A Major Project On

Fake Review Detection using Machine Learning

Submitted in partial fulfillment of the requirements for the award of the

Bachelor of Technology

In

Department of Computer Science and Engineering

By

Amar Aniketh Varma	**18241A0563**
Vishnu Vardhan Reddy	**18241A0584**
Mohammed Aftab Ahmed	**18241A0596**
Sumant Kumar Sharma	**18241A05B2**

Under the Esteemed guidance of

Dr. G Charles Babu - Professor

Department of Computer Science and Engineering

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY (Approved by AICTE, Autonomous under JNTUH, Hyderabad, Bachupally, Kukatpally, Hyderabad-500090

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY (Approved by AICTE, Autonomous under JNTUH, Hyderabad, Bachupally, Kukatpally, Hyderabad-500090

CERTIFICATE

This is to certify that the major project entitled "**Fake Review Detection using Machine Learning Algorithms**" is submitted by **Amar Aniketh Verma (18241A0563), Vishnu Vardhan Reddy (18241A0584), Mohammed Aftab Ahmed (18241A0596), Sumant Kumar Sharma (18241A05B2)** in partial fulfillment of the award of degree in BACHELOR OF TECHNOLOGY in Computer Science and Engineering during academic year 2020-2021.

INTERNAL GUIDE

Dr. G Charles Babu

(Professor)

HEAD OF THE DEPARTMENT

Prof. Dr. K. MADHAVI

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

There are many people who helped us directly and indirectly to complete our project successfully. We would like to take this opportunity to thank one and all. First we would like to express our deep gratitude towards our internal guide **Dr. Charles Babu, Prof.** Department of CSE for his support in the completion of our dissertation. We wish to express our sincere thanks to **Dr. K. Madhavi, HOD, Department of CSE** and to our principal **Dr. J. Praveen** for providing the facilities to complete the dissertation. We would like to thank all our faculty and friends for their help and constructive criticism during the project period. Finally, we are very much indebted to our parents for their moral support and encouragement to achieve goals.

Amar Aniketh Varma (18241A0563)
Vishnu Vardhan Reddy (18241A0584)
Mohammed Aftab Ahmed (18241A0596)
Sumant Kumar Sharma (18241A05B2)

DECLARATION

We hereby declare that the industrial major project entitled **"Fake review Detection using Machine Learning"** is the work done during the period from **1th March 2021 to 21st June 2022** and is submitted in the partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering from Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous under Jawaharlal Nehru Technology University, Hyderabad). The results embodied in this project have not been submitted to any other university or Institution for the award of any degree or diploma.

Amar Aniketh Varma (18241A0563)
Vishnu Vardhan Reddy (18241A0584)
Mohammed Aftab Ahmed (18241A0596)
Sumant Kumar Sharma (18241A05B2)

ABSTRACT

With the advent of E-commerce, many products are being sold online in a global marketplace. People can buy, sell and leave feedback in the form of a review for other customers to see and make an informed decision about purchasing a particular product. Due to his, customers are increasingly relying on product reviews for information. Reviews have a direct impact on the sales of a product. But fake reviews, on the other hand, reduce the value of online reviews by painting an untruthful picture of product quality. Thus, people get the wrong idea when they come across such a review. As a result, fake review detection is required. This paper is aimed at developing a machine learning model that can detect a fake review from a real one by using review and reviewer centric features which is then integrated with a web application.

User reviews can play a big part in deciding an organization's revenue in e-commerce. Before purchasing any product or service, online users consult reviews. Since a result, the trustworthiness of online reviews is critical for organizations, as it can have a direct impact on their reputation and profitability. As a result, some companies pay spammers to write phony reviews. These fake reviews take advantage of people's purchasing decisions.
As a result, in the last twelve years, strategies for detecting phony reviews have been actively investigated. Machine learning algorithms are proposed to analyses the primary review-centric features in order to detect bogus reviews.

TABLE OF CONTENTS

Chapter-1

INTRODUCTION

As reviews become more popular on social media platforms, there is no way to evaluate which user-generated content is credible or which source is trustworthy. The dissemination of such disinformation has negative implications, causing harm to both users and businesses. The main goal is to present an examination of the main approaches to detecting fraudulent reviews that have been proposed, particularly those that use machine learning techniques. When detecting bogus reviews, review-centric sites such as Yelp can be evaluated. The supervised machine learning approaches take into account several features derived from the review's text. Yelp has made a publicly available big scale and created dataset available, which contains reviews that are classified using well-known supervised classifiers that analyze numerous elements of the data to split the reviews as true or deceptive.

Rationale

In general, e-commerce sites allow customers to leave feedback about their services. The fact that these assessments exist can be used as a source of data. Companies, for example, can use it to create their products or services, while potential customers can use it to decide whether to buy or use a product. Unfortunately, the value of a review has been abused by some parties who have attempted to produce fake reviews in order to boost the popularity of a product or to discredit it.

Goal

Our current research is focused on two main areas:

- Understanding patterns of fake / illegitimate reviews that could aid in recognizing genuine reviewers.
- Detecting false reviews in an E-commerce environment.

Objective

The goal of our research is:

- To predict fraudulent reviews and classify them based on their characteristics.
- Sort reviews into categories based on their review, review centric and metadata features.
- Analysis of Fake and Real reviews.

Methodology

Machine learning:

The automated recognition of meaningful patterns in data is referred to as machine learning. It has become a common technique in practically every endeavor that demands information extraction from massive data sets over the last few decades. Machine learning, which is generally split into supervised, semi-supervised, and unsupervised learning, plays a crucial role in detecting false reviews. A supervised learning system can be used to determine whether or not reviews are genuine. The majority of studies use logistic regression and SVM as learning models. Deep learning models have also been the subject of recent research. In our study, we put all of these models to the test and compare their results. As features, we employ unigram and bigram, and as a classification algorithm, we use support vector machine.

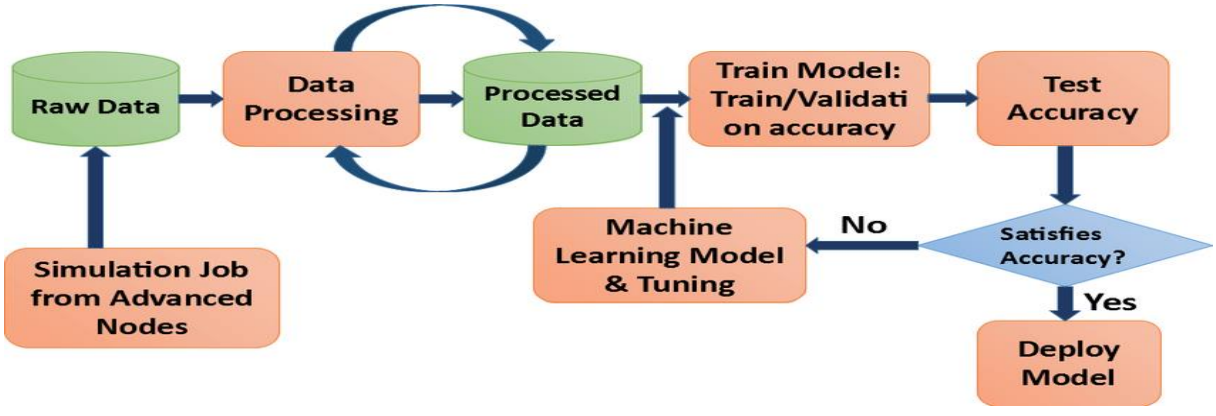

Fig 1.1-Machine learning process

Our Dataset

Dataset which we are using are the reviews of the users on certain products and we categorize them based on some features for detecting fake reviews.

In our dataset, we have 21000 Columns (10500 True & 10500 False).

We then select the columns we need.

Attributes of this dataset

- Reviewer ID
- Product ID
- Rating Label
- Average Rating
- Rating
- Deviation
- Date
- Review Text

Preprocessing and Cleaning

To deal with missing, noisy, and inconsistent data, we apply a variety of pre-processing approaches at this step. Uppercase letter and Unicode character removal, Lemmatization, Tokenization, bag of words model, and stop-words removal are among the pre-processing techniques available.

We also clean and normalize the data to remove any outliers, Null values, or incorrect data. This is done because any form of unclean data or inconsistent data will hamper the learning process in turn reducing the model's accuracy.

Cleaning is necessary so that the data is consistent and our model can be properly trained without bias.

N-Gram Analysis:

An N-gram can be defined as a continuous sequence of n items from a given sample of text. Depending on the application, the elements can be words, language corpus or letters. The word input string used in this technique is usually extracted from a corpus of text or voice. N-grams are one technique to aid machines in comprehending a word in context to gain a better understanding of its meaning. "We need to book our tickets as soon as possible," versus "We need to study this book as soon as possible." Because the word "book" is employed as a verb, it is an action. The noun "book" is used in the latter case. The n in n-grams stands for the number of words you want to look at. Unigram is a model that just considers the frequency of a word and ignores previous words. Bigram is a model that predicts the current word only based on the previous word. If two previous words are considered, the model is a trigram.

Text	N-gram
Data	1-gram
Great information	2-gram
I am fine	3-gram
Nice to meet you	4-gram

Figure 1.2-Example of n-grams

SVM (Support Vector Machine):

SVM is used for both classification and regression. Though we might also argue regression difficulties, categorization is the best fit. The classification of datapoints is done by locating a hyperplane in an N-dimensional space. After training the model when a new data point is input, it can classify it into its correct class / category. The size of the hyperplane is determined by the number of features of the data. It is one of the most successful ML algorithms in the constraint of limited training data.

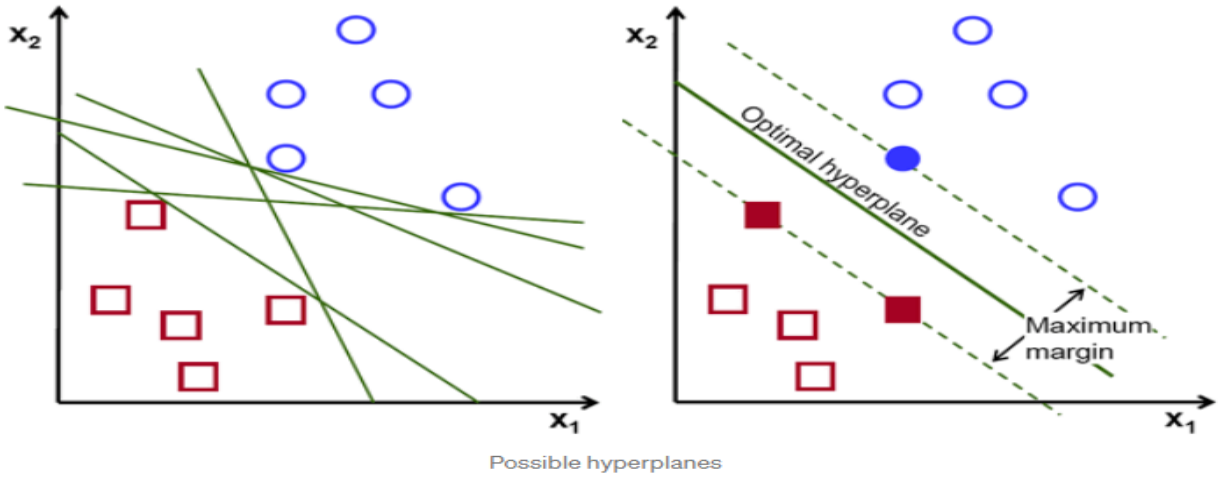

Figure 1.2- Support Vector Machine

Passive Aggressive Classifier:

Passive Aggressive Classifiers belong to the Online ML Model where data is input in a consecutive order. Over here, the ML model is updated step-by-step. This is different from other learning models where data comes in form of batches and is used all at once. An online-learning system will simply obtain a training example, update the classifier, and then discard the sample. This classifier highly useful in places where there is a large amount of data and training the full dataset is computationally very difficult due to the size of the data.

Role and Responsibilities

Role	Name	Responsibilities
Data Entry & Tester	Aftab Ahmed	• Data Entry • Documentation & Presentation • Testing
Data Entry & Tester	Sumant Kumar Sharma	• Data Entry • Documentation & Presentation • Testing
Coding & Research	Amar Aniketh Varma	• Coding • Documentation • Presentation • Research
Coding & Research	Vishnu Vardhan Reddy	• Coding • Dataset • Testing • Presentation

Contribution of Project

Market potential:

The use of AI and machine learning to detect fraudulent reviews is already in use, has been demonstrated to function, and is anticipated to grow further. The use of AI/ML to predict false reviews has potential, however it is still a work in progress due to new reviews submitted by actual individuals. A product's sale can be made or broken by a review. As a result, we must be able to distinguish between fraudulent and genuine evaluations in a proper manner. However, humans are unable to tell whether a review is genuine or not merely by glancing / having a look at it. As a result, we must dig deeper to find patterns.

Innovativeness:

The idea behind this research is that legitimate user reviews can be predicted (to a certain extent); all it takes is the ability to go through a large amount of data to uncover patterns that can be used to distinguish bogus from real reviews. This type of data analysis would have been technologically unfeasible only a few decades ago, but current advances in machine learning may be up to the challenge.

Usefulness:

Companies can use the reviews to figure out what their products and services' strengths and flaws are from the perspective of their customers. They can use these findings in the future to increase consumer satisfaction and encourage more people to purchase their products. Furthermore, they can look into customer reviews of competitor's items to learn about the general hunger of customers: their preferences, priorities, and so on. Producers can improve their own products by adding or magnifying desirable qualities of competitor products, according to customers' perceptions. Furthermore, by becoming aware of discreditable features of competitors' products, manufacturers will ensure that those features are not emphasized in their production or that flaws are addressed before buyers notice them.

Report Organization

The report's remaining sections are organized as follows:

Chapter 2 contains the Functional and Non-Functional requirements

Chapter 3 contains the project's analysis and design

Chapter 4 contains the project's construction and implementation details

Chapter 5 contains the project's conclusion, future scope and future applicability.

Chapter-2

REQUIREMENT ENGINEERING

Functional Requirements:

The functional requirements define the application's essential functionality.

Interface Requirement:

- Screen 1 shows project abstract.
- Button 1 for login page.
- Screen 2 shows login page.
- Field 1 inputs username.
- Field 2 inputs password.
- Button 1 for Analysis page.
- Screen 3 accepts user inputs.
- Field 1 inputs Review text data.
- Field 2 inputs review rating.
- Field 3 inputs Verified Purchase.
- Field 4 inputs category.
- Submit Button to send user data to model and display output.
- Screen 4 displays model output.

Non-Functional Requirements:

Non-functional requirements are the requirements of the system which are not directly concerned with specific functionality delivered. It rather specifies the quality aspect of a system (usually software). They may be related to emergent properties such as reliability, usability etc.

- To provide maximum accuracy
- Provide precise analysis
- Ease of use
- Reliability
- Availability
- Maintainability

Chapter-3

ANALYSIS AND DESIGN

Class diagram:

A class diagram (part of the Unified Modeling Language (UML)) is a static structural diagram used in software engineering to describe the structure of a system by exhibiting the system's classes, properties, actions (or methods), and linkages between objects.

The objective of a class diagram is to show the static perspective of an application. The only diagrams that can be directly mapped with object-oriented languages are class diagrams, which is why they're so common in development.

Class diagrams are different from activity diagrams and sequence diagrams in that they only display the application's sequence flow. The UML chart among the programmer community is the most popular.

The objective of the class diagram is summarized as follows:

- Static view analysis and design for an application.
- Describe the responsibilities of a system.
- Sets the stage for component and deployment diagrams.
- Forward and reverse engineering

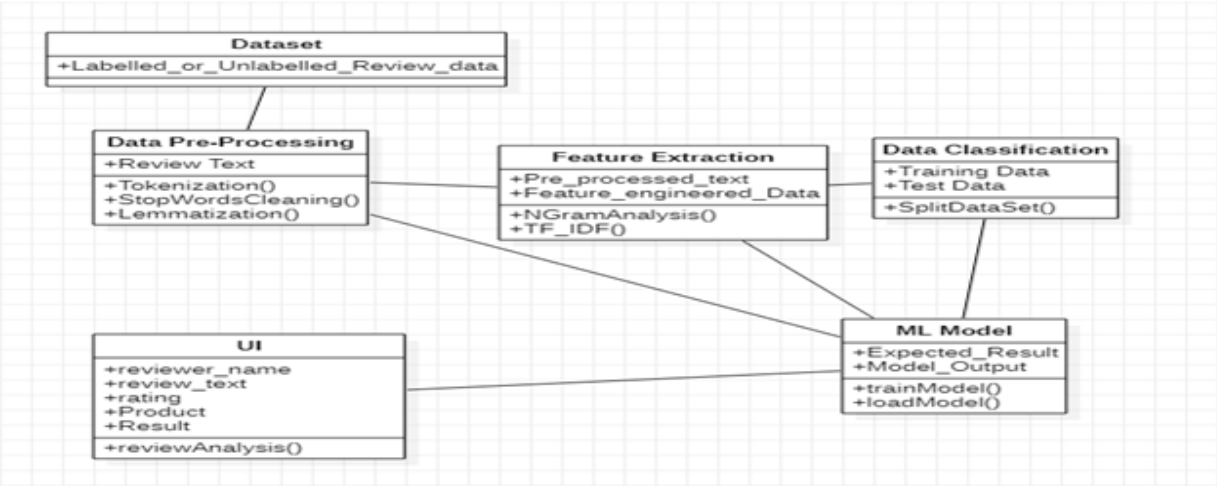

Fig. 3.1. Class diagram

Use case diagram:

A use case diagram is a diagram that depicts the dynamic behavior of a system. It incorporates use cases, actors, and their interactions to encapsulate the system's functionality. It denotes the activities, services, and functionalities that a system/subsystem of an application requires. It depicts the high-level functionality of a system as well as how the user interacts with it.

A use case is a collection of scenarios linked by a common aim. As a result, use case diagrams are created to display the system's functionalities.

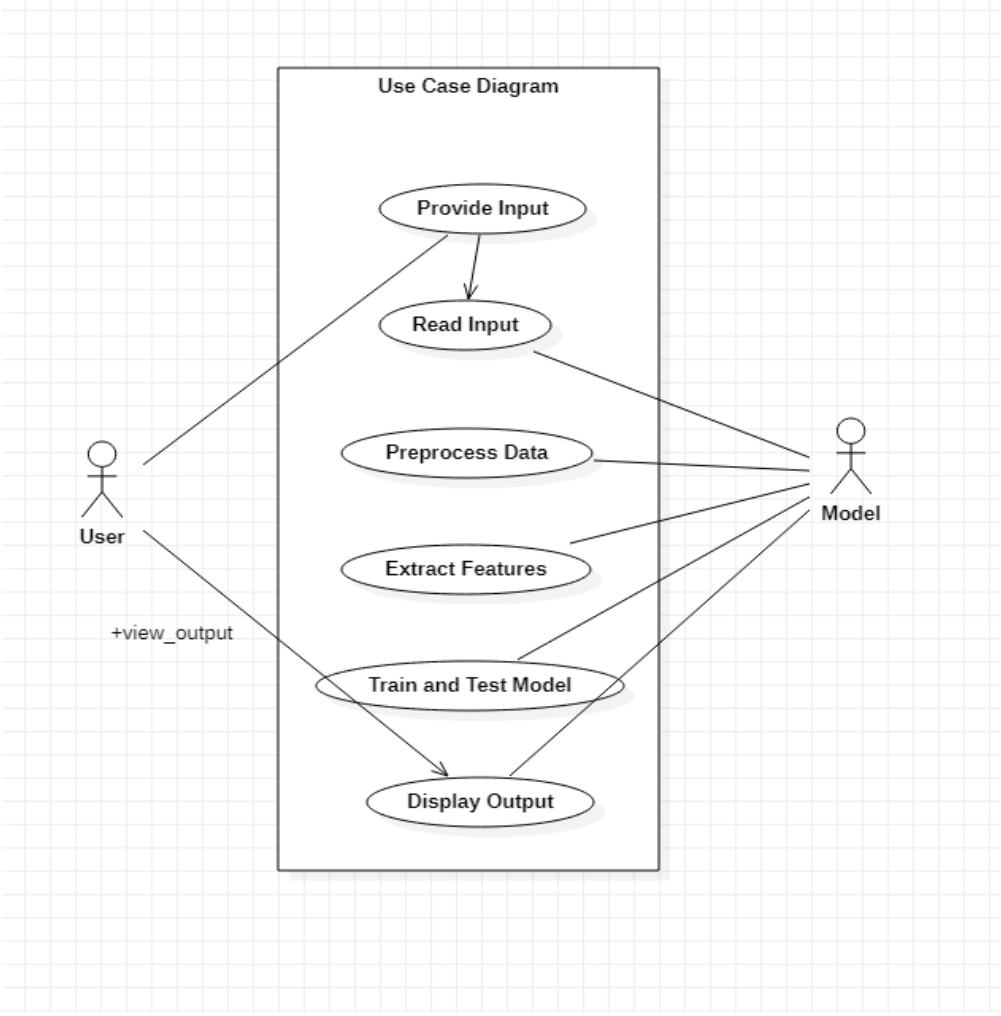

Fig 3.2-Use case diagram

Activity Diagram:

The activity diagram is used in UML to show the system's flow of control. It helps visualize how work flows from one activity to the next. It concentrated on the flow condition and the order in which it occurred. The flow can be sequential, branched or concurrent, and the activity diagram can handle all of them with forks, joins, and other capabilities. It's also known as an object-oriented flowchart. It refers to behaviors or procedures that are utilized to model the behavioral diagram.

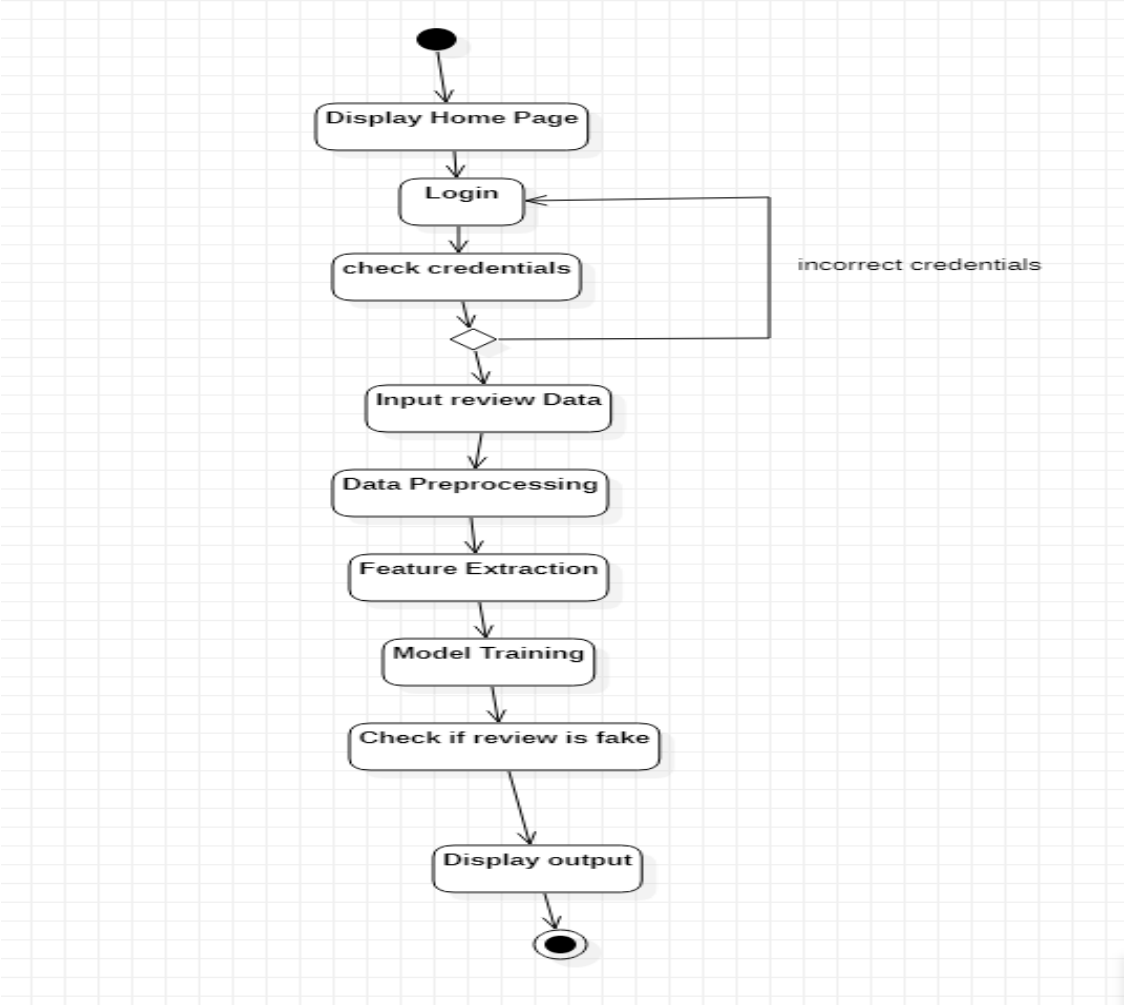

Fig. 3.3. Activity diagram

Sequence diagram:

A sequence diagram, also known as an event diagram or an event scenario, depicts how things interact in a sequential fashion or in the order in which they happen. A sequence diagram, is a diagram that depicts a series of events. Sequence diagrams depict how and in what order the various components of a system interact. These diagrams are frequently used by businesspeople and software engineers to explain and comprehend requirements for new and existing systems.

The following are the purposes of a sequence diagram:

- Create a model of the interactions between object instances in a collaboration that accomplishes the goal of a use case.
- Create a model of how items interact in a collaborative effort to execute a task.
- Model generic interactions (showing all possible paths through the interaction) or specialized interactions (showing specific examples of an interaction) (showing just one path through the interaction).

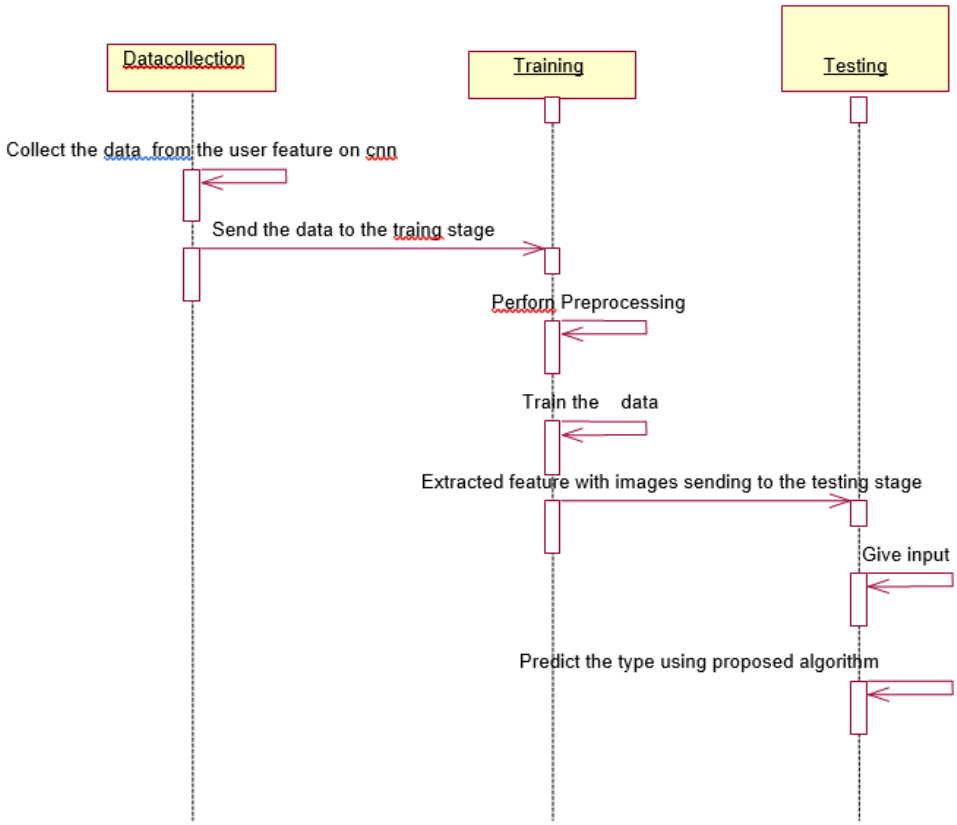

Fig. 3. 4. Sequence diagram

System architecture:

The mission and life cycle principles of the system are at the heart of system architecture, which is abstract, intellectual, and global. It also focuses on the high-level structure of systems and system elements. It explains the principles, concepts, traits, and qualities of the architectural system of interest. In some cases, it can be utilized on numerous systems, producing a common structure, pattern, and set of requirements for classes or families of similar or related systems.

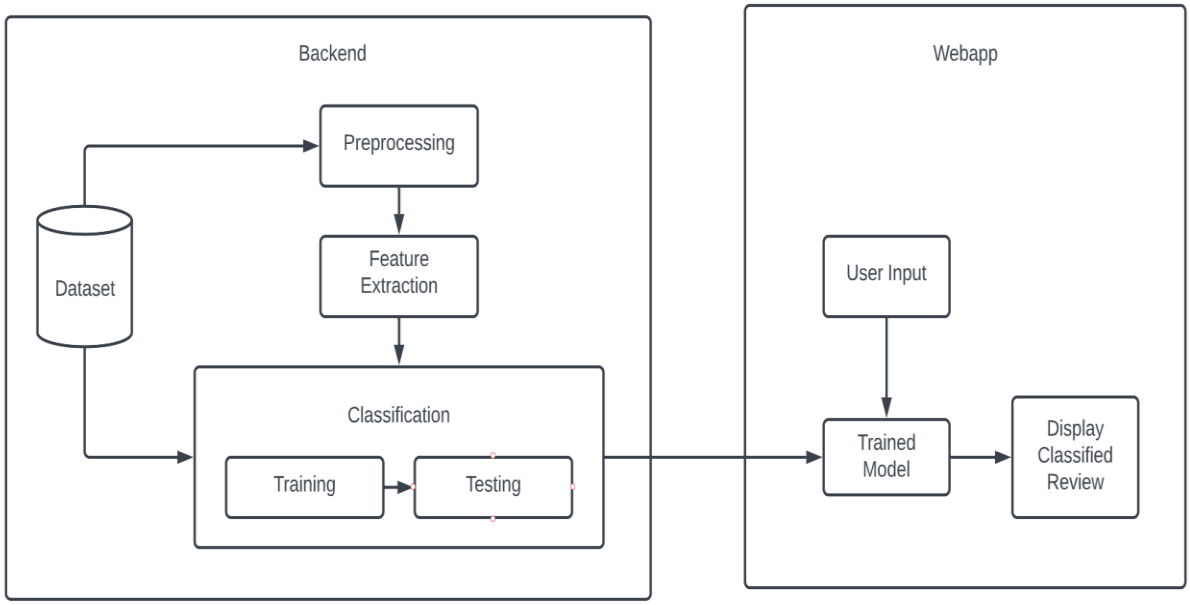

Fig. 3.5. System architecture diagram

Chapter-4

CONSTRUCTION

Implementation

We use the Python programming language to implement our project. To be particular, for the purpose of machine learning and to avoid a lot of installation problems, we use the google Colab Platform.

Colab is a free to use cloud-based Jupiter Notebook environment which allows anybody to write and execute python code through the browser, and is especially well suited for machine learning, data science and education. Colab supports many common machine learning libraries and can be quickly load then into your notebook via imports. It can also connect with google drive which makes file loading easier than ever.

The reasons to use Google Colab is:

- It enables us to ise Python on the internet.
- Distinguishing across diverse contexts.
- GPU-based High-Quality Compute Resources.
- Getting specified packages and libraries up and running.
- You may create, share, and upload notebooks.
- Notebooks that can be taken with you wherever you go i.e., Highly portable and accessible.
- It's simple to use and doesn't require any sort of configuration at any point.
- Python packages that are widely used are already preinstalled for the user. (For example, NumPy, pandas, OpenCV, and so on.)

We chose Colab for our project because it gives us free access to a powerful GPU, which we require to train our model.

Usually features such as Review Length and Textual Features are used but those are not the only features can be used to find out whether or not a review is legit or not. In our approach, we found that a few more additional features can be used. Some are:

- Review Rating
- Emoji Count
- Verified Purchase
- Readability Score, etc

By using these features along with several Supervised and Semi-Supervised Machine Learning algorithms, we select the best algorithm to make the model.

We start off by importing our dataset and using pandas to convert it into a Data Frame.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
color = sns.color_palette()
%matplotlib inline
```

+ Code + Text

```
data = pd.read_csv(path)
```

```
[5] data.head()
    # data.columns
    # print(data['REVIEW_TEXT'][0])
```

	DOC_ID	LABEL	RATING	VERIFIED_PURCHASE	PRODUCT_CATEGORY	PRODUCT_ID	PRODUCT_TITLE	REVIEW_TITLE	REVIEW_TEXT
0	1	__label1__	4	N	PC	B00008NG7N	Targus PAUK10U Ultra Mini USB Keypad, Black	useful	When least you think so, this product will sav...
1	2	__label1__	4	Y	Wireless	B00LH0Y3NM	Note 3 Battery : Stalion Strength Replacement ...	New era for batteries	Lithium batteries are something new introduced...
2	3	__label1__	3	N	Baby	B000I5UZ1Q	Fisher-Price Papasan Cradle Swing, Starlight	doesn't swing very well.	I purchased this swing for my baby. She is 6 m...
3	4	__label1__	4	N	Office Products	B003822IRA	Casio MS-80B Standard Function Desktop Calculator	Great computing!	I was looking for an inexpensive desk calcolat...
4	5	__label1__	4	N	Beauty	B00PWSAXAM	Shine Whitening - Zero Peroxide Teeth Whitenin...	Only use twice a week	I only use it twice a week and the results are...

```
[6] data.columns
```

```
Index(['DOC_ID', 'LABEL', 'RATING', 'VERIFIED_PURCHASE', 'PRODUCT_CATEGORY',
       'PRODUCT_ID', 'PRODUCT_TITLE', 'REVIEW_TITLE', 'REVIEW_TEXT'],
      dtype='object')
```

Fig. 4.1-Dataset and columns(features) present

In our dataset, we have 21000 Columns (10500 labelled as Fake & 10500 labelled as True). This data is then explored to find out hidden patterns and relationships. Eg: Relation between Review Label and the number of Emojis in the review.

Based on this, we form new attributes, select the attributes we need and they go into our training.

Then we apply preprocessing methods on the dataset such as Stop word Cleaning, Lemmatization, Tokenization and POS Tagging. We then apply n--grams to the text (Bigrams in this case) to split words into groups of 2 to perform further analysis.

This is how the textual data looks after preprocessing:

```
['i like', 'like laptop', 'laptop work', 'work well', 'i', 'like', 'laptop', 'work', 'well']
```

Fig. 4.2-Bigrams and Unigrams

Some of the features found through Data exploration are: Emoji Count, FV Score, Caps Count, and Stop words Count.

```
[ ]  cnt_srs = data.groupby(data["LABEL"]).RATING.value_counts()
     cnt_srs
     plt.figure(figsize=(16,8))
     sns.barplot(cnt_srs.index, cnt_srs.values, alpha=0.8, color=color[1])
     plt.ylabel('Number of Occurrences', fontsize=16)
     plt.xlabel('(Label, Rating)', fontsize=16)
     plt.title('Label Vs Rating', fontsize=18)
     plt.xticks(rotation='horizontal')
     plt.show()
```

Fig. 4.3.1-Code for Relation between Review Rating and Label

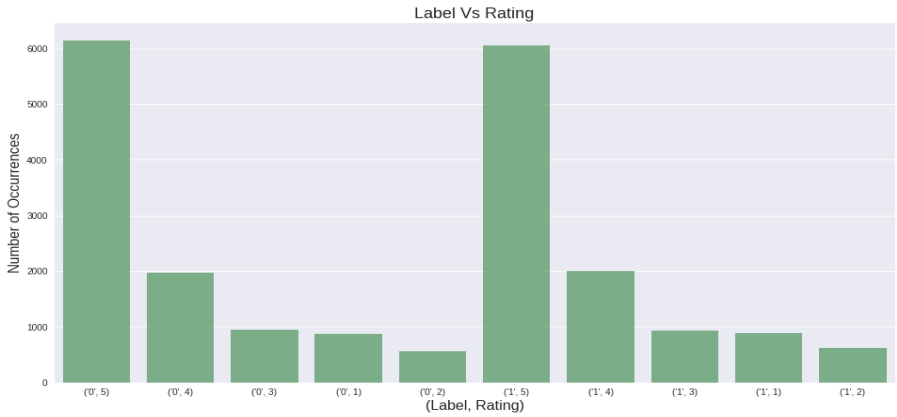

Fig. 4.3.2-Relation between Review Rating and Label

```
#readability score
from textstat.textstat import textstat
#fk score -> readability score

data["FK_Score"] = data["REVIEW_TEXT"].apply(textstat.flesch_kincaid_grade)
FKScoreValue = data.groupby(["LABEL"]).FK_Score.agg(lambda x: sum(x)/len(x))
FKScoreValue
```

```
LABEL
__label1__    5.697829
__label2__    5.558076
Name: FK_Score, dtype: float64
```

Fig. 4.4.1-Code for Relation between Review Rating and Label

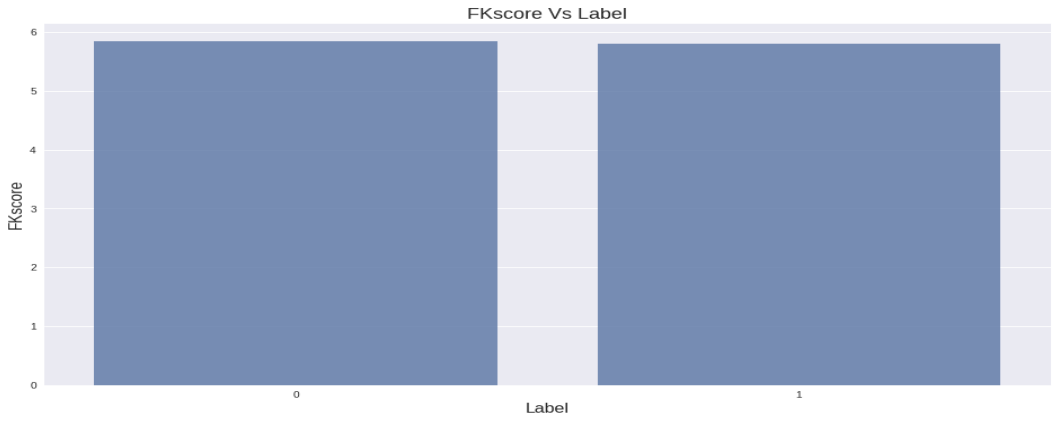

Fig. 4.4.2-Relation between FV Score and Label

```
def preProcess(text):
    # Should return a list of tokens
    lemmatizer = WordNetLemmatizer()
    filtered_tokens=[]
    lemmatized_tokens = []
    stop_words = set(stopwords.words('english'))
    text = text.translate(table)
    for w in text.split(" "):
        if w not in stop_words:
            lemmatized_tokens.append(lemmatizer.lemmatize(w.lower()))
        filtered_tokens = [' '.join(l) for l in nltk.bigrams(lemmatized_tokens)] + lemmatized_tokens
    return filtered_tokens

print(preProcess("I like this laptop because it works very well"))
```

Fig. 4.5-Removing Stop words and making Bigrams from Review Text

After that, we use TF-IDF to rank the words which appear the most frequently. We do this to find out the most used words in fake and real reviews.

We finally use multiple machine learning models and select the best one.

As of now, Passive Aggressive Classifier and SVM gave the best accuracy.

```
tfidf_vectorizer = TfidfVectorizer(stop_words='english', max_df=0.7)
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_vectorizer= TfidfVectorizer(min_df=1,stop_words='english')
```

Fig. 4.6-TFIDF Vectorizer

```
pac = PassiveAggressiveClassifier(max_iter=50)
pac.fit(tfidf_train,y_train)
```

```
PassiveAggressiveClassifier(max_iter=50)
```

```
In [1]: from sklearn.metrics import accuracy_score
        y_pred = pac.predict(X_test_counts)
        print(accuracy)

        0.7624413145539906
```

Fig. 4.7-Passive Aggressive Classifier

```
def trainClassifier(trainData):
    print("Training Classifier...")
    pipeline =  Pipeline([('svc', LinearSVC(C=0.01))])
    return SklearnClassifier(pipeline).train(trainData)
```

```
classifier = trainClassifier(trainData)
predictions = predictLabels(testData, classifier)
true_labels = list(map(lambda d: d[1], testData))
a = accuracy_score(true_labels, predictions)
p, r, f1, _ = precision_recall_fscore_support(true_labels, predictions, average='macro')
print("accuracy: ", a)
print("Precision: ", p)
print("Recall: ", a)
print("f1-score: ", f1)
```

```
Training Classifier...
accuracy:  0.806138933764
Precision:  0.813054281193
Recall:  0.806138933764
f1-score:  0.805062394205
```

Fig. 4.8-SVC Model (Linear SVM)

After making the model, we save it by dumping it into a pickle file. Pickle is a python module which allows you to pickle a file using serialization and de-serialization. This means simply breaking down an object into its constituting components. Any python object can be serialized by breaking its components into a byte stream (stream of 0's and 1's). This process is also called "Flattening". If we want to use that specific object again, pickle turns the serialized / pickled objects back to python object by a process called "De-Serialization".

After loading out finished model, we use Flask, which is a web application framework written in Python; to make our Webapp. Flask enables users to develop web applications easily in Python. It is very easy to

get started with as it is written in pure Python and it doesn't have a huge learning curve. Its simplistic coding style and ease of deployment made it the perfect choice to use in our project.

We made a Home Page, a Login Page and a Fake Review Detection Page using HTML, CSS and JavaScript. We integrated our model and the GUI using Pickle. Finally, our website runs on the user's localhost (port 5000) using Flask.

Fig. 4.9-Flask

This is how our Webapp Looks like:

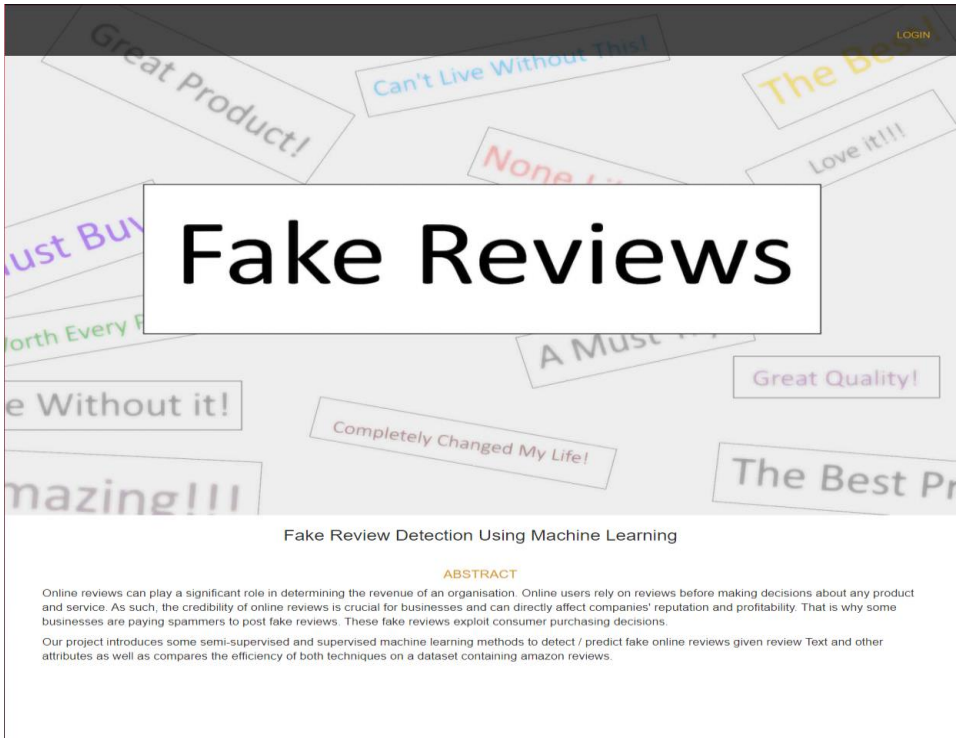

Fig. 4.9-Home Page

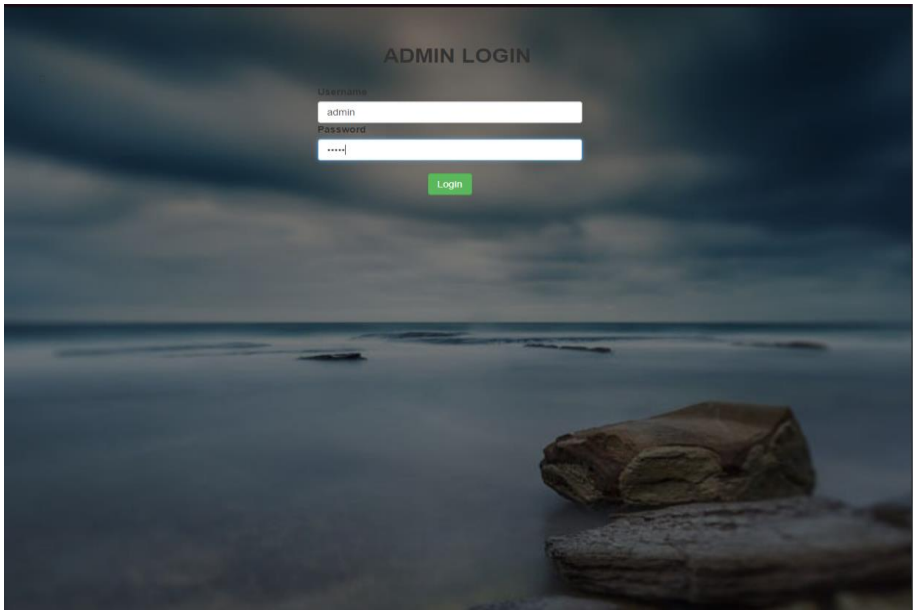

Fig. 4.10-Login Page

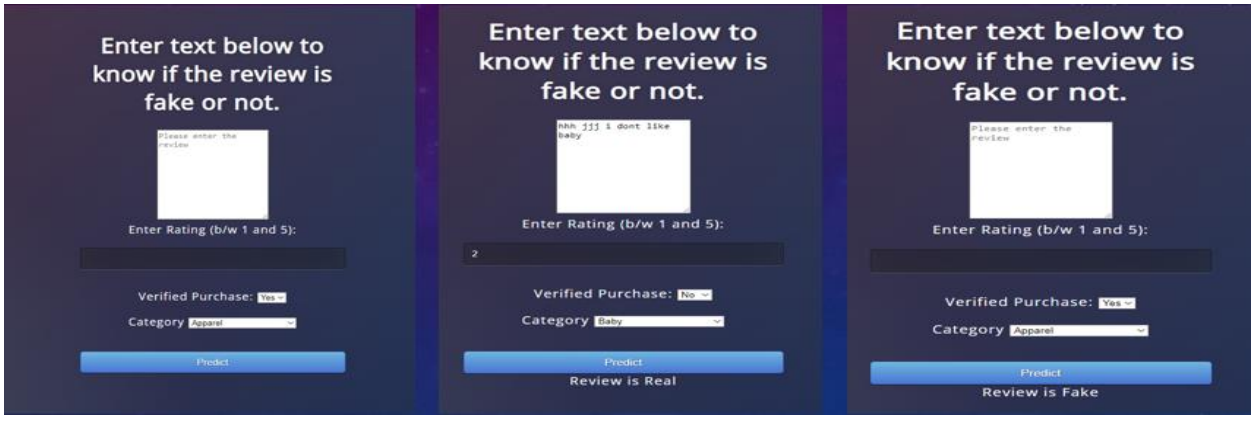

Fig. 4.11-Fake Review Detection

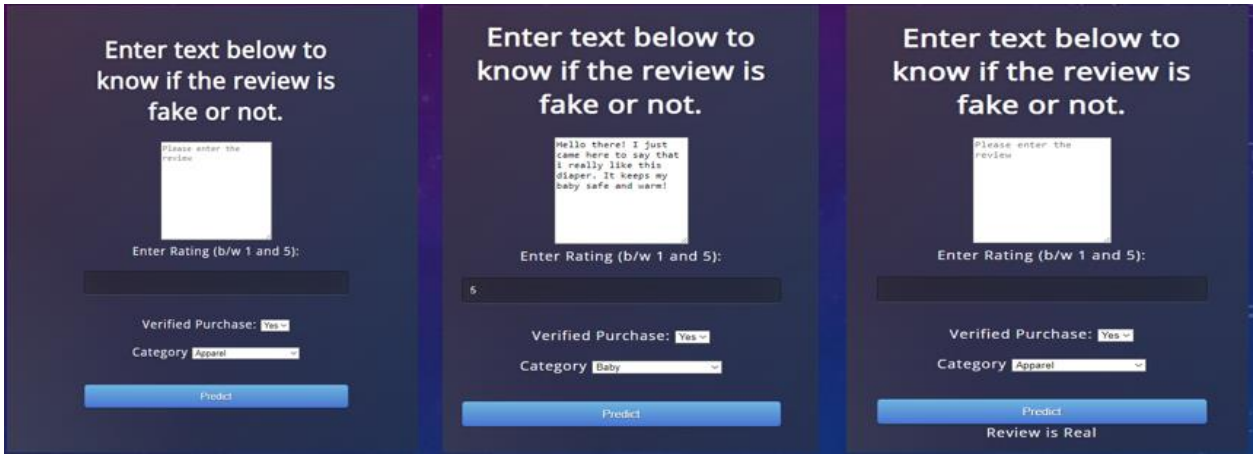

Fig. 4.12-Real Review Detection

Software Details

- Google Colab
- Jupyter Notebook
- Python (3.10.2)
- Python Libraries – Pandas, sklearn, Flask, pickle, NumPy, textstat and nltk (natural language toolkit).

Hardware Details

- Operating system: Windows 7 or newer, 64-bit macOS 10.9+, or Linux.
- System architecture: 64-bit x86, 32-bit x86 with Windows or Linux.
- CPU: Intel Core 2 Quad CPU Q6600 @ 2.40GHz or greater.
- RAM: 4 GB or greater.

Testing

Testing is the process of evaluating a system and its components to determine if they meet defined criteria. It is also defined as the process of running your system to find vulnerabilities, bugs, or additional requirements and patchwork. It is a consultation designed to provide information to the interested entities on the quality of the product or service. The program tests provide an objective and unbiased overview of the software to understand the risks associated with implementing it. This process of running a program or application to find errors and ensure that it is suitable for use is called testing.

The test is the process of assessing the quality of one or more of interest through the execution of a software component, service or system. In general, these qualities show how well the component under the test is. When it meets the requirements of the design and development guide, is suitable for a wide range of inputs, performs its function in a reasonable amount of time, runs in the intended environment where it can be used and installed, and achieves the overall result desired by produce; then it is said to be well developed and tested.

White box testing

White box testing is a software testing approach which involves testing the structure, underlying code and architecture of a product to validate its input-output flows and to improve its design, capabilities, and performance. It testing is also referred to as “Transparent-box testing” or “Open-box testing”. The 3 general steps taken while performing white box testing are:

- Test the transparent box
- Test the code base
- Test the glass box so that the code is visible on the tester.

It is one of two techniques used in software testing known as “Box testing” where the other is called “Blackbox Testing”; which focuses more on internal application behavior and internal software engineering testing.

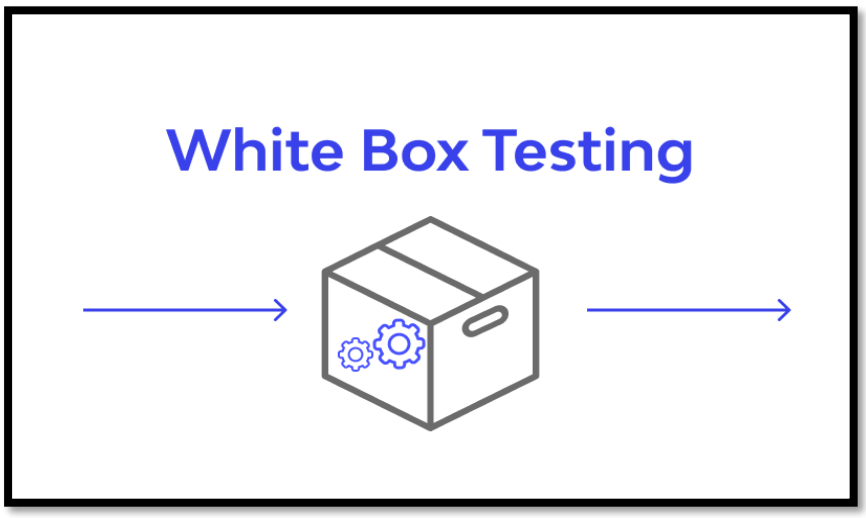

Fig. 4.13-White Box Testing

Black box testing

Black box testing is a software testing methodology that deals with testing the functionality of a software application without any knowledge of the internal code structure, internal origins or details of implementation. It is a type of software test that focuses on and validates the inputs and outputs of a software application rather than concentrating on the internal details. Black box testing is also called "Crash testing".

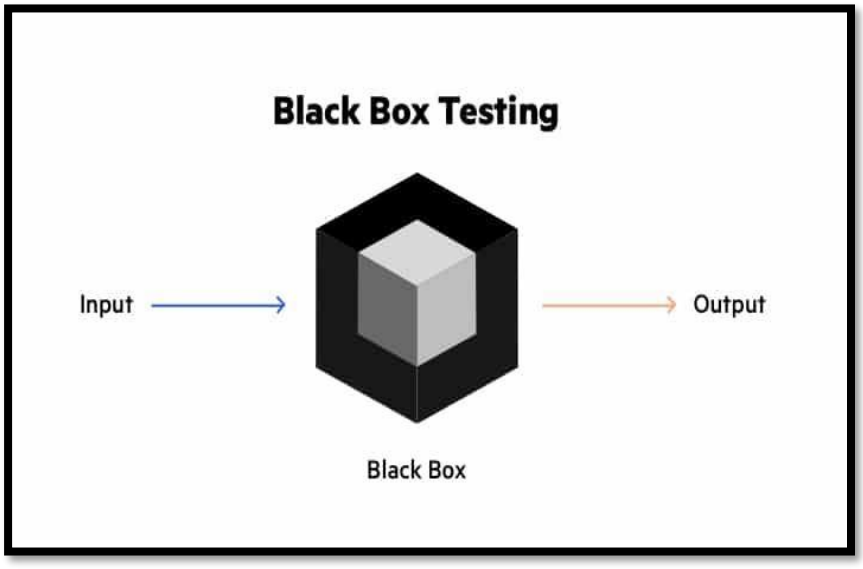

Fig. 4.14-Black Box Testing

Chapter-5

CONCLUSION AND FUTURE SCOPE

Conclusion

We demonstrated the importance of reviews and how they affect practically everything linked to web-based data in this project. Reviews, without a doubt, play an important role in people's decisions and as a result, detecting false reviews is an active and ongoing research topic. A machine learning strategy to detecting fake and illegitimate reviews is discussed here. The features of the reviews and reviewers, review content and additional metadata are taken into account in the suggested method. The proposed method is evaluated using the Amazon dataset and various classifiers are used. In the developed technique, the SVC and Passive Aggressive models are applied and contrasted.

Future Scope

In future work, we may consider including additional behavioral features of the reviewer such as features that depend on the, the time reviewers take to complete reviews, frequent times the reviewers write the reviews and how often they are submitting positive or negative reviews (time interval). This will be possible if a dataset containing relevant information is made available to the public as most datasets containing this information are owned by private parties. It is expected that considering more behavioral features (like the ones mentioned above) will enhance and improve the performance of the presented detection approach as the legitimacy of a review not only depends on its textual content but also the patterns and behavior of the person posting the review.

References

- Anusha Sinha, Nishant Arora, Shipra Singh, Mohita Cheema, Akthar Nazir, “Fake Product Review Monitoring Using Opinion Mining”, International Journal of Pure and Applied Mathematics (IJPAM), Volume 119 No. 12 2018.

- Ahmed M. Elmogy , Usman Tariq , Atef Ibrahim, Ammar Mohammed, “Fake Reviews Detection using Supervised Machine Learning”, International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 12, No. 1, 2021

- Swathi Raje, Gouri Patil, “A Machine Learning Model to Predict Fake Review using

Classifier on Yelp Dataset", International Journal of Engineering Research & Technology (IJERT), Vol. 10 Issue 08, August-2021.

► Hema Dewangan, Prof. Om Prakash Dewangan, "Opinion Spam Detection, Tools and Techniques : A Review ", International Journal of Computational Intelligence Research (IJCIR) ISSN 0973-1873 Volume 13, Number 7 (2017).

► M. Ott, C. Cardie, and J. Hancock, "Estimating the prevalence of deception in online review communities," in Proceedings of the 21st International Conference on World Wide Web, pp. 201–210, ACM, 2012.

► Chengai Sun, Qiaolin Du and Gang Tian, "Exploiting Product Related Review Features for Fake Review Detection," Mathematical Problems in Engineering, 2016.

► A. Heydari, M. A. Tavakoli, N. Salim, and Z. Heydari, "Detection of review spam: a survey", Expert Systems with Applications, vol. 42, no. 7, pp. 3634–3642, 2015.

► M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT), vol. 1, pp. 309–319, Association for Computational Linguistics, Portland, Ore, USA, June 2011.

► J. W. Pennebaker, M. E. Francis, and R. J. Booth, "Linguistic Inquiry and Word Count: Liwc," vol. 71, 2001.

► S. Feng, R. Banerjee, and Y. Choi, "Syntactic stylometry for deception detection," in Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers, Vol. 2, 2012.

► J. Li, M. Ott, C. Cardie, and E. Hovy, "Towards a general rule for identifying deceptive opinion spam," in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL), 2014.

► E. P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, "Detecting product review spammers using rating behaviors," in Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM), 2010.

► J. K. Rout, A. Dalmia, and K.-K. R. Choo, "Revisiting semi-supervised learning for online deceptive review detection," IEEE Access, Vol. 5, pp. 1319–1327, 2017.

CSE-B15

ORIGINALITY REPORT

23%	10%	3%	19%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	innovate.mygov.in Internet Source	3%
2	Submitted to Northcentral Student Paper	2%
3	Submitted to University of Hertfordshire Student Paper	2%
4	Submitted to University of Pretoria Student Paper	1%
5	Submitted to University of Greenwich Student Paper	1%
6	Submitted to The British College Student Paper	1%
7	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	1%
8	Submitted to Victorian Institute of Technology Student Paper	1%
9	Submitted to National Institute of Technology Karnataka Surathkal	1%

Student Paper

10	Submitted to University of Wolverhampton Student Paper	1 %
11	Submitted to Birla Institute of Technology and Science Pilani Student Paper	1 %
12	Submitted to West Windsor Plainsboro High School South Student Paper	1 %
13	T R Mahesh, V Vivek, Vinoth V Kumar, Rajesh Natarajan, S. Sathya, S. Kanimozhi. "A Comparative Performance Analysis of Machine Learning Approaches for the Early Prediction of Diabetes Disease", 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), 2022 Publication	1 %
14	Submitted to The Robert Gordon University Student Paper	1 %
15	Submitted to Universiti Teknologi Malaysia Student Paper	1 %
16	Submitted to Wakefield College Student Paper	1 %
17	www.ijert.org Internet Source	1 %

18	Deepak Pareta, Indukuri Nishat Verma, Bhanu Prakash Lohani, Pradeep Kumar Kushwaha, Vimal Bibhu. "IoT Enabled Smart and Efficient Musical Water Fountain", 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), 2022 Publication	1 %
19	Submitted to CSU, Los Angeles Student Paper	<1 %
20	Amartya Chakraborty, Sunanda Bose. "Around the world in 60 days: an exploratory study of impact of COVID-19 on online global news sentiment", Journal of Computational Social Science, 2020 Publication	<1 %
21	Submitted to BITS, Pilani-Dubai Student Paper	<1 %
22	Submitted to University of Sunderland Student Paper	<1 %
23	Submitted to King's Own Institute Student Paper	<1 %
24	Submitted to NCC Education Student Paper	<1 %
25	Submitted to Glasgow Caledonian University Student Paper	<1 %

26	scholarsmepub.com Internet Source	<1%
27	www.guru99.com Internet Source	<1%
28	www.sebokwiki.org Internet Source	<1%
29	devqa.io Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	Off		