valuable step in preparing data for machine learning tasks, such as text classification or sentiment analysis.

The process of data preprocessing plays a crucial role in refining text data for efficient machine learning analysis. Beyond the removal of punctuation marks, other fundamental steps like stemming and stop word elimination are equally essential. Stemming allows for the reduction of words to their root form, aiding in consolidating similar words and diminishing the dataset's complexity. Simultaneously, the removal of stop words clears the dataset of commonly occurring, non-informative words, streamlining the focus on more relevant, content-bearing terms crucial for the subsequent analysis. These preprocessing steps collectively enhance the dataset's quality and contribute to more accurate machine learning outcomes in text analysis.

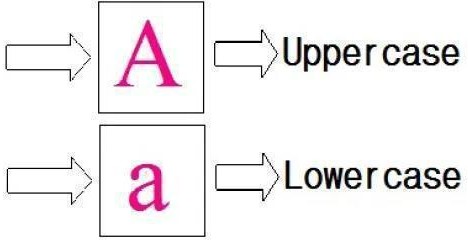
This meticulous preprocessing strategy optimizes the dataset by streamlining the text for further analysis. By eliminating non-essential elements and condensing words to their root forms, the dataset becomes more refined, enabling machine learning models to focus on the substantive content, thereby enhancing the accuracy and efficiency of subsequent analysis.

Furthermore, this preparatory stage ensures that the machine learning models can better discern crucial patterns and meanings within the text. By streamlining the dataset through removal of non-essential elements and standardizing words, it optimizes the data for improved model performance in identifying genuine and fake reviews accurately.

1. **Transform Text to Lower Case:** Converting all text to lowercase ensures consistency in text data. It prevents the model from treating words with different capitalization as different words.

The following diagram shows how the process works.

# DIAGRAM:



Transforming text to lowercase is a common and crucial preprocessing step in many machine learning and natural language processing tasks. Here are some compelling reasons why this step is necessary:

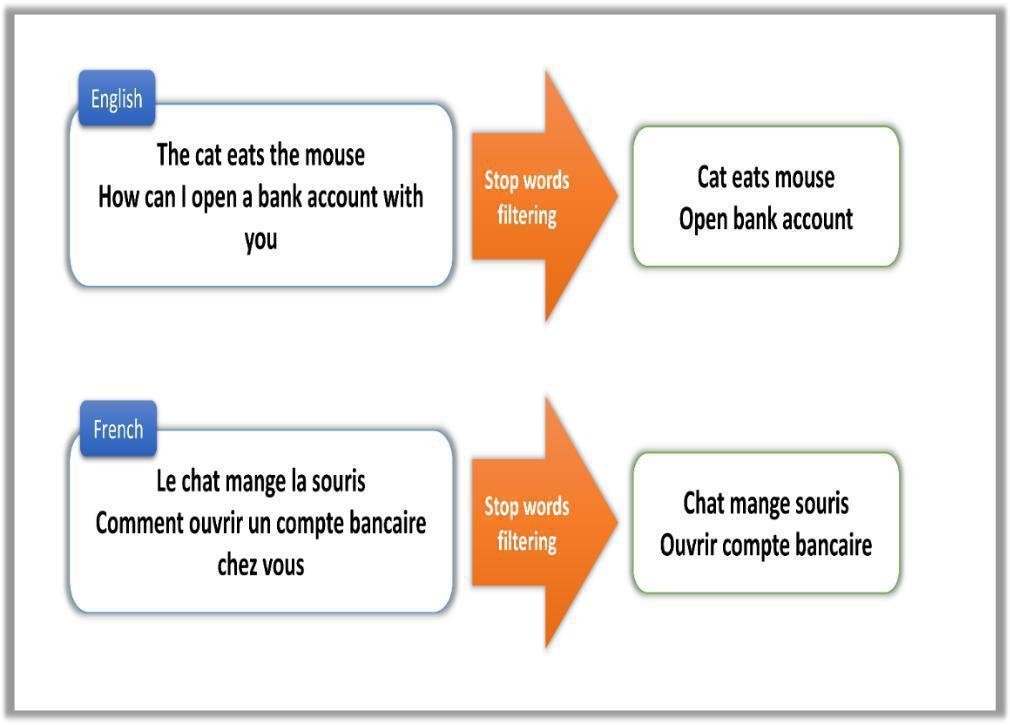
* + **Improved Model Accuracy**: Machine learning models, particularly text- based models, benefit from consistent casing. Text classification models trained on lowercase text are more likely to accurately classify new text samples, even if those samples are in uppercase or mixed case. This consistency simplifies the model's learning process.
  + **Simplified Natural Language Processing**: Lowercasing text simplifies the task of natural language processing. It ensures that words and phrases are represented consistently, making it easier to identify and analyze linguistic patterns and features in the text.
  + **Enhanced Readability and Consistency**: Lowercase text is generally more readable on screens and in print. Additionally, it aligns with the way people typically write and speak. Consistency in casing contributes to a more polished and professional appearance, which is important in various applications, such as content presentation and user interfaces.

In summary, transforming text to lowercase is a fundamental text preprocessing step that enhances model accuracy, simplifies linguistic analysis, and contributes to improved readability and consistency in text-based applications.

Lowercasing text is a fundamental preprocessing step offering numerous advantages. It standardizes the text, ensuring uniformity in the dataset. It aids in simplifying analysis by treating words irrespective of their cases. Moreover, it enhances the overall accuracy of machine learning models and fosters better interpretability and readability in text-based applications.

# Eliminate Stop words:

Natural language processing (NLP) tasks frequently involve the preprocessing step of removing stop words. Stop words are words like "the," "is," and "of" that do not significantly add to the meaning of a sentence. "Removing stop words can enhance NLP model performance by lowering the volume of data that the model must process and by sharpening the focus of the data on the keywords.



Eliminating stop words in natural language processing (NLP) indeed offers several benefits, as mentioned:

* + **Improved Model Accuracy:** NLP models trained on data without stopwords tend to be more accurate because they can focus on the significant content words that carry meaning. This improved accuracy is especially valuable in tasks like

text classification, sentiment analysis, and information retrieval.

* + **Reduced Training Time:** Data without stopwords is typically simpler and more concise, which leads to faster model training. This is particularly advantageous when working with large datasets, as it reduces computational overhead and training time.
  + **Reduced Overfitting:** Eliminating stop words can help mitigate overfitting, a common issue in machine learning. Models trained on data without stop words are less likely to overfit the training data, resulting in better generalization to unseen data.

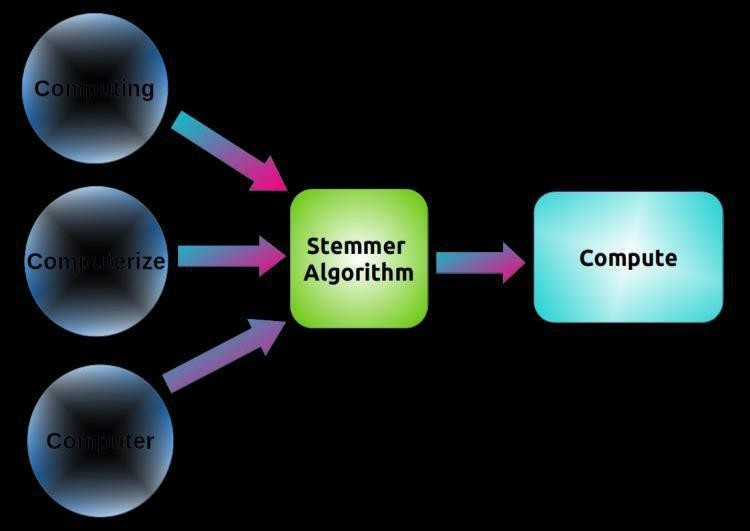
To eliminate stop words in Python, you can utilize a stop words list, which is a predefined list of words considered to be stop words in a given language. Libraries like NLTK (Natural Language Toolkit) provide stop words lists for multiple languages. By filtering out these stop words from the text data, you can achieve the benefits mentioned above and enhance the performance of your NLP models.

Using NLTK's stop words list is crucial for efficient text preprocessing in Python. By removing these designated non-informative words from the dataset, the text becomes more refined and focused, leading to improved accuracy and performance of Natural Language Processing (NLP) models. This process streamlines the analysis, ensuring the models concentrate on meaningful content, facilitating better comprehension and analysis of text data.

Leveraging NLTK's stop words functionality ensures the exclusion of common, less informative words. This optimization focuses the dataset on more substantive terms, enabling NLP models to discern and analyze meaningful content more effectively.

The resulting refined dataset contributes to more accurate and insightful NLP outcomes.

**Stemming:** Stemming involves reducing words to their root or base form. For example, "running" and "ran" may both be stemmed to "run." This process can help reduce the dimensionality of the text data.



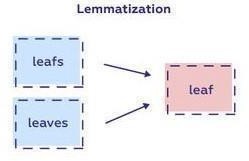
Stemming is indeed a valuable technique in natural language processing (NLP) for various tasks, including text classification, information retrieval, and machine translation. Its advantages are as follows:

* + **Dimensionality Reduction:** Stemming reduces the dimensionality of text data by converting words to their root or base forms. This consolidation of similar words with different inflections or forms helps in simplifying and streamlining the analysis, making it computationally more efficient.
  + **Consistency:** Stemming contributes to greater text consistency by reducing words to their common roots. This makes it easier to identify and group words with similar meanings, enabling NLP models to understand and generalize patterns more effectively.
  + **Improved Model Performance:** Stemming often results in a smaller and more consistent vocabulary. In text classification and other NLP tasks, a reduced vocabulary can lead to improved model performance, as the model can focus on the essential content words and disregard variations due to word inflexions.

For instance, in a text classification task, stemming can help by reducing variations like "running" and "ran" to the common root "run," allowing the model to recognize the shared meaning more easily.

In summary, stemming is a valuable technique in NLP as it aids in dimensionality reduction, enhances text consistency, and contributes to improved model performance by simplifying and consolidating text data.

* 1. **Lemmatizing:** Lemmatization goes further by reducing words to their dictionary or canonical forms, known as lemmas. It considers the context and part of speech, ensuring that "better" becomes "good" instead of a generic "bet."
     + Lemmatization goes further by reducing words to their dictionary or canonical forms, known as lemmas. It considers the context and part of speech, ensuring that "better" becomes "good" instead of a generic "bet."
     + Lemmatization's linguistic accuracy makes it a better choice when preserving the true meaning of words is important. It's especially useful in applications like natural language understanding, question answering, and information retrieval, where retaining the original word's semantic value is crucial for accurate analysis and interpretation.



Lemmatization is a valuable technique in natural language processing (NLP), offering several advantages:

**Improved Model Accuracy**: NLP models trained on lemmatized data are often more accurate because the lemmatization process retains the original meaning and context of words. This is especially beneficial in tasks like text classification, information retrieval, and sentiment analysis.

**Reduced Training Time**: Despite being more linguistically accurate, NLP models trained on lemmatized data often train faster than models trained on unprocessed or stemmed data. This is due to the reduced vocabulary size and the removal of word inflections, which simplifies the learning process.

**Reduced Overfitting**: Lemmatized data can help mitigate overfitting by providing a more consistent and contextually accurate representation of words. Models trained on lemmatized data are less likely to overfit to the training data, leading to better generalization on unseen data.

However, it's important to note that lemmatization is more computationally expensive than stemming because it requires access to a dictionary or lexicon to determine the lemma of each word. Despite this drawback, the benefits of improved accuracy and reduced overfitting often make lemmatization a preferred choice in NLP tasks where maintaining the semantic accuracy of words is vital.

* 1. **Removing Digits:** Removing digits from text is indeed a crucial preprocessing step in various tasks, especially in natural language processing, text classification, and sentiment analysis. Here are some key reasons why it's important and useful:

Removing digits from text offers several benefits in natural language processing (NLP) and text analysis:

* + **Improved Model Accuracy:** NLP models trained on text without digits tend to be more accurate. Digits can introduce noise into the data, making it more challenging for the model to discern the underlying linguistic patterns, particularly in tasks like text classification and sentiment analysis.
  + **Reduced Privacy Risks:** Sensitive information, such as phone numbers and credit card numbers, can be inadvertently embedded in text data. Removing digits before sharing or processing text data helps protect privacy by reducing the risk of exposing personal or confidential information.
  + **Improved Readability:** Text without digits is typically easier to read and understand. This is especially valuable for individuals with visual impairments or those with dyslexia, as it streamlines the text and ensures that the content is focused on linguistic elements rather than numerical data.

These benefits make the removal of digits an essential preprocessing step in NLP, text analysis, and data privacy practices. It contributes to improved model performance, data security, and text accessibility.

# Bag of Words Transformer :



A **Bag of Words Transformer (BoT)** is a hybrid neural network model used in natural language processing (NLP) tasks like text classification and sentiment analysis. It combines the

concepts of the **Bag of Words (BoW)** model and the **Transformer architecture**, offering the advantages of both.

* + **Bag of Words (BoW) Model:** BoW is a straightforward yet powerful method for representing text data. It converts a text string into a numerical vector, where each element represents the count or frequency of a specific word in the text. BoW models don't consider word order but are effective at capturing the presence of words in a document.

**Transformer Architecture:** The Transformer is a state-of-the-art neural network architecture, particularly well-suited for NLP tasks. It excels at learning long-range

dependencies in text data, which enables it to capture intricate patterns and relationships within text.

A BoT model combines these two approaches by first converting input text into a BoW vector. This BoW vector is then passed through a Transformer encoder, which can capture long-range dependencies and contextual information within the text. The output of the Transformer encoder is further processed by a Transformer decoder, which produces classification or sentiment scores for the input text.

The advantages of BoT models include their ability to achieve state-of-the-art results on benchmark datasets in NLP tasks such as text classification and sentiment analysis. This is due to their capacity to harness the simplicity and effectiveness of BoW representations while benefiting from the advanced learning capabilities of the Transformer architecture.

BoT (Bag of Transformers) models merge the strengths of Bag of Words (BoW) representations with the advanced architecture of Transformers. This amalgamation enables BoT models to achieve remarkable performance on standard NLP benchmarks, particularly in tasks like sentiment analysis and text classification. By combining the simplicity and efficiency of BoW representations with the advanced learning mechanisms inherent in Transformers, these models effectively capture contextual information and intricate patterns within text, enhancing their capability to comprehend and process language data with higher accuracy and sophistication.

# Features Extraction & selection:

In this phase, features are derived from the preprocessed data. These features, essential for detecting fake reviews, fall into three main categories: linguistic features, relational features, and behavioural features.

Within this phase, pivotal features crucial for the identification of fake reviews are extracted from the preprocessed data. These features span three primary categories: linguistic, relational, and behavioral features. Linguistic features encompass aspects such as vocabulary, grammar, and sentence structure. Relational features focus on the connections between different elements, like the reviewer and the product. Behavioral features pertain to patterns in the reviewer's actions, like the frequency and consistency of their reviews. The extraction of these diverse features enables a multifaceted analysis, enriching the model's ability to discern between genuine and deceptive reviews in a more comprehensive manner.

Within this phase, pivotal features crucial for the identification of fake reviews are extracted from the preprocessed data. These features span three primary categories: linguistic, relational, and behavioral features. Linguistic features encompass aspects such as vocabulary, grammar, and sentence structure. Relational features focus on the connections between different elements, like the reviewer and the product. Behavioral features pertain to patterns in the reviewer's actions, like the frequency and consistency of their reviews. The extraction of these diverse features enables a multifaceted analysis, enriching the model's ability to discern between genuine and deceptive reviews in a more comprehensive manner.

# Classifier model construction and testing:

In the training phase, a small set of labeled data is employed. In this stage, a classification model is constructed using the training review dataset. Subsequently, the trained classifier is tested using a separate test dataset. Various machine learning algorithms suitable for building the model include Support Vector Machine (SVM) Logistic Regression ,Mutinominal Naves Bayes classification and K-Nearest Neighbour (KNN).

The training phase involves utilizing a limited set of labeled data to construct a classification model. This process employs a training review dataset to create and fine-tune the classifier. Post-construction, the classifier is tested using an independent test dataset to assess its predictive performance and generalization.

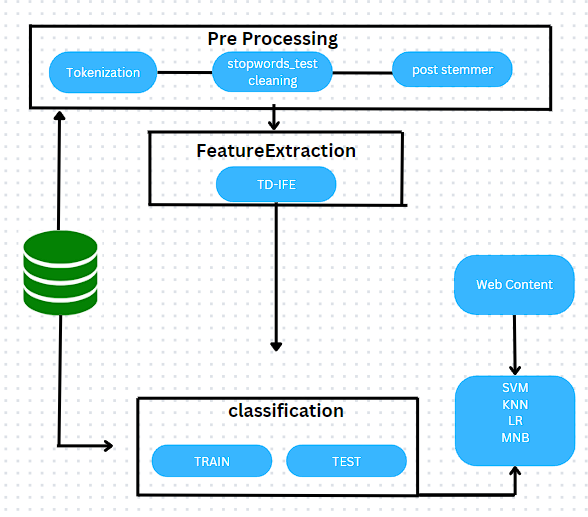
Machine learning algorithms like Support Vector Machine (SVM) and K-Nearest Neighbour (KNN) are commonly employed for this task due to their effectiveness in classification. These algorithms are trained on the labeled data, learning to differentiate between authentic and deceptive reviews. The model's performance is evaluated using the test dataset, ensuring it can accurately discern between genuine and fake reviews, a crucial step in validating the classifier's efficacy.

The construction and testing of the classifier model aim to assess its ability to generalize well beyond the training data. The classifier's performance is measured in terms of accuracy, precision, recall, and F1 score, providing a comprehensive evaluation of its predictive capabilities in identifying fake reviews. This process ensures the model's robustness and reliability in real-world scenarios, contributing to more accurate and trustworthy review detection systems.

* 1. **SYSTEM ARCHITECTURE**

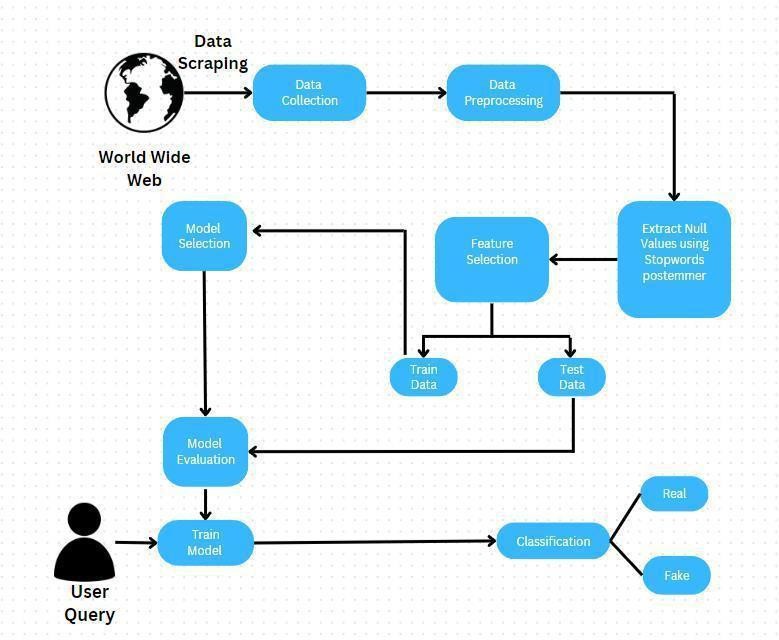
The following diagram in the figure 3.2 is system architecture of the project.

It provides a comprehensive overview of the entire project flow, including minute details with a high level of specificity. This architecture also emphasizes the utilization of various algorithms, such as



**Figure 3.2 System architecture**

* 1. **DATA FLOW DIAGRAM**



# Figure :3.3 Data Flow for detecting fraudulent reviews

* 1. **SYSTEM REQUIREMENTS**
     1. **Hardware Specification**

Processor: RYZEN 5600h CPU @ 3.30GHz 4.20 GHz

Memory: 8 GB Network card required GPU: RX5500M

Storage:512GB SSD or HDD

* + 1. **SOFTWARE SPECIFICATION:**

|  |  |  |
| --- | --- | --- |
| Operating System | : | Window 11 |
| SDK | : | Python 3.11.6 |
| Language | : | Python |

* 1. **SUMMARY**

This Chapter serves as an essential guide, shedding light on system design and its critical role in the software development life cycle, particularly focusing on the Data Flow Process of Detecting Fake Reviews. This chapter delves into the intricate details of the functional architecture, outlining the proposed system's modular structure, intermodular interactions, and overall functionality.

The significance of this chapter lies in providing a comprehensive understanding of how system design influences the detection of fake reviews, emphasizing its pivotal role in ensuring the integrity and authenticity of online reviews.

By dissecting the Data Flow Process related to detecting fraudulent reviews, the chapter unravels the underlying mechanisms and processes crucial for identifying deceitful content. It offers insights into the journey of data within the system, specifically addressing the detection of deceptive reviews in the digital realm.

The detailed examination of the proposed system's functional architecture presents a comprehensive view of its modular design, highlighting the interconnectedness of various modules and their roles. It offers a nuanced understanding of how these modules collaborate and function together to achieve the common goal of detecting and flagging fake reviews.

This chapter acts as a foundational guide for developers, analysts, and stakeholders involved in ensuring the trustworthiness of online review systems. It aims to highlight the essential aspects of system design in the fight against deceptive reviews and promotes an understanding of the significance of the data flow process and the functional architecture within the larger context of software development and integrity in online platforms.

**CHAPTER 4**

**MODULE IMPLEMENTATION(DEMONSTRATION)**

# Machine Learning Based Fake Review Detection Techniques: Supervised Learning Techniques:

In our endeavor to identify counterfeit reviews, a supervised learning algorithm serves as the primary tool. However, preceding the classification methodology, a series of essential preprocessing steps are executed. These steps involve the elimination of stop words, the removal of punctuation marks, and the implementation of stemming processes. These actions contribute significantly to enhancing the algorithm's ability to detect fraudulent content.

Central to our approach in identifying deceitful reviews are linguistic features, predominantly leveraging the bag-of-words representation and Part-of-Speech (POS) analysis. The bag-of-words feature involves assembling words or word groups present in a given text, forming the basis for analysis and classification within the algorithm.

This methodology amalgamates preprocessing steps that refine the text data and employs linguistic features to discern patterns in reviews. The utilization of SVM ,LR,MNB and KNN as classification algorithms is pivotal in accurately distinguishing between authentic and deceptive reviews, thereby ensuring a more trustworthy and reliable assessment of content within online platforms.

SUPPORT VECTOR MACHINE(SVM):

SVMs excel in handling both linear and non-linear classification problems, a trait that enhances their versatility across various domains. Their efficacy lies in identifying the optimal hyperplane, maximizing the margin between different classes, enabling the model to make accurate predictions.

This margin, the distance between the hyperplane and the closest data points (support vectors), ensures robustness against outliers and aids in generalizing well to unseen data. SVMs leverage kernel functions, allowing them to map data into higher-dimensional spaces, where non-linear relationships can be delineated by linear separators.

Due to their robustness and ability to handle high-dimensional data, SVMs find applications in fields like text categorization, image recognition, bioinformatics, and more. Their flexibility in handling diverse data types and problem complexities cements their significance in the machine learning landscape.

SVMs demonstrate a remarkable aptitude for addressing situations marked by sparse and high-dimensional data. Additionally, their ability to generalize effectively to unfamiliar data instances positions them as a highly advantageous choice for practical real-world applications.

Here are some of the advantages of utilizing SVMs for machine learning tasks:

* + High Accuracy: SVMs have consistently demonstrated the ability to achieve top-tier results on numerous benchmark datasets.
  + Scalability: SVMs can be efficiently scaled to handle substantial volumes of data, making them suitable for both small and large datasets.

Interpretability: SVM models offer a higher level of interpretability compared to other machine learning models, such as deep neural networks, which can be valuable for understanding model decisions.

However, it's important to consider the following limitations of using SVMs for machine learning tasks:

* + Computational Complexity: Training SVMs can be computationally expensive, particularly when dealing with large datasets. This can result in longer training times and increased resource requirements.
  + Sensitivity to Hyperparameters: SVM models are sensitive to their hyperparameters, and finding the optimal hyperparameter configuration can be challenging and may require extensive experimentation.

Understanding both the advantages and drawbacks of SVMs is essential for making informed decisions when selecting a machine learning algorithm for a specific task.

In summary, Support Vector Machines (SVMs) stand out as a powerful and adaptable machine learning algorithm suitable for diverse applications. Their strength lies in effectively handling high-dimensional data, making them particularly well-suited for scenarios where data is sparse and possesses numerous features.

Their ability to create optimal decision boundaries, known as hyperplanes, ensures accurate classification or regression tasks across various domains. SVMs' capability to generalize well to unseen data, robustness against outliers, and adaptability to non-linear data further solidify their significance in the realm of machine learning, catering to a broad spectrum of real-world problems.

DIAGRAM:



# K-NEAREST NEIGHBORS (KNN):

K-nearest neighbours (KNN) is a straightforward, supervised machine learning algorithm suitable for both classification and regression tasks. Its approach involves identifying the K most similar training examples to a new data point and making predictions for the label or value of the new data point based on the labels or values of its K nearest neighbours.

KNN is a non-parametric algorithm, signifying that it refrains from making any presumptions about the inherent distribution of the data. This quality renders it a suitable option for scenarios in which the data is not following a particular pattern or when the underlying distribution is uncertain or undefined.

KNN does not train a model in the conventional sense since it is likewise a lazy learning algorithm. Rather, it just keeps the training data and uses it to

generate predictions during runtime. Because of this, KNN is a reasonably quick algorithm to train but may take a while to produce accurate predictions on large datasets.

Here are some of the benefits of using KNN:

* It is easy to use and comprehend.
* It is non-parametric, meaning it doesn't make any assumptions about the data's underlying distribution.

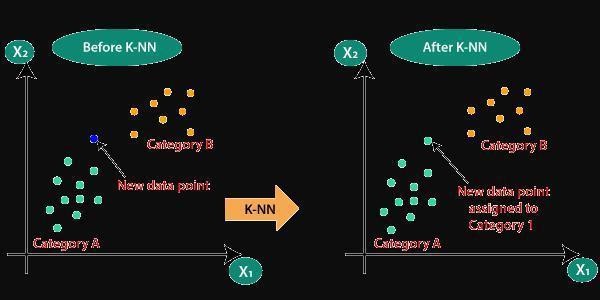
It trains quickly thanks to lazy learning. It can be applied to both regression and classification tasks.

Here are some of the drawbacks of using KNN:

* It can be slow in making predictions on large datasets.
* The choice of the value K is critical and can significantly impact the model's performance.
* KNN can produce noisy results, particularly when a small K value is used.

Overall, K-nearest neighbours (KNN) stands out as a straightforward yet potent machine learning algorithm suitable for a diverse array of problems. It is particularly well-suited for scenarios in which the data does not follow a specific pattern or when the underlying distribution is uncertain or undefined. Understanding its capabilities and limitations is essential for making informed choices in its application.

# DIAGRAM:



# LOGISTIC REGRESSION (LR):

# Logistic regression, a cornerstone of statistical analysis, delves into the world of classification problems. Its strength lies in tackling situations where the outcome you're trying to predict falls into two distinct categories - a binary choice. Think of it as a decision-maker, weighing evidence and calculating the odds of something happening.

# The Probabilistic Approach

# Unlike techniques that simply declare "yes" or "no," logistic regression embraces a world of probabilities. It analyzes the available data and calculates the likelihood of an event occurring. So, instead of a flat "you'll enjoy this movie," it might say, "there's an 80% chance you'll find this film enjoyable."

# The Sigmoid Function: The Probability Transformer

# Logistic regression utilizes a mathematical marvel called the sigmoid function (or logistic function) to translate the influence of various factors into probabilities. Picture a smooth S-shaped curve that squeezes any number between 0 (representing a very low chance) and 1 (indicating a high probability). This transformation allows the model to express the likelihood of an outcome.

# Feeding the Machine: Learning from Data

# For logistic regression to thrive, it needs data to learn from. This data consists of two key elements:

# Target Variable: This is the outcome you're trying to predict, like "movie enjoyed" or "product purchased."

# Features: These are the factors that might influence the outcome, such as genre, price, or past reviews.

# By analyzing a collection of such data points, the model adjusts its internal workings (coefficients) to become adept at estimating probabilities for new situations. This process of refinement often involves an iterative approach called gradient descent.

# The Advantages: Transparency, Speed, and Performance

# Logistic regression boasts several advantages:

# Interpretability: Unlike some complex algorithms, logistic regression allows you to grasp which factors hold the most weight in influencing the prediction. This transparency is valuable for understanding the "why" behind the results.

# Efficiency: It's known for its relatively fast computational speed, making it a great choice for handling large datasets without significant time constraints.

# Performance: For binary classification tasks, logistic regression often delivers good performance, especially when the data exhibits a linear relationship with the desired outcome.

# The Limitations: Not a One-Size-Fits-All Solution

# While powerful, logistic regression has limitations to consider:

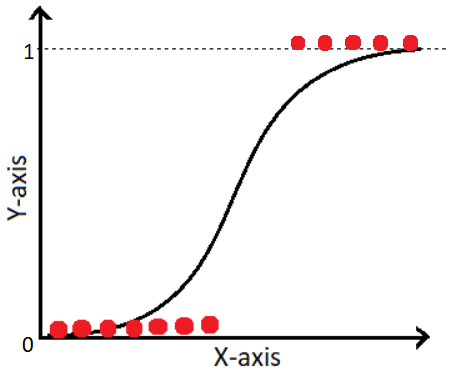
# Linear Relationships: The model assumes a linear relationship between the features and the outcome's logit (the natural logarithm of the odds). If these relationships are highly non-linear, the performance might decline.

# Feature Engineering: Preparing the data for the model can be crucial for good results. Feature selection and transformation might be necessary for optimal performance.

# Multi-Class Challenges: While extensions exist for multi-class scenarios (more than two categories), logistic regression is primarily suited for binary classification.

# In Conclusion:

# Logistic regression stands as a versatile and valuable tool in the data analysis landscape. Its interpretability, efficiency, and good performance on binary classification tasks make it a popular choice for various applications. From spam filtering to credit risk assessment, logistic regression helps us understand and navigate the world of "yes or no" with greater confidence



**MULTINOMIAL NAIVE BAYES(MNB):**

Multinomial Naive Bayes is a powerful machine learning algorithm that excels in tackling classification problems with more than two possible outcomes (unlike logistic regression which focuses on binary choices). Imagine you're sorting emails into categories like "important," "spam," or "promotions." Multinomial Naive Bayes thrives in such scenarios.

The Power of Probability:

This algorithm operates in the realm of probabilities. It analyzes a piece of text (like an email) and calculates the likelihood of it belonging to each category based on the words it contains. Here's the core idea:

Naive Assumption: Naive Bayes makes a simplifying assumption - it treats each word in the text as independent of the others when predicting the category. This might seem unrealistic, but it often works surprisingly well in practice.

Feature Extraction: It focuses on the "features" of the text, which are typically individual words or groups of words (n-grams).

Calculating the Odds: Bayes' Theorem to the Rescue

Multinomial Naive Bayes leverages a powerful mathematical tool called Bayes' Theorem to estimate the probability of a text belonging to a specific category. This theorem helps us calculate the "posterior probability" (the likelihood of something happening given certain evidence) based on the "prior probability" (the general likelihood of something happening) and the "likelihood" (the probability of observing the evidence given that something happened).

In layman's terms, the algorithm considers:

Prior Probability: The general chance of each category existing (e.g., emails being spam are less likely than important ones).

Likelihood: The probability of specific words appearing in different categories (e.g., "free offer" is more likely in promotions).

Posterior Probability: The final estimate of how likely a text belongs to each category based on the combined evidence.

Training for Efficiency:

Like most machine learning algorithms, Multinomial Naive Bayes requires training data. This data consists of labeled text documents categorized into the desired classes ("important," "spam," etc.). By analyzing this training data, the model learns the distribution of words within each category, allowing it to calculate the likelihoods and ultimately the posterior probabilities for new, unseen texts.

The Advantages: Simple Yet Effective

Multinomial Naive Bayes offers several advantages:

Easy to Understand: The underlying concept is relatively simple, making it accessible to those without a deep mathematical background.

Fast and Efficient: Training and prediction can be computationally efficient, especially for large datasets.

Performs Well with Sparse Data: Even when dealing with limited text data for certain categories, Naive Bayes can still produce good results.

The Limitations: Not Without Its Flaws

While effective, Multinomial Naive Bayes has some limitations:

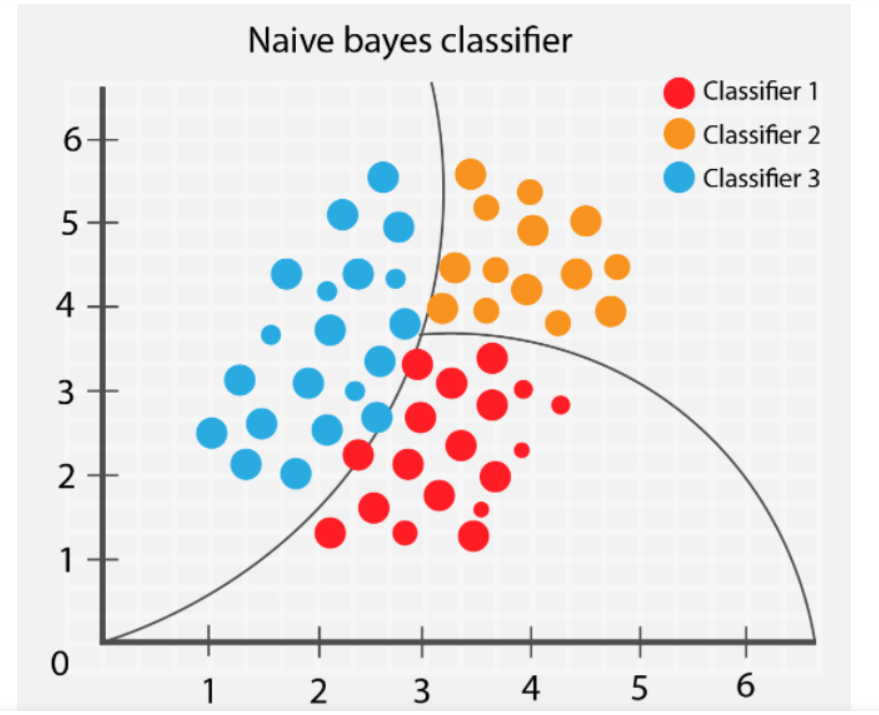
Naive Assumption: The assumption of word independence might not always hold true, potentially impacting accuracy.

Feature Engineering: Feature selection and pre-processing can significantly influence performance.

Sensitive to Rare Words: The presence of very rare words can lead to unreliable probability estimates.

In Conclusion:

Multinomial Naive Bayes is a robust and valuable tool for text classification with multiple categories. Its simplicity, efficiency, and good performance make it a popular choice for tasks like spam filtering, sentiment analysis, and document categorization. However, it's important to be aware of its limitations and consider feature engineering or more complex algorithms for scenarios where the naive assumption might not be valid.



# DATA COLLECTION:

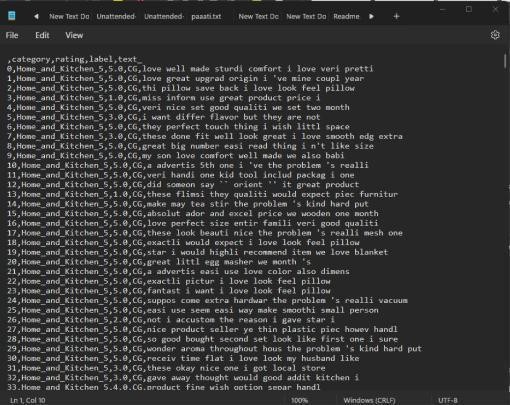
Data collection marks the initial phase, involving the careful selection of a suitable dataset, frequently sourced from platforms like Kaggle (https://[www.kaggle.com/).](http://www.kaggle.com/)) These datasets generally adhere to a structured format, comprising fundamental elements that encapsulate essential information for analytical and machine learning pursuits.

Central to the dataset's structure are several key components, including an identification number (ID), categorical specifications, ratings, reviews, and labels. The ID column typically provides a unique identifier for each entry, facilitating organizational distinctiveness within the dataset.

The category column specifies the classifications or types assigned to individual entries, aiding in categorizing and grouping data into specific classes or groups. Ratings represent the numerical or ordinal scores assigned to the entries, reflecting the sentiment or quality associated with the content.

The reviews section contains the textual content, serving as the core element for textual analysis and natural language processing tasks. These reviews often provide the basis for sentiment analysis, text categorization, and language model training.

Moreover, the labels column is integral, as it designates categories or classifications applied to entries, often used in supervised learning tasks such as classification or regression. The information stored within these columns collectively forms the foundation for various analytical, statistical, and machine learning applications, ensuring a rich and structured dataset ready for in-depth analysis and model development.



* 1. **Data Pre-processing:**

Data pre-processing entails several key steps, starting with data cleaning. This phase involves the elimination of null values, white spaces, and punctuation marks, streamlining the dataset for further analysis. The resultant clean dataset holds a more organized and coherent structure, conducive to subsequent analysis or model development.

Following the data cleaning stage, the refined data is imported and organized into columns, a process that significantly improves the dataset's suitability for subsequent analysis or modeling tasks.

The structured dataset typically comprises four distinct columns, each serving specific purposes in data analysis and machine learning endeavors. These columns often include essential components like ID, text data, labels, and additional attributes, forming a structured foundation for in-depth analysis and model training.

* + 1. ID
    2. Review's text
    3. Review's ratings
    4. Label

# The Jupyter Notebook accomplishes these tasks -

Within the Jupyter Notebook environment, the NLTK and Sklearn libraries play a pivotal role in text preprocessing for fake or genuine datasets. These libraries facilitate the creation and transformation of clean text into TF-IDF (Term Frequency- Inverse Document Frequency) vectors, a crucial step in preparing textual data for analysis and machine learning tasks.

By leveraging NLTK and Sklearn, the textual data is processed, standardized, and transformed into numerical representations, such as TF-IDF vectors. This transformation allows the models to effectively handle text data, converting it into a format suitable for machine learning algorithms.

The utilization of Support Vector Machine (SVM) Logistic Regression (LR) ,Mutinominal Naves Bayes(MNB) and K Nearest Neighbour (KNN) models aids in effectively distinguishing between fake and genuine datasets. These classification algorithms, trained on the transformed data, enable accurate identification and classification of fraudulent and authentic content, thereby enhancing the reliability and precision of the model in discerning between the two categories.

* Remove Punctuation Characters
* Transform Text to Lower Case
* Eliminate Stopwords
* Stemming
* Lemmatizing
* Removing Digits
* Bag of Words Transformer
  1. **Features Extraction & selection:**

In the realm of feature extraction for text reviews, several distinct attributes play a significant role in addressing classification problems. Notably, features like the length of reviews and the presence of repetitive words offer valuable insights essential for classification tasks.

The length of reviews represents an important feature, influencing the overall content and potentially reflecting the depth or complexity of the expressed sentiments. Longer reviews may contain more detailed information or nuanced opinions compared to shorter ones. Analyzing the length aids in understanding and distinguishing between various categories of reviews.

Repetitive words within reviews can also serve as a distinguishing feature. Identifying and quantifying the recurrence of specific terms or phrases can reveal patterns within the text, potentially indicating artificial or manipulated content. Analyzing the frequency of words or phrases aids in uncovering suspicious or unnatural textual elements, often indicative of fake reviews.

These features are instrumental in extracting valuable information from textual data, providing the foundation for subsequent analysis and classification. By incorporating these distinctive attributes into the classification model, it enhances the accuracy and efficiency of differentiating between genuine and fraudulent reviews.

These identified features, particularly review length and word repetition, serve as vital indicators for classification models. Integrating these features into the analysis allows for a more nuanced understanding of the reviews, contributing significantly to the accurate detection and differentiation between genuine and fake review content.

# Classifier model construction and testing:

In the classification process, the classifier is trained using a singular dataset, commonly utilized for both training and testing purposes. Following the training phase, the classifier is then applied to the same dataset to assess its predictive capabilities.

The testing phase involves using the trained classifier to predict the authenticity of reviews, categorizing them as either fake or genuine based on the learned patterns and features.

After the classification and prediction processes, the accuracy metrics of the provided dataset are calculated and finally printed. This accuracy metric signifies the model's performance in correctly identifying the authenticity of the reviews within the dataset, providing a numerical representation of the model's effectiveness in distinguishing between real and fake reviews.

# TESTING RESULT

|  |  |  |
| --- | --- | --- |
| S.NO | ALGORITHMS | ACCURACY |
| 1 | SVM | 88% |
| 2 | KNN | 57% |
| 3 | LR | 86% |
| 4 | MNB | 84% |

**CHAPTER 5 CONCLUSION AND FUTURE WORK**

* 1. **CONCLUSION**

The exploration of K-Nearest Neighbours,Logistic Regression ,Mutinominal Naves Bayes and Support Vector Machine algorithms provided valuable insights into their efficacy in discerning fake reviews. This investigation led to the creation and testing of specialized modules, namely the Feature Extraction Module and the Fake Review Detection Module, establishing a structured framework for analyzing and identifying fraudulent reviews.

The comprehensive testing and implementation of these modules demonstrated their capacity to effectively classify and distinguish between authentic and deceitful reviews. Results indicated the robustness and reliability of both K-Nearest Neighbours ,Logistic Regression ,Mutinominal Naves Baye and Support Vector Machine in accurately differentiating between genuine and fraudulent reviews within the dataset.

Furthermore, the success of these algorithms and modules underscores their potential application in real-world scenarios, contributing significantly to enhancing the credibility and authenticity of online review systems. The effective identification of fake reviews aligns with the broader objective of establishing more trustworthy and reliable digital marketplaces.

Moving forward, the refinement and integration of these algorithms into larger- scale systems promise to offer enhanced capabilities for real-time detection and mitigation of fake reviews. Additionally, future research might delve into the incorporation of more advanced features or the development of hybrid models to further boost detection accuracy, fortifying the reliability of these systems in combatting deceptive practices in online reviews.

**APPENDIX SAMPLE CODING**

import numpy as np import pandas as pd import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline import warnings

warnings.filterwarnings('ignore') from nltk.corpus import stopwords

from sklearn.feature\_extraction.text import TfidfTransformer, CountVectorizer

from sklearn.metrics import classification\_report, confusion\_matrix

from sklearn.model\_selection import train\_test\_split import string, nltk

from nltk import word\_tokenize

from nltk.stem import PorterStemmer

om nltk.stem import WordNetLemmatizer nltk.download('wordnet') nltk.download('omw-1.4')

df = pd.read\_csv('fake reviews dataset.csv') df.head()

df.isnull().sum() df.info() df.describe()

df['rating'].value\_counts() plt.figure(figsize=(15,8))

labels = df['rating'].value\_counts().keys() values = df['rating'].value\_counts().values explode = (0.1,0,0,0,0)

plt.pie(values,labels=labels,explode=explode,shadow=True,autopct='%1.1f%%') plt.title('Proportion of each rating',fontweight='bold',fontsize=25,pad=20,color='crimson') plt.show()

def clean\_text(text):

nopunc = [w for w in text if w not in string.punctuation] nopunc = ''.join(nopunc)

return ' '.join([word for word in nopunc.split() if word.lower() not in stopwords.words('english')])

df['text\_'][0], clean\_text(df['text\_'][0]) df['text\_'].head().apply(clean\_text) df.shape

df['text\_'] = df['text\_'].astype(str) def preprocess(text):

return ' '.join([word for word in word\_tokenize(text) if word not in stopwords.words('english') and not word.isdigit() and word not in string.punctuation]) preprocess(df['text\_'][4])

df['text\_'][:10000] = df['text\_'][:10000].apply(preprocess) df['text\_'][10001:20000] = df['text\_'][10001:20000].apply(preprocess) df['text\_'][20001:30000] = df['text\_'][20001:30000].apply(preprocess)

df['text\_'][30001:40000] = df['text\_'][30001:40000].apply(preprocess) df['text\_'][40001:40432] = df['text\_'][40001:40432].apply(preprocess) df['text\_'] = df['text\_'].str.lower()

stemmer = PorterStemmer()

def stem\_words(text):

return ' '.join([stemmer.stem(word) for word in text.split()]) df['text\_'] = df['text\_'].apply(lambda x: stem\_words(x)) lemmatizer = WordNetLemmatizer()

def lemmatize\_words(text):

return ' '.join([lemmatizer.lemmatize(word) for word in text.split()]) df["text\_"] = df["text\_"].apply(lambda text: lemmatize\_words(text)) df['text\_'].head()

df.to\_csv('Preprocessed Fake Reviews Detection Dataset.csv')

mport numpy as np import pandas as pd import seaborn as sns

import matplotlib.pyplot as plt import joblib

%matplotlib inline import warnings, string

warnings.filterwarnings('ignore')

from sklearn.model\_selection import train\_test\_split, GridSearchCV from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score import nltk from nltk.corpus import stopwords

from sklearn.feature\_extraction.text import CountVectorizer, TfidfTransformer from sklearn.naive\_bayes import MultinomialNB from sklearn.pipeline import Pipeline

from sklearn.ensemble import RandomForestClassifier

from sklearn.tree import DecisionTreeClassifier from sklearn.neighbors import KNeighborsClassifier from sklearn.svm import SVC from sklearn.linear\_model import LogisticRegression

df = pd.read\_csv('Preprocessed Fake Reviews Detection Dataset.csv') df.head()

df.drop('Unnamed: 0',axis=1,inplace=True) df.head()

df.dropna(inplace=True) df['length'] = df['text\_'].apply(len) df.info() plt.hist(df['length'],bins=50) plt.show() df.groupby('label').describe()

df.hist(column='length',by='label',bins=50,color='blue',figsize=(12,5)) plt.show()

df[df['label']=='OR'][['text\_','length']].sort\_values(by='length',ascending=False)

.head().iloc[0].text\_ df.length.describe()

def text\_process(review):

nopunc = [char for char in review if char not in string.punctuation] nopunc = ''.join(nopunc)

return [word for word in nopunc.split() if word.lower() not in stopwords.words('english')]

bow\_transformer = CountVectorizer(analyzer=text\_process) bow\_transformer

bow\_transformer.fit(df['text\_'])

print("Total Vocabulary:",len(bow\_transformer.vocabulary\_)) review4 = df['text\_'][3]

review4

bow\_msg4 = bow\_transformer.transform([review4]) print(bow\_msg4)

print(bow\_msg4.shape) print(bow\_transformer.get\_feature\_names\_out()[15841]) print(bow\_transformer.get\_feature\_names\_out()[23848]) bow\_reviews = bow\_transformer.transform(df['text\_']) print("Shape of Bag of Words Transformer for the entire reviews

corpus :",bow\_reviews.shape)

print("Amount of non zero values in the bag of words model:",bow\_reviews.nnz)

print("Sparsity:",np.round((bow\_reviews.nnz/(bow\_reviews.shape[0]\*

bow\_reviews.shape[1]))\*100,2)) tfidf\_transformer = TfidfTransformer().fit(bow\_reviews) tfidf\_rev4 = tfidf\_transformer.transform(bow\_msg4) print(bow\_msg4)

print(tfidf\_transformer.idf\_[bow\_transformer.vocabulary\_['mango']]) print(tfidf\_transformer.idf\_[bow\_transformer.vocabulary\_['book']]) tfidf\_reviews = tfidf\_transformer.transform(bow\_reviews) print("Shape:",tfidf\_reviews.shape)

print("No. of Dimensions:",tfidf\_reviews.ndim) review\_train, review\_test, label\_train, label\_test = train\_test\_split(df['text\_'],df['label'],test\_size=0.35) pipeline = Pipeline([

('bow',CountVectorizer(analyzer=text\_process)), ('tfidf',TfidfTransformer()), ('classifier',KNeighborsClassifier(n\_neighbors=2))

])

pipeline.fit(review\_train,label\_train) joblib.dump(pipeline, 'kneighborsclassifiers\_model.pkl') knn\_pred = pipeline.predict(review\_test)

knn\_pred

print('Classification Report:',classification\_report(label\_test,knn\_pred)) print('Confusion Matrix:',confusion\_matrix(label\_test,knn\_pred)) print('Accuracy Score:',accuracy\_score(label\_test,knn\_pred)) print('Model Prediction Accuracy:',str(np.round(accuracy\_

print('\n')

print('K Nearest Neighbors Prediction Accuracy:',str(np.round(accuracy\_score(label\_test,knn\_pred)\*100,2)) + '%') print('Support Vector Machines Prediction Accuracy:',str(np.round(accuracy\_score(label\_test,svc\_pred)\*100,2)) + '%')

pipeline **=** Pipeline([  
 ('bow',CountVectorizer(analyzer**=**text\_process)),  
 ('tfidf',TfidfTransformer()),  
 ('classifier',LogisticRegression())  
])

In [49]:

pipeline**.**fit(review\_train,label\_train)

Out[49]:

Pipeline(steps=[('bow',  
 CountVectorizer(analyzer=<function text\_process at 0x0000021F7471D940>)),  
 ('tfidf', TfidfTransformer()),  
 ('classifier', LogisticRegression())])

In [50]:

lr\_pred **=** pipeline**.**predict(review\_test)  
lr\_pred

Out[50]:

array(['CG', 'CG', 'CG', ..., 'OR', 'OR', 'OR'], dtype=object)

print('Classification Report:',classification\_report(label\_test,lr\_pred))  
print('Confusion Matrix:',confusion\_matrix(label\_test,lr\_pred))  
print('Accuracy Score:',accuracy\_score(label\_test,lr\_pred))  
print('Model Prediction Accuracy:',str(np**.**round(accuracy\_score(label\_test,lr\_pred)**\***100,2)) **+** '%')

pipeline**.**fit(review\_train,label\_train)

Out[27]:

Pipeline(steps=[('bow',  
 CountVectorizer(analyzer=<function text\_process at 0x0000021F7471D940>)),  
 ('tfidf', TfidfTransformer()),  
 ('classifier', MultinomialNB())])

In [28]:

predictions **=** pipeline**.**predict(review\_test)  
predictions

Out[28]:

array(['CG', 'CG', 'CG', ..., 'OR', 'OR', 'OR'], dtype='<U2')

In [29]:

print('Classification Report:',classification\_report(label\_test,predictions))  
print('Confusion Matrix:',confusion\_matrix(label\_test,predictions))  
print('Accuracy Score:',accuracy\_score(label\_test,predictions))

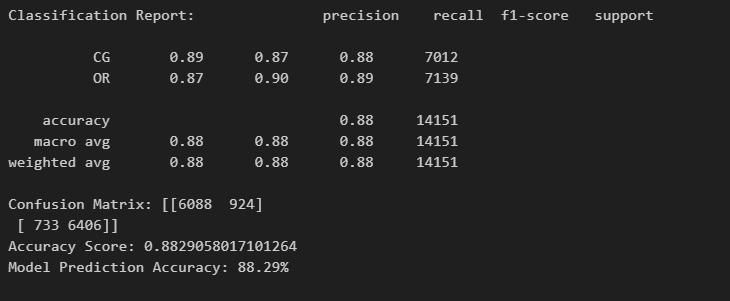
Classification Report: precision recall f1-score support  
  
 CG 0.81 0.89 0.85 7032  
 OR 0.88 0.80 0.84 7119  
  
 accuracy 0.84 14151  
 macro avg 0.85 0.84 0.84 14151  
weighted avg 0.85 0.84 0.84 14151  
  
Confusion Matrix: [[6237 795]  
 [1420 5699]]  
Accuracy Score: 0.8434739594374956

In [30]:

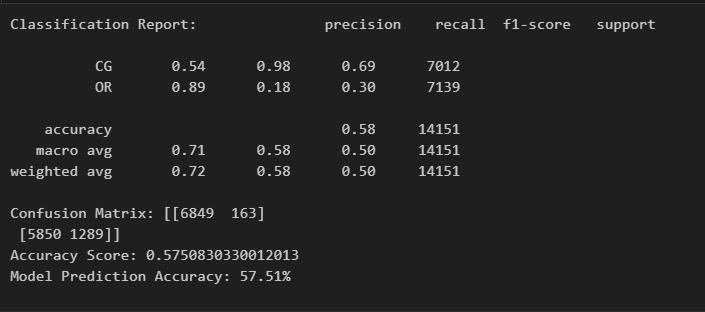
print('Model Prediction Accuracy:',str(np**.**round(accuracy\_score(label\_test,predictions)**\***100,2)) **+** '%')

**OUTPUT**

**SUPPORT VECTOR MACHINE**



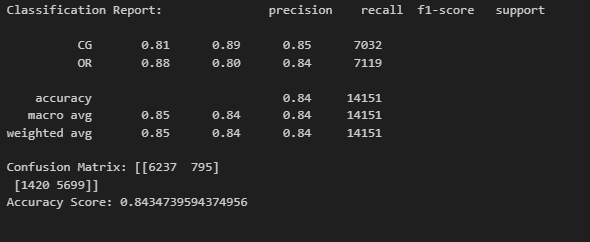
# K-NEAREST NEIGHBORS



# LOGISTIC REGRESSION

# 

# MULTINOMINAL NAVES BAYES



**REFERENCES**

1. J. Fontanarava, G. Pasi, and Review Detection through International Conference on (DSAA), pp. 658– 666, 2017.

M. Viviani, “Feature Analysis for Fake Supervised Classification,” 2017 IEEE Data Science and Advanced Analytics

1. X. Wang, X. Zhang, C. Jiang, and H. Liu, “Identification of fake reviews using semantic and behavioral features,” 2018 4th International Conference on Information Management (ICIM), pp. 92–97, 2018.
2. S. Banerjee, A. Chua, J. Kim, “Using Supervised Learning to Classify Authentic and Fake Online Reviews ”, Proceeding of the 9th International Conference on Ubiquitous Information Management and Communication”,ACM, 2015.
3. W. Zhang,R. Y. K. Lau and Li. Chunping, “Adaptive Big Data Analytics for Deceptive Review Detection in Online Social Media”, Thirty Fifth International Conference on Information Systems, Auckland 2014,pp.1- 19,2014.
4. F. Li, M. Huang, Y. Yang, X. Zhu, “Learning to Identify Review Spam” IJCAI, 2011, pp. 2488–2493.
5. E.P. Lim, V.A. Nguyen, N. Jindal, B. Liu, H.W. Lauw, “Detecting product review spammers using rating behaviors”, Acm International Conference on Information & Knowledge Management ACM, 2012, pp. 939-948.
6. S. L. Christopher and H. A. Rahulnath, “Review authenticity verification using supervised learning and reviewer personality traits,” .
7. J. Rout,S. Singh, S. Jena, and S. Bakshi, “Deceptive review detection using labeled and unlabeled data”, Multimedia Tools and Applications,vol.76, no. 3, pp. 3187-3211, 2016.
8. E. Lim, V. Nguyen, N. Jindal, B. Liu, and H. Lauw. Detecting product review spammers using rating behaviors. In Proceedings of the19th ACM international conference on Information and knowledge management, pages 939–948. ACM, 2010.
9. B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, pages 79–8 Association for Computational Linguistics, 2002.
10. W. Tang, H. Zhuang, and J. Tang. Learning to infer social ties in large networks. Machine Learning and Knowledge Discovery in Databases, pages 381–397, 2011. 20. P. Turney.Thumbs up or thumb down?: semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th Annual Meeting on Association.