“**DETECTING FAKE REVIEWS ON ONLINE PRODUCTS WITH MACHINE LEARNING**”

A Project Report submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR.

In Partial Fulfillment of the Requirements for the Award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

BY

|  |  |
| --- | --- |
| **ADAVENI SAI CHARAN** | **20121A0502** |
| **BOJJA OMKIRAN REDDY** | **20121A0541** |
| **BOMMAGANDLA SUPRIYA** | **20121A0542** |
| **CHILLU HARISH KUMAR** | **20121A0576** |

Under the Guidance of

**Mrs. Koruturu Harika**

Assistant Professor

Dept of CSE, SVEC



Department of Computer Science and Engineering

### **SREE VIDYANIKETHAN ENGINEERING COLLEGE**

(Affiliated to JNTUA, Anantapuramu)

Sree Sainath Nagar, Tirupathi – 517 102

2020-2024

**SREE VIDYANIKETHAN ENGINEERING COLLEGE**

(Affiliated to Jawaharlal Nehru Technological University Anantapur)

Sree Sainath Nagar, A. Rangampet, Tirupati – 517 102, Chittoor Dist., A.P.

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

CERTIFICATE

This is to certify that the Project Work entitled

**“DETECTING FAKE REVIEWS ON ONLINE PRODUCTS WITH MACHINE LEARNING“**

is the bonafide work done by

|  |  |
| --- | --- |
| **ADAVENI SAI CHARAN** | **20121A0502** |
| **BOJJA OMKIRAN REDDY** | **20121A0541** |
| **BOMMAGANDLA SUPRIYA** | **20121A0542** |
| **CHILLU HARISH KUMAR** | **20121A0576** |

In the Department of Computer Science and Engineering, Sree Vidyanikethan Engineering College, A. Rangampet. is affiliated to JNTUA, Anantapuramu in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering during 2020-2024.

This is work has been carried out under my guidance and supervision.

The results embodied in this Project report have not been submitted in any University or Organization for the award of any degree or diploma.

**Internal Guide Head**

###### Mrs. Koruturu Harika Dr. B. Narendra Kumar Rao

Assistant Professor Prof & Head

Dept of CSE Dept of CSE

Sree Vidyanikethan Engineering College Sree Vidyanikethan Engineering College

Tirupathi Tirupathi

**INTERNAL EXAMINER EXTERNAL EXAMINER**

###### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

###### VISION AND MISSION

**VISION**

To become a Centre of Excellence in Computer Science and Engineering by imparting high quality education through teaching, training and research.

##### MISSION

The Department of Computer Science and Engineering is established to provide undergraduate and graduate education in the field of Computer Science and Engineering to students with diverse background in foundations of software and hardware through a broad curriculum and strongly focused on developing advanced knowledge to become future leaders.

Create knowledge of advanced concepts, innovative technologies and develop research aptitude for contributing to the needs of industry and society.

Develop professional and soft skills for improved knowledge and employability of students.

Encourage students to engage in life-long learning to create awareness of the contemporary developments in computer science and engineering to become outstanding professionals.

Develop attitude for ethical and social responsibilities in professional practice at regional, National and International levels.

###### Program Educational Objectives (PEO’s)

##### 1. Pursuing higher studies in Computer Science and Engineering and related disciplines

##### 2. Employed in reputed Computer and I.T organizations and Government or have established startup companies.

##### 

##### 3. Able to demonstrate effective communication, engage in team work, exhibit leadership skills, ethical attitude, and achieve professional advancement through continuing education.

###### Program Specific Outcomes (PSO’s)

##### 1. Demonstrate knowledge in Data structures and Algorithms, Operating Systems, Database Systems, Software Engineering, Programming Languages, Digital systems, Theoretical Computer Science, and Computer Networks. (PO1)

##### 2. Analyze complex engineering problems and identify algorithms for providing solutions (PO2)

##### 3. Provide solutions for complex engineering problems by analysis, interpretation of data, and development of algorithms to meet the desired needs of industry and society. (PO3, PO4)

##### 4. Select and Apply appropriate techniques and tools to complex engineering problems in the domain of computer software and computer based systems (PO5)

###### Program Outcomes (PO’s)

##### 1. Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems (Engineering knowledge).

##### 2. Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences (Problem analysis).

##### 3. Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations (Design/development of solutions).

##### 4. Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions (Conduct investigations of complex problems).

##### 5. Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations (Modern tool usage)

##### 6. Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice (The engineer and society)

##### 7. Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the

##### knowledge of, and need for sustainable development (Environment and sustainability).

##### 8. Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice (Ethics).

##### 9. Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings (Individual and team work).

##### 10. Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions (Communication).

##### 11. Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments (Project management and finance).

##### 

##### 12. Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change (Life-long learning).

###### Course Outcomes

##### CO1. Knowledge on the project topic (PO1)

##### CO2. Analytical ability exercised in the project work.(PO2)

##### CO3. Design skills applied on the project topic. (PO3)

##### CO4. Ability to investigate and solve complex engineering problems faced during the project work. (PO4)

##### CO5. Ability to apply tools and techniques to complex engineering activities with an understanding of limitations in the project work. (PO5)

##### CO6. Ability to provide solutions as per societal needs with consideration to health, safety, legal and cultural issues considered in the project work. (PO6)

##### CO7. Understanding of the impact of the professional engineering solutions in environmental context and need for sustainable development experienced during the project work. (PO7)

##### CO8. Ability to apply ethics and norms of the engineering practice as applied in the project work.(PO8)

##### CO9. Ability to function effectively as an individual as experienced during the project work. (PO9)

##### CO10. Ability to present views cogently and precisely on the project work. (PO10)

##### CO11. Project management skills as applied in the project work. (PO11)

##### CO12. Ability to engage in life-long learning as experience during

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** | **PSO4** |
| **CO1** | 3 |  |  |  |  |  |  |  |  |  |  |  | 3 |  |  |  |
| **CO2** |  | 3 |  |  |  |  |  |  |  |  |  |  |  | 3 |  |  |
| **CO3** |  |  | 3 |  |  |  |  |  |  |  |  |  |  |  | 3 |  |
| **CO4** |  |  |  | 3 |  |  |  |  |  |  |  |  |  |  | 3 |  |
| **CO5** |  |  |  |  | 3 |  |  |  |  |  |  |  |  |  |  | 3 |
| **CO6** |  |  |  |  |  | 3 |  |  |  |  |  |  |  |  |  |  |
| **CO7** |  |  |  |  |  |  | 3 |  |  |  |  |  |  |  |  |  |
| **CO8** |  |  |  |  |  |  |  | 3 |  |  |  |  |  |  |  |  |
| **CO9** |  |  |  |  |  |  |  |  | 3 |  |  |  |  |  |  |  |
| **CO10** |  |  |  |  |  |  |  |  |  | 3 |  |  |  |  |  |  |
| **CO11** |  |  |  |  |  |  |  |  |  |  | 3 |  |  |  |  |  |
| **CO12** |  |  |  |  |  |  |  |  |  |  |  | 3 |  |  |  |  |

###### CO-PO Mapping

##### 

##### 

##### (Note: 3-High, 2-Medium, 1-Low)

##### DECLARATION

We hereby declare that this project report titled **“Detecting Fake Reviews On Online Products with Machine Learning”** is a genuine project work carried out by us, in **B.Tech *(Computer Science and Engineering)*** degree course of **Jawaharlal Nehru Technological University Anantapur** and has not been submitted to any other course or University for the award of any degree by us.

Signature of the student

1. Adaveni Sai Charan

2. Bojja Omkiran Reddy

3. Bommagandla Supriya

4. Chillu Harish Kumar

**ACKNOWLEDGEMENT**

We are extremely thankful to our beloved Chairman and founder **Dr. M. Mohan Babu** who took keen interest to provide us the infrastructural facilities for carrying out the project work.

We are highly indebted to **Dr. B.M. Satish**, Principal of Sree Vidyanikethan Engineering College for his valuable support and guidance in all academic matters.

We are very much obliged to **Dr. B. Narendra Kumar Rao,** Professor & Head, Department of CSE, for providing us the guidance and encouragement in completion of this project.

We would like to express our indebtedness to the project coordinator, **Dr. K. Padmaja**, Professor, Department of CSE for her` valuable guidance during the course of project work.

We would like to express our deep sense of gratitude to **Mrs. Koruturu Harika**, Assistant Professor, Department of CSE, for the constant support and invaluable guidance provided for the successful completion of the project.

We are also thankful to all the faculty members of CSE Department, who have cooperated in carrying out our project. We would like to thank our parents and friends who have extended their help and encouragement either directly or indirectly in completion of our project work.

**ABSTRACT**

Online reviews play a pivotal role in shaping consumers' purchasing decisions and serve as a crucial information source for gauging public sentiment on products or services. Given their significant impact, manufacturers and retailers are deeply concerned about customer feedback and reviews. However, the prevalence of review spam, where individuals generate fake, misleading, or dishonest reviews for personal gain, has become a pressing issue. To address this, there is a growing demand for the implementation of Review Spam detection systems to differentiate between computer generated and human written reviews. The existing model relies on the Naïve-Bayes algorithm for classification, but there is room for improvement by incorporating more accurate algorithms such as support vector machine, random forest, or logistic regression in conjunction with Countvectorizer, Hashing Vectorizer and TF-IDF scores. Developing an effective spam review detection system ensures users can trust that the reviews they rely on are genuine and not manipulated.

**Keywords :** Fake review detection, logistic regression, support vector machine, random forest, Natural language processing, Classification.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Title** | **Page No.** |
|  | Vision and Mission | ii |
|  | Program Educational Objectives | iii |
|  | Program Specific Outcomes | iv |
|  | Program Outcomes | v |
|  | Course Outcomes | vii |
|  | CO-PO Mapping | viii |
|  | Declaration | ix |
|  | Abstract | xi |
| 1. | INTRODUCTION  1.1 Introduction  1.2 Statement of the problem  1.3 Objectives  1.4 Scope  1.5 Applications  1.6 Limitations | 1-5 |
| 2. | LITERATURE SURVEY | 6-8 |
| 3. | ANALYSIS  3.1 Existing System  3.2 Proposed System  3.3 Software and Hardware Requirements | 9-13 |
| 4. | DESIGN  4.1 Process Architecture  4.2 Uml Diagrams | 14-22 |
| 5. | IMPLEMENTATION  5.1 Software tools and libraries  5.2 Development Environment  5.3 Workflow  5.4 Challenges and solutions | 23-24 |
| 6. | EXECUTION PROCEDURE AND TESTING  6.1 Dataset Description  6.2 Data Preprocessing  6.3 Data Splitting  6.4 Training  6.5 Making Prediction  6.6 Model Evaluation  6.7 Testing Overview | 25-28 |
| 7. | RESULTS AND PERFORMANCE EVALUATION  7.1 Performance Metrics  7.2 Comparative Analysis  7.3 Discussion of Results | 28-33 |
| 8. | CONCLUSION AND FUTURE WORK  8.1 Summary of Findings  8.2 Future Research Directions | 34-35 |
| 9. | APPENDIX  9.1 Program Code  9.2 List of Figures | 36-45 |
| 10. | REFERENCES | 46-47 |

**CHAPTER-1**

**INTRODUCTION**

**1.1 Introduction**

Engaging in online product transactions has become a part of our daily routines. Nowadays, a plethora of items is accessible through diverse online marketplaces. When contemplating a purchase, the majority of individuals typically first assess the product on platforms such as Amazon, Flipkart, Myntra, etc. Similarly, for travel-related activities like hotel bookings, purchasing air tickets, and various other tasks, reliance on online service providers has become common. Due to the inability to physically inspect the products or services before making a purchase, people rely on the opinions shared by others. Consequently, online reviews assume a highly crucial role in the decision-making process when it comes to procuring products or services through online channels.

Positive product reviews carry greater influence on customers compared to negative ones, significantly impacting businesses and their financial outcomes. This dynamic also opens avenues for unscrupulous individuals to exploit the situation by posting deceptive comments and reviews. These bad actors may employ fake reviews to boost their own products or undermine those of competitors. Consequently, the detection of counterfeit online reviews is of utmost importance, ensuring that users can derive genuine benefits from reviews, and companies can safeguard their reputation with consumers.

The identification of fake online reviews essentially constitutes a binary classification challenge. When the detection system is solely grounded in reviews, it is referred to as a content-based study. Classification tasks involve utilizing features extracted from the content of the reviews, encompassing the textual information. Another approach involves a reviewer-based study, known as user behavior-based classification. A fundamental challenge in all these studies is the labeling of data, particularly when relying on human assessment of content. In the

absence of reliable labeling, there is a risk of perpetuating inaccuracies in the dataset, resulting in a "garbage in, garbage out" scenario.

To navigate this challenge, recourse to semi-supervised machine learning algorithms becomes necessary, albeit at the cost of some accuracy in detection. Therefore, ensuring reliable data labeling is imperative for effective and precise review fraud detection.

To identify fake online reviews with high accuracy, we employ supervised machine learning classification methods. These techniques play a pivotal role in categorizing data, specifically utilizing content-based features for classification. Our feature set includes term-frequency and inverse document-frequency (TF-IDF) as well as count vectorizer. The classification process is executed through Random Forest classifier, Logistic Regression, and Support Vector Machine (SVM). Through this approach, our solution effectively distinguishes fake or spam reviews from authentic ones, ensuring that customers are shielded from the influence of deceptive reviews when making online purchases.

**1.2 Statement of Problem**

Fake online reviews damage trust in product feedback, affecting how confident consumers feel and the choices they make when buying. Current detection methods, mainly using Naïve-Bayes algorithms, aren't always accurate enough. To improve, we need to use more advanced algorithms like support vector machines or random forests, along with techniques such as Count Vectorizer and TF-IDF scores. This research aims to create a better system for spotting fake reviews, so consumers can trust online feedback and make informed decisions.

**1.3 Objectives**

1. To enhance trust in online product reviews by developing and implementing an advanced fake review detection system.

2. Enhancing the existing fake review detection model.

3. To empower users in the online shopping ecosystem. By ensuring the reliability of reviews, the research seeks to give users confidence in their purchasing decisions, contributing to a healthier online marketplace.

4. Continuous improvement and adaptation to the new techniques in detecting fake reviews.

**1.4 Scope**

The online review space is currently facing a trust crisis due to rampant fake reviews, which mislead consumers and damage the credibility of review platforms. Establishing sophisticated detection systems is critical to regain consumer confidence and support informed purchasing decisions. This involves advancing beyond the Naïve-Bayes algorithm to incorporate more complex algorithms like support vector machines, random forests, and logistic regression, which offer greater precision in distinguishing authentic reviews from fraudulent ones.

Improvement of the detection model also requires the adoption of various vectorization methods such as Count vectorizer, Hashing Vectorizer, and TF-IDF to accurately identify the subtleties of fake reviews. A robust and effective detection system is key to ensuring the reliability of online reviews, which in turn fosters a trustworthy e-commerce environment. This system must be holistic, taking into account the timeliness of reviews, the trustworthiness of reviewers, and the reputation of the platforms themselves.

**1.5 Applications**

**1.5.1 E-commerce Platforms:** Your system could be integrated into e-commerce platforms like Amazon, eBay, or Etsy to automatically flag suspicious reviews, improving the overall quality of product feedback and enhancing trust among consumers.

**1.5.2 Review Aggregator Websites:** Websites that aggregate reviews from various sources could benefit from your system by filtering out fake reviews, providing users with more reliable and trustworthy information when making purchasing decisions.

**1.5.3 Brand Reputation Management:** Companies can use your system to monitor and manage their online reputation by identifying and addressing fake reviews that may harm their brand image.

**1.5.4 Consumer Protection:** Regulatory bodies and consumer protection agencies could leverage your system to identify fraudulent practices and protect consumers from misleading information and deceptive advertising.

**1.5.5 Market Research:** Your project's insights into fake review patterns and detection techniques could be valuable for market research firms and companies looking to understand consumer behavior and sentiment accurately.

**1.5.6 Content Moderation:** Social media platforms and online communities could use your system to moderate user-generated content, ensuring that fake reviews do not influence public opinion or deceive users.

**1.6 Limitations**

**1.6.1 Data Imbalance:** The dataset may suffer from an imbalance between genuine and fake reviews, making it challenging to train the model effectively. Imbalanced data could lead to biases in the classification results.

**1.6.2 Evolution of Techniques:** As malicious actors continually refine their methods for generating fake reviews, our model might face challenges in keeping up with emerging tactics, requiring regular updates and adaptations.

**1.6.3 Contextual Understanding:** Our model primarily relies on textual features, potentially lacking nuanced contextual understanding. It may struggle with sarcasm, irony, or cultural references that could impact the accuracy of the classification.

**1.6.4 Dependency on Labeled Data:** Supervised machine learning requires labeled training data, and obtaining a sufficiently large and accurately labeled dataset for fake reviews can be resource-intensive and may introduce biases based on human interpretation.

**1.6.5 Generalization Across Platforms:** The model's effectiveness may vary across different online platforms, as writing styles, user behaviors, and review norms differ. Achieving a high level of generalization across diverse platforms can be challenging.

**1.6.6 Computational Resources:** Depending on the size of the dataset and the complexity of the model, the computational resources required for training and inference may be significant.

**CHAPTER-2**

**LITERATURE SURVEY**

Joni Salminen et al. [1] presented a model designed to identify untruthful online product reviews. This model leverages AI models to create a synthetic dataset of fake product reviews for the purpose of distinguishing them. The research evaluates the authenticity of machine-generated reviews and compares the efficacy of machines versus humans in identifying fraudulent reviews. However, a limitation lies in the utilization of AI models to generate fake reviews, potentially enhancing detection efficiency. However, this approach raises concerns about the quality and representativeness of the dataset, introducing biases that may affect the model's ability to generalize to new and unseen forms of fake reviews.

Sinha Anusha et al. [2] introduced a model for monitoring fake reviews using opinion mining. Their approach involves decision trees and sentiment analysis to filter reviews from a dataset of 2.2 million reviews from Flipkart. Sentiment analysis identifies positive and negative reviews, and the decision tree creates a training model for predictions. However, manually labeling reviews as fake or genuine for training data is time-consuming and may not result in an accurate model since humans struggle to classify reviews just by reading them.

Ata-Ur-Rehman et al. [3] developed a model for fake Review monitoring and removal System, which analyzes product reviews in different languages from Amazon, Flipkart offering customers accurate and original ratings. Proposed model uses an SVM classifier for categorizing the text. It might find challenging to grasp informal language, slang, and abbreviations often used in online reviews. Also, it may struggle to understand the distinct expressions.

Elmogy Ahmed et al. [4] introduced a model for detecting fake reviews through supervised machine learning, which considers both content and behavior of a review for effective classification. It uses classifiers like KNN, Naïve Bayes along with feature engineering which mainly focuses on Yelp dataset. Privacy concerns arise from analyzing and potentially storing personal data related to reviewers' behaviors. This raises ethical and legal considerations about handling personal information.

N. Ruan et al. [5] discussed the problem of deceptive opinion spam in online product reviews and explored detection methods using human computation. The study proposed a hybrid model which uses human computation along computer generated statistics in classification of reviews. While providing human assessors with linguistic metadata can enhance spam detection, there's a risk that spammers could misuse this information to adapt and create more convincing fake reviews. This poses a challenge in balancing model effectiveness and preventing misuse.

Kashti Ms Rajshri P., et al. [6] addresses the existence of fabricated reviews in the era of online shopping and underscores the significance of their identification in facilitating informed consumer choices. It proposes an active learning methodology involving training the model with real-life data through multiple iterations, constructing feature vectors and employing classifiers like Rough Set, Decision Tree. The model depends on human-labeled data for training, which can introduce bias and subjectivity in the classification process.

H.A. Najada et al. [7] focuses on detecting spam reviews in online platforms, addressing the challenge of imbalanced data where spam reviews constitute a small portion. To overcome this, a bagging-based approach is proposed. It builds balanced datasets through random under-sampling, training multiple classifiers on these sets, and using their ensemble to detect review spams. The random under-sampling method might lose important information from the majority class, creating a bias that could affect how well the model works in real situations.

M. Ott C. Cardie, et al. [8] discussed negative deceptive reviews which aim at defaming competitors, it uses a dataset of 400 reviews of 20 Chicago hotels and compares untrained human judges with n-gram-based SVM classifiers in classifying the reviews. The dataset focuses on negative deceptive reviews in the context of hotel reviews in Chicago, which may limit the generalizability of findings to other domains or positive deceptive reviews.

R. Hassan et al. [9] utilized supervised machine learning methods to categorize reviews into either fake or authentic categories, utilizing a dataset composed of hotel reviews sourced from online platforms. The research introduces a feature set that improves classification accuracy when contrasted with earlier unsupervised techniques. The efficacy of the proposed model is demonstrated with a particular dataset, and its performance is influenced by the size and diversity of the dataset.

N. Hussain H. Turab Mirza et al. [10] presented two approaches to counter review spamming. SRD-BM employs 13 spammers' behavioral features to compute a spam score, distinguishing both spammers and spam reviews. On the other hand, SRD LM concentrates on textual features, using transformation, feature extraction, and identification of fake reviews. However, the comprehensive testing of the models on diverse e-commerce platforms is lacking. The distinctive features and patterns of spam reviews on varied platforms might not be sufficiently encapsulated.

**CHAPTER-3**

**ANALYSIS**

In this chapter, we delve into the analysis of the problem domain concerning the detection of fake reviews on online products using machine learning techniques. We examine the challenges and considerations involved in addressing this issue effectively.

**3.1 Existing System**

The Naive Bayes algorithm calculates the probability that a review is fake based on the frequency of words in genuine vs. fake reviews. It assumes that the presence or absence of each word is independent of other words in the review. Using Bayes' theorem, it combines these probabilities to discriminate reviews as truthful or fake.

The steps involved in classification of product reviews using Naïve Bayes classifier are mentioned in the fig.1 below:

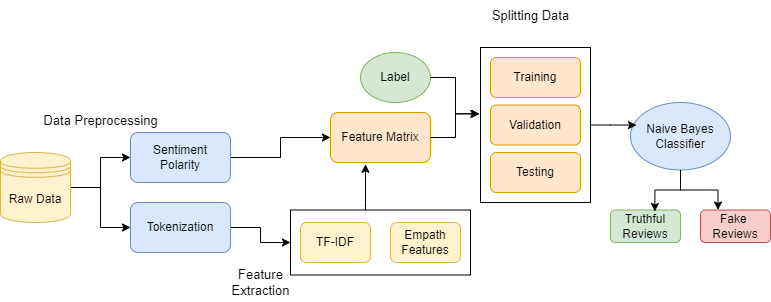


Figure. 3.1.1 Naïve Bayes Algorithm Data Flow

i) Dataset Partitioning: The dataset, consisting of 1600 examples, is partitioned into following subsets comprising train, validate, and test. The test set encompasses 20% of the whole corpus, while the remaining 80% is subdivided into the train-validation set. This train-validate set is subsequently segregated with a ratio of 75:25 to derive the train set and

validation set. Randomized sampling techniques are employed for the allocation of instances within each set.

ii) Data Preprocessing: The dataset undergoes preprocessing to tokenize the text within the review and eliminate stop words from the tokens, as these hold no individual meaning.

iii) Feature Extraction: Features like TF-IDF scores, Empathetic classifications, and sentiment orientation are employed in constructing a model that proficiently categorizes deceptive and genuine reviews. The Empath tool is applied to extract approximately 200 categories, and sentiment polarity is incorporated as an additional feature.

Sentiment Polarity: It explains the emotion or feeling of the review whether it is positive or negative.

iv) Feature Matrix Composition: The feature matrix is composed of TF-IDF scores, Empath classifications, and sentiment orientation. The data undergoes a random shuffling process prior to the division into the training, validation and test sets.

v) Model Training: The classifier undergoes training by being supplied with the feature matrix and corresponding labels.

vi) Fine Tuning: The validation data is employed for the control of overfitting and optimize various parameters of the classifier through fine tuning.

vii) Thresholding: Threshold value is set and modified after every iteration until an optimized value is reached. This process continues for entire training dataset.

viii) Classification: Based on the threshold value generated the new reviews are classified and performance is measured.

**Advantages:**

1. High Accuracy

2. Text features learning

**Disadvantages:**

1. Data Imbalance

2. Overfitting

3. Large dataset requirement

**3.2 Proposed System:**

The optimized SVM model involves a transition from a basic frequency of words to a more advanced approach using TF-IDF scoring alongside Count Vectorizer for text representation. This change will reduce the time required to calculate the frequencies of occurrences of words in review and in whole dataset, as it prepares a vector representation and stores them.

This change aims to enhance feature representation by considering the importance of words not only by their frequency in a review but also with respect to their frequency across all documents. The updated model is trained using feature vectors that incorporate both Count Vectorizer and TF-IDF scores, providing a distinct representation of textual features. These adjustments offer several benefits, including improved feature representation, enhanced discriminative power, better handling of textual data by reducing noise, increased model sensitivity to unique document aspects, and a potential reduction in overfitting, contributing to a more generalized and accurate SVM model for distinguishing between fake and genuine reviews.

We can also use Hashing Vectorizer, which uses a hash function to convert the words into feature indices directly, without maintaining an explicit vocabulary. This proves particularly beneficial when handling extensive datasets, as it circumvents the necessity to store the complete vocabulary in the system's memory. Hashing Vectorizer is computationally efficient and well-suited for online learning scenarios.

1. Collection of reviews: collecting reviews provided by the user.

2. Data preprocessing: Removal of nulls, addition of ‘length’ column, punctuation removal, case normalization, and stop word elimination using NLP tools.

3. Tokenization: Segmenting text into words and removing stop words.

Text Vectorization: Employing Count Vectorization to represent reviews as vectors and forming a matrix for all documents.

4. TF-IDF Scores: Calculating word significance scores within the dataset and storing them for classification use.

5. Feature Extraction: Setting and refining threshold values based on TF-IDF scores to differentiate between fake and genuine reviews.

6. Making Prediction or Classification: Processing reviews through tokenization and vectorization to assign TF-IDF scores and classify using a threshold value.

**Advantages:**

1. High Accuracy

2. Faster results

3. Continuous Improvement

**Disadvantages:**

1. Dependency on labelled data

2. Platform dependent

**3.3 Software and Hardware Requirements:**

Software Requirements:

• Operating System : Windows 7 , 8, 10 (64 bit)

• Software : Python 3.7

• Tools : Google Colab , Sklearn

Hardware Requirements:

• Hard Disk : 500GB and Above

• RAM : 4GB and Above

• Processor : I3 and Above

**CHAPTER-4**

**DESIGN**

**4.1 PROCESS ARCHITECTURE**

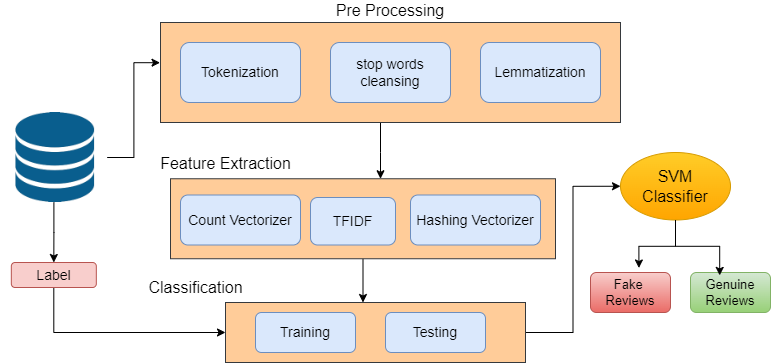


Figure 4.1.1 Architecture of SVM Model

In this chapter, we outline the design considerations and architecture of our fake review detection system. We detail the steps involved in preprocessing the data, feature extraction, model selection, and evaluation methodology.

**4.1.1 DATA PREPROCESSING**

Before training the models, we preprocess the raw review data to ensure its suitability for analysis. This involves tasks such as text normalization, tokenization, removing stop words, and handling missing values. Additionally, we balance the dataset to address any class imbalance issues, ensuring equal representation of genuine and fake reviews.

**4.1.2 FEATURE EXTRACTION**

Building upon the feature selection techniques discussed in Chapter 3, we extract features from the preprocessed text data. We employ

Count vectorizer, TF-IDF, and Hashing Vectorizer to transform the textual reviews into numerical feature vectors, which serve as input to the classification algorithms.

**4.1.3 MODEL SELECTION**

Based on the analysis conducted in Chapter 3, we select the Support Vector Machines (SVM) algorithm as the primary classifier for fake review detection.

SVM has demonstrated superior performance and robustness in handling high-dimensional data, making it well-suited for our task. We also consider Random Forest and Logistic Regression as alternative classifiers for comparison purposes.

**4.1.4 EVALUATION METHODOLOGY**

To evaluate the performance of our fake review detection system, we employ a rigorous evaluation methodology. We split the dataset into training and testing sets, ensuring that the models are trained on a representative subset of the data and evaluated on unseen instances. We use cross-validation techniques to assess the generalization capability of the models and avoid overfitting.  
  
**4.2 UML Diagrams**

Unified Modeling Language diagrams are a visual representation of a software system or a process. UML diagrams are used to model, design, and document software systems. There are several types of UML diagrams, each serving a different purpose.

The ultimate objective is for UML to become a standard language for creating models of item-based PC programming. Two notable parts of UML are included in its gift frame: a Meta-show and

documentation. Later, other sorts of methods or systems may also be added to or connected to UML.

These UML diagrams can be used in different stages of the software development life cycle, from requirements gathering to design, implementation, testing, and maintenance. They help in communicating the system design and architecture to different stakeholders, including developers, testers, project managers, and customers.

**4.2.1 USE CASE DIAGRAM:**

**Input Reviews**: Users submit reviews to the system.

**Preprocess Reviews**: The system performs tokenization, stop word removal, and normalization on the input reviews.

**Vectorize Text**: The preprocessed text is converted into numerical representations using techniques like TF-IDF and/or Count Vectorization.

**Train SVM Model**: A Support Vector Machine (SVM) model is trained using the vectorized text and labeled training data.

**Classify Review**: The trained SVM model classifies new reviews as genuine or fake.

**Evaluate Model**: The system evaluates the model's performance using metrics such as accuracy, precision, recall, and F1 score.

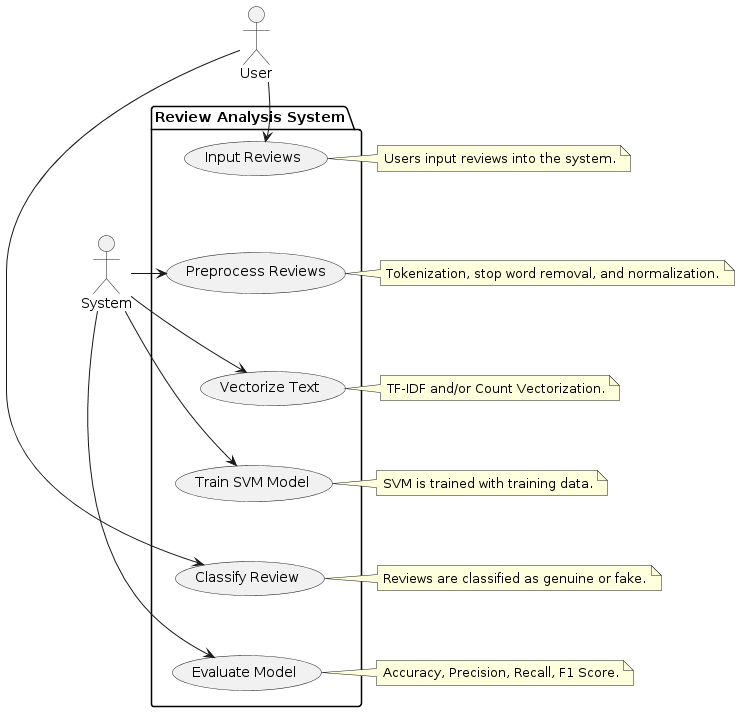


Figure 4.2.1 Use Case Diagram

**4.2.2 ACTIVITY DIAGRAM:**

The process starts with collecting reviews. We then clean them up to make them easier to understand (like taking out typos and weird symbols). Next, we turn the words into a special code that computers can understand (like turning words into numbers).

This coded data is then used to train a special machine to tell the difference between happy and sad reviews. Once trained, the machine is tested on new reviews to see if it can guess correctly. If it does well, it can be used to understand any new review that comes along!

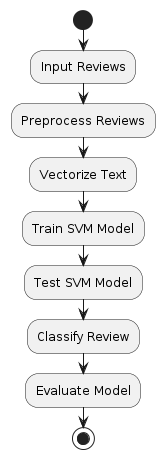
****

Figure 4.2.2 Activity Diagram

**4.2.3 SEQUENCE DIAGRAM**

Sequence Diagram Illustrates the sequence of messages or interactions between objects or components over time.

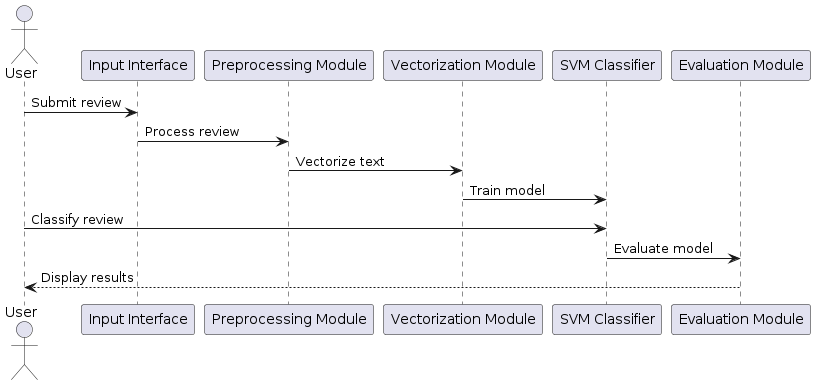


Figure 4.2.3 Sequence Diagram

The system starts by preprocessing user-submitted reviews to ensure proper formatting. Then, the review undergoes vectorization, converting text into numerical features for the SVM classifier. Using these features, the SVM classifier is trained to distinguish positive from negative sentiment in reviews, likely employing optimization techniques like gradient descent. After training, the classifier performance is evaluated on a separate dataset to assess its accuracy. With a trained and evaluated model, the system classifies new reviews, assigning sentiment and displaying results to users for informed decision-making.

**4.2.4 DEPLOYMENT DIAGRAM**

Deployment Diagram Illustrates the physical deployment of software components across hardware nodes in a system.

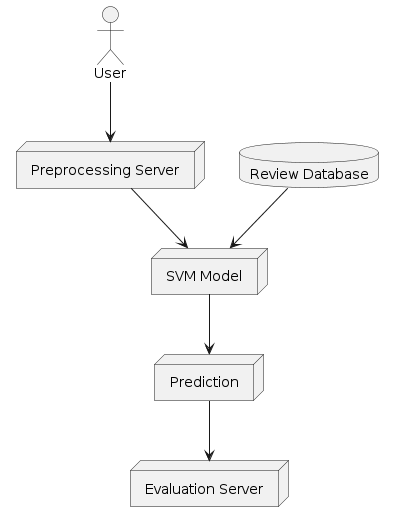


Figure 4.2.4 Deployment Diagram

The system begins with user-submitted reviews undergoing preprocessing to ensure proper formatting. Subsequently, the review is vectorized, converting text into numerical features for the SVM classifier. Utilizing these features, the SVM classifier is trained to distinguish positive from negative sentiment in reviews, potentially employing optimization techniques like gradient descent. Following training, the classifier's performance is evaluated on a separate dataset to assess its accuracy. With a trained and evaluated model, the system classifies new reviews, assigning sentiment and displaying results to users for informed decision-making.

**4.2.5 CLASS DIAGRAM**

Class Diagram Represents the static structure and relationships of classes and their members within a system.

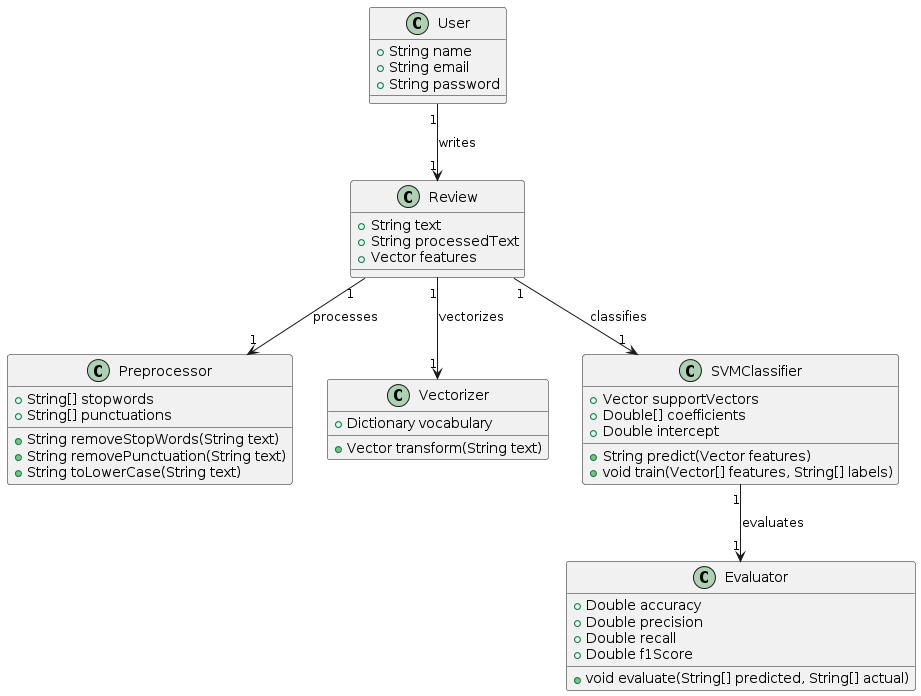


Figure 4.2.5 Class Diagram

The review analysis system initiates with text preprocessing, employing methods like removeStopWords(), removePunctuation(), and toLowerCase() to refine the input for analysis. Following this, the Vectorizer transforms the preprocessed text reviews into numerical features using transform(), facilitating compatibility with the SVM Classifier. This classifier, trained on the provided vectorized features and corresponding sentiment labels using train(), then predicts sentiment for new reviews through predict().

Once predictions are made, the system evaluates classifier performance using metrics like accuracy, precision, recall, and F1 score via the Evaluator component. This holistic approach ensures efficient sentiment analysis and continual improvement of the model's predictive capabilities.

**IMPLEMENTATION**

**CHAPTER-5**

In this chapter, we describe the implementation details of our fake review detection system. We discuss the software tools and libraries used, the development environment, and the workflow followed during the implementation phase.

**5.1 SOFTWARE TOOLS AND LIBRARIES**

We utilize popular machine learning libraries such as scikit-learn, NLTK (Natural Language Toolkit), and pandas for implementing our fake review detection system. These libraries provide robust implementations of various algorithms, feature extraction techniques, and evaluation metrics, enabling efficient development and experimentation.

**5.2 DEVELOPMENT ENVIRONMENT**

The development of our fake review detection system is carried out using Python programming language in a Google Colab environment. Google Colab offers an interactive computing environment that facilitates code development, visualization, and documentation, making it well-suited for exploratory data analysis and iterative model development.

**5.3 WORKFLOW**

The implementation of our fake review detection system follows a systematic workflow, encompassing data preprocessing, feature extraction, model training, evaluation, and result analysis. We document each step of the workflow, ensuring reproducibility and transparency in our approach.

**5.4 CHALLENGES AND SOLUTIONS**

Throughout the implementation process, we encounter various challenges such as handling large datasets, optimizing model hyperparameters, and addressing computational constraints. We devise solutions to overcome these

challenges, ensuring the successful implementation and deployment of our fake review detection system.

**CHAPTER-6**

**EXECUTION PROCEDURE AND TESTING**

In this chapter, we detail the execution procedure and testing methodology utilized to validate the efficacy of our fake review detection system. We outline the steps taken to execute the system, evaluate its performance, and validate its accuracy in distinguishing between genuine and fake reviews.

**6.1 DATASET DESCRIPTION**

The dataset utilized in the proposed model is amazon reviews dataset which contains reviews labeled as computer generated and human written. A balance is maintained in the dataset with a total of 20,000 artificially generated reviews and 20,000 human written reviews, ensuring impartiality and preventing bias toward a single class during the training process.

**6.2 DATA PREPROCESSING**

Null values from the dataset are dropped and a new column ‘length’ is added which defines the length of each review. A function is used to remove the punctuations from the text, convert the text into lowercase and remove the stop words with help of natural language processing tools.

**6.3 DATA SPLITTING**

The pre-processed data is partitioned into training set and validation set with 80% of the data being used to train the model and the 20% of data being used to validate the model and measure performance metrics like accuracy, precision, f1 score …etc.

**6.4 TRAINING**

The training process includes random assumption of threshold values which are tf-idf scores and validating them during the iterations of training data which makes the model to be able to discover the relationships and hidden patterns in the review data along with labels.

**6.4.1 Tokenization:** In this process the review text is divided into individual words and stop words are removed.

**6.4.2 Text Vectorization:** Count vectorization stands as a widely employed technique for text vectorization, wherein every review is depicted as a vector. In this representation, each element of the vector signifies the count of a specific word within the review. The aggregation of all document vectors results in the formation of a matrix, wherein each row corresponds to a document, and each column corresponds to a distinct word across the entire corpus. An alternative method, instead of Count Vectorization, involves converting words into hash values using a function. These hash values are stored, offering memory efficiency and faster processing, as opposed to storing the complete vocabulary.

**6.4.3 TF-IDF scores:** Term Frequency – Inverse Document Frequency (TF-IDF) serves as a metric indicating the significance of a word within the entire dataset. The computation of these scores for each word results in a dictionary, where they are stored for subsequent retrieval during the classification process.

**6.4.4 Feature Extraction:** Based on these tf-idf scores the proposed model will assume some threshold value which acts as a bar dividing the fake reviews and genuine reviews. This value is updated after training with each dataset i.e., in iterative manner, during the iterations the proposed model will identify the relationship and hidden patterns in the scores of words along with labels. This will help in classification of reviews.

**6.5 MAKING PREDICTION OR CLASSIFICATION**

A review is taken as input for classification. The review text undergoes various steps during the classification process. The review is tokenized i.e., breaking down the review into simple words and then punctuations and stop-words are removed with help of natural language processing tools.

Either through Count Vectorization or Hashing Vectorization, the review is transformed into a vector. Each element of this vector denotes the count of a specific word within the review. Subsequently, individual words receive corresponding TF-IDF scores, contributing to an overall score computed for each review. The determination of whether a review is spam or truthful is then made with the assistance of a threshold value.

**6.6 MODEL EVALUATION**

The assessment of the proposed model involves predicting the class of the test or validation data and evaluating the accuracy based on the pre-defined class labels of the reviews.

**6.7 Testing Overview:** Testing is a critical stage in software development that checks the system’s functionality and performance, ensuring it meets the required standards and operates correctly.

Functional Testing: This process involves testing each software module to confirm it performs its designated function, using test cases that cover a range of inputs and scenarios.

**6.7.1 Performance Testing:** This type of testing measures the system’s response and efficiency under various conditions, identifying any performance-related issues such as bottlenecks or resource limitations.

**6.7.2 Program Testing:** Focuses on finding syntax errors, which are typically caught by compilers, and logic errors, which occur when the program’s behavior doesn’t match the intended design or requirements.

**6.7.3 Validation Testing:** Verifies that the software meets the user’s needs and business goals, functioning as expected, through a series of black-box tests against acceptance criteria.

**CHAPTER-7**

**RESULTS AND PERFORMANCE EVALUATION**

In this chapter, we present the results of our fake review detection system's performance evaluation. We analyze the effectiveness of the implemented algorithms and discuss the achieved accuracy in distinguishing between genuine and fake reviews.

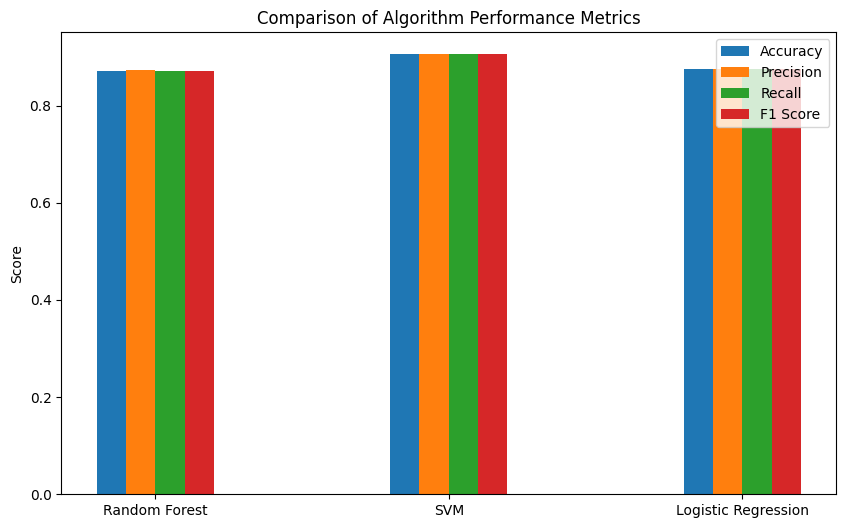
**7.1 PERFORMANCE METRICS**

We begin by reporting the performance metrics obtained during the testing phase. These metrics include accuracy, precision, recall, and F1-score for each classification algorithm used in the fake review detection system. Accuracy represents the overall correctness of the classification results, while precision measures the proportion of correctly classified fake reviews out of all reviews predicted as fake. Recall calculates the percentage of correctly identified fake reviews out of all actual fake reviews, and F1-score provides a balanced measure of precision and recall.

|  |  |
| --- | --- |
| Figure 7.1.1 | Figure 7.1.2 |
| Figure 7.1.3 | Figure 7.1.4 |
| Figure 7.1.5 | Figure 7.1.6 |

**7.2 COMPARATIVE ANALYSIS**

Next, we conduct a comparative analysis of the performance of different classification algorithms – Support Vector Machines (SVM), Random Forest, and Logistic Regression. We compare their accuracy scores and other evaluation metrics to identify the most effective algorithm for fake review detection. Insights gained from this analysis help in understanding the strengths and weaknesses of each algorithm in the context of the problem domain.

Figure 7.2.1 Comparison of Algorithms performance

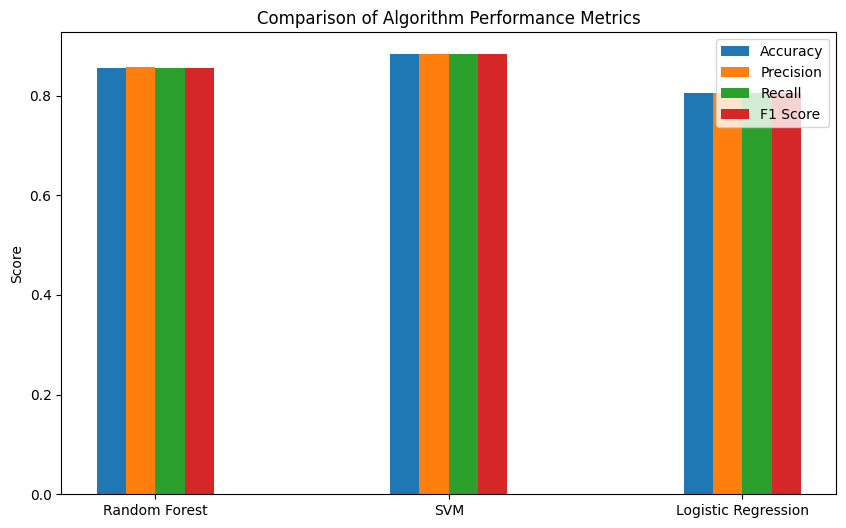


Figure 7.2.2 Comparison of Algorithms performance

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset** | **Features** | **Algorithm** | **Accuracy** |
| **Amazon reviews dataset** | **TF-IDF scores with Count Vectorizer** | Support Vector Machine | 91% |
| Random Forest | 87% |
| Logistic Regression | 86% |
| **TF-IDF scores with Hashing Vectorizer** | Support Vector Machine | 88% |
| Random Forest | 85% |
| Logistic Regression | 82% |

Table 7.2.1 Results Table

**7.3 DISCUSSION OF RESULTS**

In the pursuit of fake review detection, experiments were conducted on a substantial and balanced dataset comprising 40,000 reviews sourced from the Amazon platform. This dataset consists of a meticulously curated set of 20,000 fake reviews and an equal number of genuine reviews. The balance in the dataset ensures a comprehensive evaluation of our models' performance, allowing us to effectively discern their ability to identify both deceptive and authentic reviews.

Techniques like TF-IDF scoring and different vectorization methods (Count vectorizer and Hashing Vectorizer) are used to create strong features for our models. This was crucial in capturing subtle language nuances, helping our models effectively distinguish between genuine and fake reviews.

We applied three algorithms namely random forest algorithm, SVM algorithm, and logistic regression to our diverse Amazon reviews dataset. SVM stood out with remarkable accuracy, proving its effectiveness in handling the complexities of the dataset. The dataset's ample size and balance allowed us to thoroughly evaluate

the models, confirming their ability to tackle real-world challenges posed by fake reviews. The strong performance of SVM in accurately identifying both real and fake reviews highlights the strength of our approach. This success points to exciting possibilities for future research to improve and advance fake review detection systems.

**CHAPTER-8**

**CONCLUSION AND FUTURE WORK**

In this final chapter, we summarize the key findings and contributions of our fake review detection project. We reflect on the achievements, discuss the implications of our work, and outline potential avenues for future research and development.

**8.1 SUMMARY OF FINDINGS**

Detecting fake reviews poses challenges due to imbalanced datasets, where fake reviews are a small fraction. This imbalance can affect model performance. Additionally, selecting features for machine learning relies on subjective judgment, leading to the inclusion of potentially irrelevant features for certain classification methods. Addressing the challenges associated with imbalanced datasets, the SVM model has been fine-tuned to provide robust performance, even when faced with a limited number of fake reviews. This optimization is crucial for real-world applications where computer generated reviews constitute only a small amount of the total corpus.

The optimized SVM model provides accurate classification of reviews. The model incorporates a combination of Countvectorizer, Hashing Vectorizer, and TF-IDF techniques, showcasing a comprehensive approach to feature extraction and representation. This combination enhances the model's capacity to proficiently classify fake reviews effectively.

**8.2 FUTURE RESEARCH DIRECTIONS**

Looking ahead, we outline several potential avenues for future research and development in the field of fake review detection:

Refinement of Algorithms: Further optimization of machine learning algorithms and feature extraction techniques to improve the accuracy and efficiency of fake review detection systems.

Incorporation of Domain-specific Knowledge: Integration of domain-specific knowledge and contextual information to enhance the understanding of review semantics and improve classification accuracy.

Exploration of Deep Learning Techniques: Investigation of deep learning architectures, such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs), for more advanced sentiment analysis and fake review detection.

Real-time Detection: Development of real-time fake review detection systems capable of identifying fraudulent reviews as they are posted, enabling timely intervention and mitigation.

Ethical Considerations: Exploration of ethical implications related to the use of automated fake review detection systems, including privacy concerns and potential biases in algorithmic decision-making.

**CHAPTER-9**

**APPENDIX**

**PROGRAM CODE**

import numpy as np

import pandas as pd

import string

from sklearn.model\_selection import train\_test\_split, GridSearchCV

from sklearn.metrics import accuracy\_score

from sklearn.ensemble import RandomForestClassifier, VotingClassifier

from sklearn.svm import SVC

from sklearn.linear\_model import LogisticRegression

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

from sklearn.pipeline import Pipeline

from nltk.corpus import stopwords

import nltk

from sklearn.utils import parallel\_backend

# Download stopwords if not already downloaded

nltk.download('stopwords')

dataframe\_main = pd.read\_csv('/content/drive/MyDrive/dataset - ac.csv')

fraction\_to\_keep = 1

# Reduce the dataset size by selecting a random sample

dataframe = dataframe\_main.sample(frac=fraction\_to\_keep, random\_state=42)

#dataframe.drop('Unnamed: 0', axis=1, inplace=True)

dataframe.dropna(inplace=True)

dataframe['length'] = dataframe['text\_'].apply(len)

def convertmyTxt(rv):

np = [c for c in rv if c not in string.punctuation]

np = ''.join(np)

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

# Split the data into training and testing sets

x\_train, x\_test, y\_train, y\_test = train\_test\_split(dataframe['text\_'], dataframe['label'], test\_size=0.2)

# print(x\_train.head())

# print("ytefjwodfnsdif")

# print(x\_test.head())

#random forest algorithm

pip\_rf = Pipeline([

('bow', CountVectorizer(analyzer=convertmyTxt)),

('tfidf', TfidfTransformer()),

('classifier', RandomForestClassifier())

])

pip\_rf = Pipeline([

('hashing', HashingVectorizer(analyzer=convertmyTxt, n\_features=1000)), # Specify the number of features

('tfidf', TfidfTransformer()),

('classifier', RandomForestClassifier())

])

# Train the Random Forest model

pip\_rf.fit(x\_train, y\_train)

# Make Random Forest predictions

rf\_predictions = pip\_rf.predict(x\_test)

# Print Random Forest accuracy

print('Random Forest Accuracy:', str(np.round(accuracy\_score(y\_test, rf\_predictions) \* 100, 2)) + '%')

#support vector machine algorithm

pip\_svm = Pipeline([

('bow', CountVectorizer(analyzer=convertmyTxt)),

('tfidf', TfidfTransformer()),

('classifier', SVC())

])

pip\_svm = Pipeline([

('hashing', HashingVectorizer(analyzer=convertmyTxt, n\_features=1500)), # Specify the number of features

('tfidf', TfidfTransformer()),

('classifier', SVC())

])

# grid\_svm = GridSearchCV(pip\_svm, param\_grid\_svm, cv=5, n\_jobs=-1)

# grid\_svm.fit(x\_train, y\_train)

# grid\_svm = GridSearchCV(pip\_svm, param\_grid\_svm, cv=5)

# grid\_svm.fit(x\_train, y\_train)

# best\_params\_svm = grid\_svm.best\_params\_

# print("Best Parameters (SVM):", best\_params\_svm)

# # Update the SVM pipeline with the best parameters

# pip\_svm.set\_params(\*\*best\_params\_svm)

# Train the SVM model

pip\_svm.fit(x\_train, y\_train)

# Make SVM predictions

svm\_predictions = pip\_svm.predict(x\_test)

# Print SVM accuracy

print('SVM Accuracy:', str(np.round(accuracy\_score(y\_test, svm\_predictions)\*100,2))+'%')

#logistic regression algorithm

pip\_lr = Pipeline([

('bow', CountVectorizer(analyzer=convertmyTxt)),

('tfidf', TfidfTransformer()),

('classifier', LogisticRegression())

])

pip\_lr = Pipeline([

('hashing', HashingVectorizer(analyzer=convertmyTxt, n\_features=1000)), # Specify the number of features

('tfidf', TfidfTransformer()),

('classifier', LogisticRegression())

])

# Train the Logistic Regression model

pip\_lr.fit(x\_train, y\_train)

# Make Logistic Regression predictions

lr\_predictions = pip\_lr.predict(x\_test)

# Print Logistic Regression accuracy

print('Logistic Regression Accuracy:', str(np.round(accuracy\_score(y\_test, lr\_predictions)\*100,2))+'%')

import matplotlib.pyplot as plt

import seaborn as sns

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

# cm\_nb = confusion\_matrix(y\_test, nb\_predictions)

# plt.figure(figsize=(6, 4))

# sns.heatmap(cm\_nb, annot=True, fmt='d', cmap='Blues', xticklabels=['Negative', 'Positive'], yticklabels=['Negative', 'Positive'])

# plt.title('Confusion Matrix (Naive Bayes)')

# plt.xlabel('Predicted')

# plt.ylabel('Actual')

# plt.show()

# Calculate and print classification report for Random Forest

print('Classification Report (Random Forest):\n', classification\_report(y\_test, rf\_predictions))

# Draw confusion matrix for Random Forest

cm\_rf = confusion\_matrix(y\_test, rf\_predictions)

plt.figure(figsize=(6, 4))

sns.heatmap(cm\_rf, annot=True, fmt='d', cmap='Blues', xticklabels=['Negative', 'Positive'], yticklabels=['Negative', 'Positive'])

plt.title('Confusion Matrix (Random Forest)')

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.show()

# Calculate and print classification report for SVM

print('Classification Report (SVM):\n', classification\_report(y\_test, svm\_predictions))

# Draw confusion matrix for SVM

cm\_svm = confusion\_matrix(y\_test, svm\_predictions)

plt.figure(figsize=(6, 4))

sns.heatmap(cm\_svm, annot=True, fmt='d', cmap='Blues', xticklabels=['Negative', 'Positive'], yticklabels=['Negative', 'Positive'])

plt.title('Confusion Matrix (SVM)')

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.show()

# Calculate and print classification report for Logistic Regression

print('Classification Report (Logistic Regression):\n', classification\_report(y\_test, lr\_predictions))

# Draw confusion matrix for Logistic Regression

cm\_lr = confusion\_matrix(y\_test, lr\_predictions)

plt.figure(figsize=(6, 4))

sns.heatmap(cm\_lr, annot=True, fmt='d', cmap='Blues', xticklabels=['Negative', 'Positive'], yticklabels=['Negative', 'Positive'])

plt.title('Confusion Matrix (Logistic Regression)')

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.show()

from sklearn.metrics import precision\_score, recall\_score, f1\_score

# # Calculate and print performance metrics for Naive Bayes

# precision\_nb = precision\_score(y\_test, nb\_predictions, average='weighted')

# recall\_nb = recall\_score(y\_test, nb\_predictions, average='weighted')

# f1\_nb = f1\_score(y\_test, nb\_predictions, average='weighted')

# print('Naive Bayes Metrics:')

# print('Precision:', precision\_nb)

# print('Recall:', recall\_nb)

# print('F1 Score:', f1\_nb)

# Calculate and print performance metrics for Random Forest

precision\_rf = precision\_score(y\_test, rf\_predictions, average='weighted')

recall\_rf = recall\_score(y\_test, rf\_predictions, average='weighted')

f1\_rf = f1\_score(y\_test, rf\_predictions, average='weighted')

print('\nRandom Forest Metrics:')

print('Precision:', precision\_rf)

print('Recall:', recall\_rf)

print('F1 Score:', f1\_rf)

# Calculate and print performance metrics for SVM

precision\_svm = precision\_score(y\_test, svm\_predictions, average='weighted')

recall\_svm = recall\_score(y\_test, svm\_predictions, average='weighted')

f1\_svm = f1\_score(y\_test, svm\_predictions, average='weighted')

print('\nSVM Metrics:')

print('Precision:', precision\_svm)

print('Recall:', recall\_svm)

print('F1 Score:', f1\_svm)

# Calculate and print performance metrics for Logistic Regression

precision\_lr = precision\_score(y\_test, lr\_predictions, average='weighted')

recall\_lr = recall\_score(y\_test, lr\_predictions, average='weighted')

f1\_lr = f1\_score(y\_test, lr\_predictions, average='weighted')

print('\nLogistic Regression Metrics:')

print('Precision:', precision\_lr)

print('Recall:', recall\_lr)

print('F1 Score:', f1\_lr)

import matplotlib.pyplot as plt

import numpy as np

# Accuracy values

# accuracy\_values = [accuracy\_score(y\_test, nb\_predictions),

# accuracy\_score(y\_test, rf\_predictions),

# accuracy\_score(y\_test, svm\_predictions),

# accuracy\_score(y\_test, lr\_predictions)]

accuracy\_values1 = [accuracy\_score(y\_test, rf\_predictions),

accuracy\_score(y\_test, svm\_predictions),

accuracy\_score(y\_test, lr\_predictions)]

# Precision values

precision\_values = [ precision\_rf, precision\_svm, precision\_lr]

# Recall values

recall\_values = [ recall\_rf, recall\_svm, recall\_lr]

# F1-score values

f1\_values = [ f1\_rf, f1\_svm, f1\_lr]

# Algorithms

algorithms = [ 'Random Forest', 'SVM', 'Logistic Regression']

# Plotting the comparison graph

fig, ax = plt.subplots(figsize=(10, 6))

bar\_width = 0.1

bar\_positions = np.arange(len(algorithms))

ax.bar(bar\_positions - 1.5 \* bar\_width, accuracy\_values1, bar\_width, label='Accuracy')

ax.bar(bar\_positions - 0.5 \* bar\_width, precision\_values, bar\_width, label='Precision')

ax.bar(bar\_positions + 0.5 \* bar\_width, recall\_values, bar\_width, label='Recall')

ax.bar(bar\_positions + 1.5 \* bar\_width, f1\_values, bar\_width, label='F1 Score')

ax.set\_xticks(bar\_positions)

ax.set\_xticklabels(algorithms)

ax.set\_ylabel('Score')

ax.set\_title('Comparison of Algorithm Performance Metrics')

ax.legend()

plt.show()

**LIST OF FIGURES**

**Figure No. Title Page No.**

4.1.1 Architecture of SVM Model 14

4.2.1 Use Case Diagram 17

4.2.2 Activity Diagram 18

4.2.3 Sequence Diagram 19

4.2.4 Deployment Diagram 20

4.2.5 Class Diagram 21

7.1.1 Confusion matrix(Random Forest

Algorithm) 29

7.1.2 Confusion matrix(Support Vector

Machine Algorithm) 29

7.1.3 Confusion matrix(Logistic Regression

Algorithm) 30

7.1.4 Confusion matrix(Random Forest

Algorithm) 30

7.1.5 Confusion matrix(Support Vector

Machine Algorithm) 30

7.1.6 Confusion matrix(Logistic Regression

Algorithm) 30

7.2.1 Comparison of Algorithms Performance 31

7.2.2 Comparison of Algorithms Performance 31

**10. REFERENCES**

[1] Joni Salminen, Chandrashekhar Kandpal, Ahmed Mohamed Kamel, Soon-gyo Jung, Bernard J. Jansen, Creating and detecting fake reviews of online products, Journal of Retailing and Consumer Services, Volume 64, 2022, 102771, ISSN 0969-6989, <https://doi.org/10.1016/j.jretconser.2021.102771>.

[2] Ata-Ur-Rehman et al., "Intelligent Interface for Fake Product Review Monitoring and Removal," 2019 16th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), Mexico City, Mexico, 2019, pp. 1-6, Doi: 10.1109/ICEEE.2019.8884529.

[3] Sinha, Anusha, et al. "Fake product review monitoring using opinion mining." International Journal of Pure and Applied Mathematics 119.12 (2018): 13203-13209.

[4] Elmogy, Ahmed & Tariq, Usman & Mohammed, Ammar & Ibrahim, Atef. (2021). Fake Reviews Detection using Supervised Machine Learning. International Journal of Advanced Computer Science and Applications. 12. 10.14569/IJACSA.2021.0120169.

[5] N. Ruan, R. Deng, and C. Su, ‘‘GADM: Manual fake review detection for O2O commercial platforms,’’ Comput. Secur., vol. 88, Jan. 2020, Art. no. 101657.

[6] Kashti, Ms Rajshri P., and Prakash S. Prasad. "Enhancing NLP techniques for fake review detection." Int. Res. J. Eng. Technol. (IRJET) 6 (2019): 241-245.

[7] H. A. Najada and X. Zhu, ‘‘ISRD: Spam review detection with imbalanced data distributions,’’ in Proc. IEEE 15th Int. Conf. Inf. Reuse Integr. (IEEE IRI), Redwood City, CA, USA, Aug. 2014, pp. 553–560.

[8] M. Ott, C. Cardie, and J. T. Hancock, ‘‘Negative deceptive opinion spam,’’ in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technologie (HLT-NAACL), Atlanta, GA, USA, Jun. 2013, pp. 497–501.

[9] R. Hassan and M. R. Islam, "A Supervised Machine Learning Approach to Detect Fake Online Reviews," 2020 23rd International Conference on Computer and Information Technology (ICCIT), DHAKA, Bangladesh, 2020, pp. 1-6, Doi: 10.1109/ICCIT51783.2020.9392727.

[10] N. Hussain, H. Turab Mirza, I. Hussain, F. Iqbal and I. Memon, "Spam Review Detection Using the Linguistic and Spammer Behavioural Methods," in IEEE Access, vol. 8, pp. 53801-53816, 2020, Doi: 10.1109/ACCESS.2020.2979226.