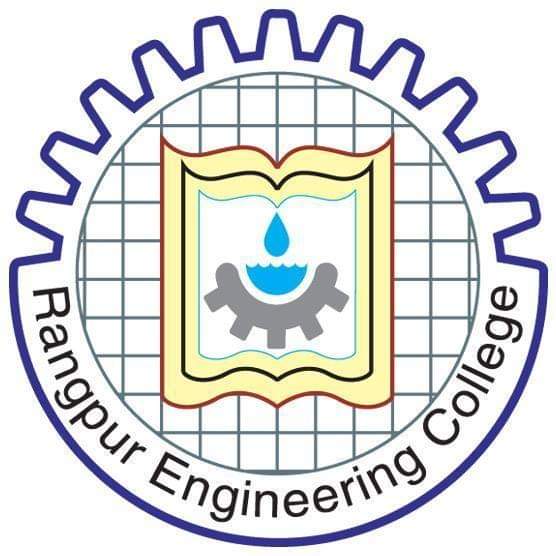
**Project Title:** “**FAKE NEWS DETECTION SOFTWARE USING MACHINE LEARNING”**



**Department of Information and Communication Engineering**

**Rangpur Engineering College, Affiliated by University of Rajshahi**

**Rangpur-5400, Rangpur**

The Report Submitted in Department of Information & Communiction Engineering in partial Fulfilment of requirements for the Degree of Bachelor of Science in Engineering

**Submitted By**

**ANNIKA ISLAM RUMKY**

**Roll: 2037621133**

**Session: 2019-2020**

**Registration No: 2037621133**

**Department of Information & Communication Engineering**

**Supervisor**

Md. Nazmul Hasan

Lecturer of of Information and Communication Engineering

Rangpur Engineering College

Department of Information and Communication Engineering

Rangpur Engineering College, Affiliated by University of Rajshahi

Rangpur-5400, Rangpur

SEPTEMBER, 2025

**Acknowledgment**

First of all, I’m thankful and expressing my gratefulness to Almighty Allah who offers us his divine blessing, patient and physical strength to complete this project work.

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success.

I am deeply indebted to our project supervisor **Md. Nazmul Hasan,** Lecturer of the dept of Information & Communication Engineering, Rangpur Engineering Collage, Rangpur. His scholarly guidance, important suggestions, work for going through our our drafts and correcting them, and generating courage from the beginning to the end of the Project work has made the completion of this project possible.

I do gratefully acknowledge and express my gratitude to all the group members of “Information and Communication engineering” and friends who supported us in preparing this project.

With best regrads,

**Annika Islam Rumky**

**Roll: 2037621133**

Abstract

False information or, as it's occasionally called, fake news, has come a global issue of immense social, political, and profitable consequences in the ultramodern, largely connected information technology( IT) period. False information can go viral on social media networks and online news platforms, frequently reaching out to millions of druggies long before a vindicated information can be released. It's hard to find similar content and it's hard to do it effectively. The pace and volume of digital information make it insolvable to corroborate the data manually. To master this challenge, this thesis proposes a machine literacy grounded fake news sensor system utilising the Passive- Aggressive Classifier.

The design uses the Kaggle Fake and Real News Dataset, which is freely available as well and contains over 6,000 news particulars that are moreover considered as" FAKE" or" REAL." All the papers have the textual features similar as body, marker, caption, and textbook. The dataset preprocessing( including tokenization, stop- word junking, and TF- IDF vectorization) styles are grounded on natural language processing( NLP) and convert raw textbook to a numerical representation that can be used in machine literacy algorithms.

Passive- Aggressive Classifier is chosen because it's effective to classify high- dimensional data and it's applicable to online literacy terrain. It's fast and memory-effective unlike traditional batch literacy models, which requires streamlining its parameters only when a misclassification takes place. crucial performance pointers( delicacy, perfection, recall, and F1- score) are used to train and test the model. It has been shown in the experimental results that the classifier is competitive in its capability to separate between genuine and fabricated news papers and the high values of perfection and recall are substantiation of the trustability of the classifier.

In order to increase operation and practical operation, the trained model is enforced in the form of a web grounded aspirant to enable druggies to feed news textbook and get immediate bracket labors. The interface is also stoner friendly and intuitive, and allows both specialized andnon-technical druggies to use the system without any difficulty.

This thesis fits into the larger design of fighting misinformation by furnishing an automated, scalable way of promoting media knowledge and responsible information consumption. It also questions the innocently correctness of automated fake news discovery and the need to be open, fair, and responsible in the digital period when AI is stationed.

**Dedication**

**“**In dedication to my Family**”**

**Table of content**

**TITLE PAGE**

**1. Certification …………………………………………………………... (i)**

**2. Acknowledgement …………………………………………………….. (ii)**

**3. Abstract ……………………………………………………………….. (iii)**

**4. Dedication …………………………………………………………….. (iv)**

**5. Chapter 1 : Introduction …………………………………………. 01-07**

**1.0 Background of study………………………..……. 01**

**1.1 Motivation ……………………………………..……. 02**

**1.2 Objectives of study ....………………………….…. 03**

**1.3 Problem statement ….…………………………..…. 03**

**1.4 Scope of the study ..……………………………….. 04**

**1.5 Significance of the study ………………………….05**

**1.6 Research questions …………………………………06**

**1.7 Organization of the thesis …………………………06**

**6.Chapter 2 : Literature Review …………………………………… 05-08**

**2.0 Understanding Fake News …...……………..…….. 05**

**2.1 Manual vs. Automated Detection .………….…….. 05**

**2.2 Machine Learning in Fake News Detection …….. 06**

**2.3 Passive-Aggressive Classifier ……………….…. 06**

**2.4 Related Work …………………………………..…….07**

**2.5 Ethical Considerations …………………………….08**

**2.6 Summary ……………………………………………..09**

**7. Chapter 3 : Methodology ……………………………………….…. 07-25**

**3.0 Introduction …………………………………………….. 07**

**3.1 Dataset Description ………………………………..….. 12**

**3.2 Data Preprocessing …..……………………………….. 13**

**3.3 Feature Extraction Using TF-IDF …….……………… 13**

**3.4 Model Selection ………………………….…………….. 14**

**3.5 Model Evaluation ..……………………………………... 19**

**3.6 Model Saving ………………………………….………… 21**

**3.7 Web Application Development ……………………….. 22**

**3.8 Summary …………………………………………………26**

**3.9 Working Environment ……………………………………**

**8. Chapter 4 : DEPLOYMENT ..….…………………………..……..…. 26-34**

**4.0 Introduction ……………….……………………….…… 26**

**4.1 Deployment Process …………………….…………… 28**

**4.2 Conclusion …………………………..…………………. 33**

**9. Chapter 5 : RESULT AND DISCUSSION ..…………………..…. 35-37**

**5.0 Introduction ……………………………………..…. 35**

**5.1 Evaluation Metrics …………………………..………………. 35**

5.2 Model Performance …………………..………….…………. 36

5.3 Confusion Matrix Analysis…………….……………………… 36

5.4 Evaluation Metrics ……………………………………………. 37

5.5 Precision, Recall, and F1 Score…….……………………… 37

5.6 Web Application Output Samples…………………………..38

5.7 Discussion of Results ………………………………………..39

**10. References ……..…………………………………………..………..…. 38-39**

**CHAPTER ONE**

**1.0 BACKGROUND OF THE STUDY**

In digital communication, the internet has brought about a revolution of information product, dispersion and consumption. The news is no longer confined to the conventional sources of news, but it's currently being circulated freely on social media, blogs, and messaging operations. Although this democratization of information has enabled individualities and communities to be empowered, it has also led to the emergence of a nocuous miracle, i.e., the massive spread of fake news. Fake news can be defined as a deliberate creation of news that resembles licit news content without having the data and having an instructional intent. It's generally cooked so as to deceive, instigate or control the millions. The counteraccusations of fake news are dramatic-- they can affect choices, beget violence, misrepresent the story of public health, and undermine institutional credibility. False information created by fake news in Bangladesh has caused fear in the case of natural disasters, circulated misreading regarding vaccination, and led to political insecurity.

It is not only that fake news requirements to be linked, but it's necessary to do it in an effective and scaly manner. Fact- checking done manually is accurate, but is time- consuming and resource- consuming. Accordingly, scientists and technologists are resorting to machine literacy( ML) and natural language processing( NLP) to produce automated systems that can descry fake news in real time.

**1.1 MOTIVATION**

This thesis is driven by the fact that we're in dire need of data- driven results to misinformation on a scale. I'm a pupil of Information and Communication Engineering( ICE) and I'm veritably keen on the question of technology crossroad with social impact. The given design will enable me to use my logical chops in relation to a real- life issue, as well as to probe the practical perpetration of machine literacy.

The Passive- Aggressive Classifier is named because it's effective in managing huge volume of textbook bracket. It's especially suited to online literacy conditions, where information comes in successionally and information has to be streamlined snappily. By incorporating this classifier into a web grounded interface, the design will offer end druggies a accessible tool to pierce in the hunt to authenticate the content of news stories.

**1.2 OBJECTIVES OF STUDYS**

The main purposes of such a thesis are

* To explore the social effect of fake news and the downsides of homemade discovery.
* To tokenize, stop- word, TF- IDF textbook data and preprocess and vectorize textual data with NLP styles.
* To emplace the Passive- Aggressive Classifier to perform the double bracket of news papers.
* To measure the performance of the model grounded on the criteria of delicacy, perfection, recall and F1- score.
* To come up with a web interface that's responsive to fake news discovery in real- time.
* To probe the ethical issues of automated misinformation discovery.

**1.3 PROBLEM STATEMENT**

The proliferation of fake news is a grave challenge to informed decision timber and responsibility of the crowd. The old fact- checking ways are n't scalable, and not all druggies have the capability or time to fact- check themselves. There's an critical necessity to find automated systems, which will be able of relating fake news effectively and efficiently.

The thesis will help break the problem by creating a machine literacy model that will classify news papers as either real or fake by assaying their features of the textbook. Passive- Aggressive Classifier is also chosen due to its capacity of recycling high dimensional data and is applicable in the environment of online literacy. The system should offer a scale result that can be applied onto digital platforms to help the druggies corroborate news content.

**1.4 SCOPE OF THE STUDY**

**This exploration is grounded on the double bracket of news papers grounded on textual information into two orders" REAL" and" FAKE" news papers. The data is in English and was attained at Kaggle. Image grounded or videotape grounded misinformation is n't covered in the design. Although the system supports scalability, presently, it's only enforced with the use of a web interface to enter the textbook.**

**The scope includes**

* **Preprocessing of data and point selection.**
* **Training and evaluation of the models.**
* **Web- grounded deployment**
* **Automated discovery is subject** **to ethics.**

**1.5 SIGNIFICANCE OF THE STUDY**

**This paper has significance in colorful fields :**

**Scholarly It adds to the current literature regarding machine literacy in textbook bracket and misinformation discovery.**

**Social It spreads the knowledge of fake news and teaches about responsible consumption of the media.**

**Specialized It illustrates how to combine NLP, machine literacy and web technologies in a practical result.**

**Ethical It brings up significant enterprises regarding bias and suppression and the influence of algorithms on the conformation of the discussion.**

**The system created during this design can be gauged on other types of content, platforms, and languages, as well as the struggle with the misinformation issue is a different tool.**

**1.6 RESEARCH QUESTIONS**

**The following exploration questions are -**

1. **Which is the Discovery delicacy of the Passive- Aggressive Classifier in relation to the rest of the models, to describe fake news?**
2. **Which preprocessing styles can be used to ameliorate the performance of the classifier?**
3. **Is it possible to make the system a part of a web grounded real- time discovery system?**
4. **Which are the sins of machine literacy when it comes to identification of subtle or environment specific misinformation?**
5. **What are ways of spanning the system or conforming it to multilingual and multimedia content?**
6. **What are the challenges in deploying fake news detection systems in real-time environments such as social media platforms?**

**1.7 ORGANIZATION OF THE THESIS**

**This thesis is divided into five chapters :**

**In Chapter 1, the problem, motivation, significance of the study are introduced.**

**Chapter 2 presents the current exploration on fake news discovery and machine literacy styles.**

**Chapter 3 explains the methodology, similar as dataset, preprocessing, enforcing models, and evaluations.**

**Chapter 4 gives the results and explains the performance of the model.**

**The last chapter of the study summarizes and suggests possibilities of the future.**

**CHAPTER TWO**

**2.0 INTRODUCTION**

Spreading fake news has come an transnational issue, particularly in the digital age, where the information propagates presto through social media and the Internet. The chapter is a literature review of the living workshop on fake news discovery, misinformation development, and uses of machine literacy styles, specifically the Passive- Aggressive Classifier in working the problem. It also discusses ethical issues and gaps in the being exploration.

**2.1** **UNDERSTANDING FAKE NEWS DETECTION**

**False or deceiving information that has been made to appear licit and is not real news is what's called fake news. Allcott and Gentzkow( 2017) note that fake news came a hot content in the 2016 presidential election in the United States, where social media was used to circulate fake news. Misinformation in Bangladesh has caused fear among the millions particularly during health tragedies and political situations. Fake news may be divided into –**

* **Clickbait : Sensational headlines to attract attention.**
* **Propaganda : Politically motivated falsehoods.**
* **Satire : Humorous content that may be misinterpreted.**
* **Hoaxes : Deliberate deception for personal or financial gain.**

**2.2 MANUAL vs. AUTOMATED DETECTION**

**Conventional fact checking involves the use of mortal specialists to check claims. Though precise, this approach is slow and is unfit to match the content creation on a diurnal base. Scalable volition Automated systems can be used to classify news papers according to verbal patterns, metadata, and source credibility using algorithms.**

|  |  |  |
| --- | --- | --- |
| **Method** | **Pros** | **Cons** |
| **Manual** | **High accuracy** | **Time consuming** |
| **Automated (ML)** | **Scalable,fast** | **Relies upon data quality** |

**2.3 MACHINE LEARNING IN FAKE NEWS DETECTION**

Machine learning has also come one of the effective tools in now-a-days. Naive bayes, Support Vector machine( SVM), Logistic Regression are exemplifications of algorithms that have been considerably applied to detect fake news. These models are trained on labeled datasets, and they can find patterns that distinguish between real and fake content.Crucial ways include –

* Supervised Learning Does need labeled data( e.g., FAKE vs. REAL).
* Natural Language Processing( NLP) Transforms textbook to numerical features.
* TF- IDF Vectorization Comparisons between the significance of words in the documents.

**2.3 PASSIVE-AGGRESSIVE CLASSIFIER**

 Passive Aggressive Classifier is an online learning algorithm where you train a system incrementally by feeding it instances sequentially, individually or in small groups called mini-batches.

In online learning, a machine learning model is trained and deployed in production in a way that continues to learn as new data sets arrive. So we can say that an algorithm like Passive Aggressive Classifier is best for systems that receive data in a continuous stream. Passive Aggressive Classifier belongs to the category of online learning algorithms in machine learning. It works by responding as passive for correct classifications and responding as aggressive for any miscalculation.

Advantages :

* Applicable to high- dimensional data.
* Time-effective.
* Competitiveness in delicacy over the traditional models.

Limitations :

* Sensitive to noisy data.
* Hyperparameters have to be tuned with care.

According to the exploration conducted by Aggarwal( 2018) and Kaliyar et al.( 2021), Passive-Aggressive models are effective in spam discovery, sentiment analysis, and fake news groups with the addition of applicable preprocessing styles.

**2.4 RELATED WORK**

A number of experimenters have delved fake news discovery by using different styles

* Rubin et al.( 2016) applied the proposition of rhetorical structure to identify the deceitful content.
* Shu et al.( 2017) suggested a model that integrates content, social environment and geste of the stoner.
* Deep literacy models like CNN and LSTM that were used by Ahmed et al.( 2018) to descry fake news.
* Kaliyar et al.( 2021) compared the classic ML algorithms and discovered that Passive- Aggressive was effective in double bracket.

2.5 ETHICAL CONSIDERATION

Automated fake news detection is immorally questionable.Bias Training data can be poisoned in the models. Censorship is also important.Also the user need to know the manner in which opinions are reached.

The deployment of AI should be done responsibly grounded on fairness, explainability, and constant monitoring. The inventors should make sure that models do not support negative conceptions or silencing of marginalized voices.

2.6 SUMMARY

The literature shows that the use of machine learning in detecting fake news has been of adding interest. Although of the models have been promising, Passive- Aggressive Classifier is effective and flexible. The thesis is grounded on former studies that enforced this algorithm using the real- world data and applied it to a web- grounded interface to make it accessible to the general population.

**CHAPTER THREE**

**3.0 INTRODUCTION**

This chapter introduces the entire approach that was followed to create an artificial news detection system based on machine learning. The method incorporates data acquisition, preprocessing, feature engineering, model selection, training, evaluation and deployment. The idea is to develop a scalable, efficient and user-friendly system that can grade news stories as either FAKE or REAL depending on their textual information. The approach is based on the practical implementation in Python and deployed through the web interface created with the help of HTML, CSS and Flask.

**3.1 DATASET DESCRIPTION**

**The dataset in this project is the Fake and Real News Dataset, which was obtained at Kaggle. It is a popular benchmark dataset on binary text classification tasks, especially on the research of fake news detection. The dataset was filtered by integrating two distinct files; fake.csv and true.csv which held labeled news articles. In this project, the files were combined and to save them as news.csv.**

**3.1.1 SOURCE AND ORIGIN**

**Website:** [**news.csv**](news.csv) **(from Kaggle)**

**Curator: Clement Bisaillon**

**License: Academic open-source.**

**Purpose: The aim of the model is to help machine learning models to differentiate fake and real news articles using textual content.**

**The dataset has been utilized in various academic research studies and competitions and therefore it is a validated and reliable data to experiment.**

**3.1.2 DATASET STRUCTURE**

**The dataset contains 6335 rows and 4 columns:**

|  |  |  |
| --- | --- | --- |
| Column name | Data type | Description |
| Unnamed | integer | Index column(not use in modeling) |
| Title | string | Headline of News |
| Text | string | Content |
| Label | string | Classification label (FAKE or REAL) |

The rows designate separate news articles. The target variable of supervised learning is the label column.

3.1.3 CLASS DISTRIBUTION

The data sample is balanced, and this is good in binary classification:

FAKE : ~ 3,171 articles

REAL : ~ 3,164 articles

This almost equal distribution will avoid the biasness of a certain class and provide equitable consideration of the models.

3.1.4 SAMPLE DATA

|  |  |  |  |
| --- | --- | --- | --- |
| UNNAMED | TITLE | TEXT | LABEL |
| 8476 | You Can Smell Hillaryâ€™s Fear | Daniel Greenfield, a Shillman Journalism Fellow at the Freedom Center, is a New York writer focusing on radical Islam.  In the final stretch of the election, Hillary Rodham Clinton has gone to war with the FBI.  The word â€œunprecedentedâ€ has been thrown around so often this election that it ought to be retired. But itâ€™s still unprecedented for the nominee of a major political party to go war with the FBI.  But thatâ€™s exactly what Hillary and her people have done. Coma patients just waking up now and watching an hour of CNN from their hospital beds would assume that FBI Director James Comey is Hillaryâ€™s opponent in this election.  The FBI is under attack by everyone from Obama to CNN. Hillaryâ€™s people have circulated a letter attacking Comey. There are curr…………. | FAKE |
| 3608 | Kerry to go to Paris in gesture of sympathy | U.S. Secretary of State John F. Kerry said Monday that he will stop in Paris later this week, amid criticism that no top American officials attended Sundayâ€™s unity march against terrorism.  Kerry said he expects to arrive in Paris Thursday evening, as he heads home after a week abroad. He said he will fly to France at the conclusion of a series of meetings scheduled for Thursday in Sofia, Bulgaria. He plans to meet the next day with Foreign Minister Laurent Fabius and President Francois Hollande, then return to Washington.  The visit by Kerry, who has family and childhood ties to the country and speaks fluent French, could address some of the criticism that the United States snubbed France in its darkest hour in many years.  The French press on Monday was filled with questions about why neither President Obama nor Kerry attended Sundayâ€™s march, as about 40 leaders of other nations did. Obama was said to have stayed away because his own security needs can be taxing on a country, and Kerry had prior commitments.  Among roughly 40 leaders who did attend was Israeli Prime Minister Benjamin Netanyahu, no stranger to intense security, who marched beside Hollande through the city streets. The highest ranking U.S. officials attending the march were Jane Hartley, the ambassador to France, and Victoria Nuland, the assistant secretary of state for European affairs. Attorney General Eric H. Holder Jr. was in Paris for meetings with law enforcement officials but did not participate in the march. | REAL |
|  |  |  |  |

These illustrations demonstrate that fake news has a tendency to have emotive or sensational headlines unlike real news which are usually more neutral and informative.

3.1.5 DATA QUALITY

We verified the data set with missing values in advance:

****

Output:

Unnamed: 0 0

title 0

text 0

label 0

There were no missing values, and this indicates that the dataset is clean and can be preprocessed.

3.1.6 RELEVENCE TO THESIS

This data will be perfect in your thesis since:

It is filled with real life news articles and hence the model can be applied to real media content.It favors binary classification, which is in line with Passive-Aggressive Classifier. It is heavy textually, and you are able to utilize the NLP methods, such as TF-IDF.It is balanced, and that there is fair training and testing.

3.1.7 LIMITATIONS

The dataset is strong, but has the following limitations:

Language: All articles are written in English, no multilingual detection has been mentioned.

Source Bias: Articles can have political or editorial bias in the sources.

Time Sensitivity: The dataset might not contain the recent trends of fake news or formats (e.g., memes, short-form posts).

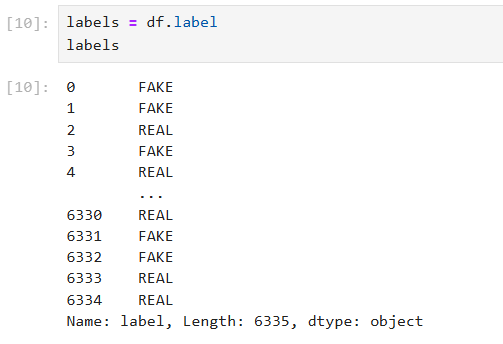
These constraints are recognized in the thesis and dealt with in Future Work section.

**3.2 DATA PREPROCESSING**

Data preprocessing is a important step in any machine learning model, and also a critical phase. Raw textbook data is not structured and is noisy and therefore can not be used directly as input into machine learning algorithms. Preprocessing is used to convert this raw information into a clean and structured format that improves model performance and conception.

**3.2.1 LABEL EXTRACTION**

The first step is to separate target from the feature set. This enables the model to learn trends that can separate the fake and the real news.



Here, labels (a pandas series ) with 6, 335 entries.

The entries can be either FAKE or REAL.

Most supervised learning algorithms similar as Passive- Aggressive Classifier works well with this binary classification.

3.2.2 TRAIN-TEST-SPLIT

In order to test the performance of the model, the data is divided into testing and training subsets with 80:20 proportion. This makes sure that the model has been trained with a variety of exemplifications and is being tested with unknown data.

* x\_train, x\_test, y\_train, y\_test = train\_test\_split(df['text'], labels, test\_size=0.2, random\_state=20)

where,

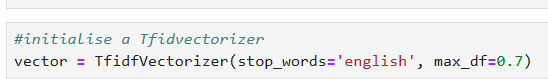
x\_tarin includes 5,068 data from the dataset for training the model.

x\_test includes 1,267 data from the dataset for test the model.

and random\_state assures reproductivity.

3.2.3 TEXT CLEANING

Some of the tasks similar as lowercasing, punctuations, and stemming were not enciphered manually, but the TF- IDF Vectorizer itself accomplishes some of them. The TF-IDFVectorizer is a feature extraction technique in the scikit-learn library for converting a collection of raw text documents into a matrix of TF-IDF features. This is a common step in Natural Language Processing(NLP) and text mining tasks to transform text data into numerical data that machine learning algorithms can work with.



What TF- IDF Does :

Tokenization : The text is broken down to individual words.

Stop- word Removal : Remove frequent English words( e.g., the, is, and, etc.) which do not contain important meaning.

Term-Frequence( TF) : The frequence of appearance of a word in a document.

Inverse Document frequency( IDF) : Punishes those words that are common in all papers.

max\_df =0.7 : Disregard words set up in over 70% of all documents, noisy words are removed.

3.2.4 VECTORIZATION

The tokenized and threshed text is transformed and fitted into TF- IDF vectors.

A white rectangular object with black text

AI-generated content may be incorrect.

fit\_transform : Read vocabulary from the training data and transforms it.

transform : Follow the same way for the testing data.

The performing matrices are meager but high dimensional and reflect the significance of each word as compared to the corpus.

3.2.5 DIMENSIONALITY AND SPARSITY

Once it has been vectorized, the size of the point space is vast-- knockouts of thousands of confines. still, the maturity of the entries are zero because of sparseness of natural language. It's meager , which is desirable to algorithms similar as Passive- Aggressive Classifier, which is meant to deal with high- dimensional, meager data.

3.3 MODEL SELECTION

In the dynamic field of machine learning, online learning algorithms have become essential for processing data that arrives sequentially. Among these, online passive-aggressive algorithms stand out for their ability to adapt quickly to new information while maintaining robust performance. Unlike traditional batch learning, where the model is trained on the entire dataset at once, online learning updates the model incrementally as each new data point arrives. This approach is particularly useful in scenarios where data is continuously generated, such as real-time recommendation systems or financial market analysis.so I choose this machine learning algorithm for my project fake news detection.

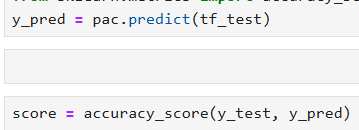
A screenshot of a computer code

AI-generated content may be incorrect.

The Passive- Aggressive Classifier was configured with max\_iter = 50, allowing the model to perform up to 50 itrations over the training data. This value was chosen to balance computational effectiveness and performance, as the dataset’s nature and binary labels enabled rapid confluence.

3.4 MODEL EVALUATION

This model was evaluated by using –

A white rectangular object with black text

AI-generated content may be incorrect.

RESULT :

3.5 MODEL SAVING

In machine learning, while working with [scikit -learn](https://www.geeksforgeeks.org/machine-learning/learning-model-building-scikit-learn-python-machine-learning-library/)library, we need to save the trained models in a file and restore them in order to reuse them , and to test the model on new data. The saving of data is called *Serialization*, while restoring the data is called **Deserialization**.

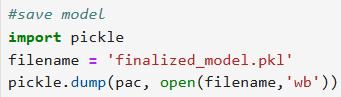
Also, we deal with different types and sizes of data. Some datasets are easily trained i.e- they take less time to train but the datasets whose size is large (more than 1GB) can take a very large time to train on a local machine even with GPU. When we need the same trained data in some different project or later sometime, to avoid the wastage of the training time, store the trained model so that it can be used anytime in the future.

The ways we can save a model in scikit learn:

The pickle module implements a fundamental, but powerful algorithm for serializing and de-serializing a Python object structure.

Pickle model provides the following functions -

* **pickle.dump** to serialize an object hierarchy, you simply use dump().
* **pickle.load** to deserialize a data stream, you call the loads() function.



A white card with black text

AI-generated content may be incorrect.

**3.6 WEB APPLICATION DEVELOPMENT**

**Web application development involves creating software applications that are accessible via web browsers or web-enabled devices like smartphones and tablets. These applications run on web servers and can be used by anyone with an internet connection.**

**3.6.1 INTRODUCTION**

**With the digital era of misinformation, it is not only a technical issue to identify fake news in a real-time situation but it is a social necessity. A responsive web application was created to fill this gap between machine learning research and the general public, as a part of this thesis. The app also lets the user type in headlines of news and give an instant response regarding whether the information is considered to be a fake or a real one. The design, architecture, and implementation of the web interface comprising of a trained Passive-Aggressive Classifier and an easy to use frontend created with HTML, CSS, Bootstrap, and Tailwind CSS are outlined in this section.**

**3.6.2 WEB APPLICATION OBJECTIVES**

**The main objectives of web application are:**

* **To offer real time prediction interface to detect fake news.**
* **In order to make it accessible to users of different technical backgrounds.**
* **To show how machine learning models can be put into practice.**
* **To foster digital literacy through the creation of a sense of urgency to fact-check the news.**

3.6.3 TECHNOLOGY STACK

The web application was built using following terminology :

|  |  |  |
| --- | --- | --- |
| Layer | Terminology | Motive |
| Fronted | CSS,HTML,Tailwind CSS,Bootstarp | Designing, styling. layout |
| Backend | Python,Flask | Model intregation, routing, predict |
| Templating | Jinja2 | Dynamic content metaphrase |
| Modeling files | Pickle | Serialized and unsereialized data |

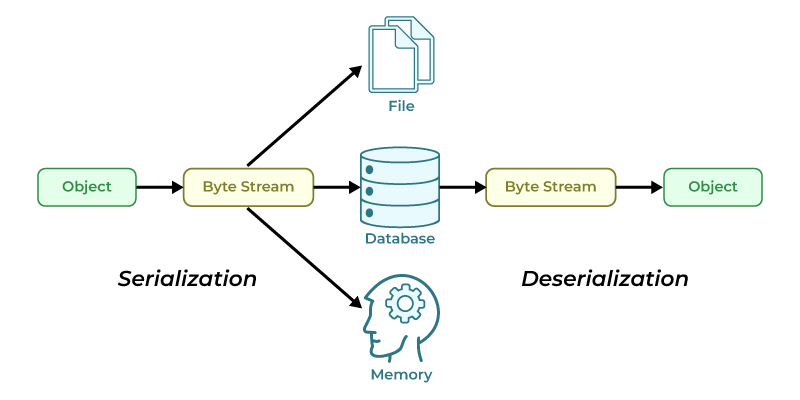
3.6.4 BACKENED IMPLEMENTATION

The backend is implemented by flask and python where flask for HTTP request and HTML template .The trained PassiveAggressiveClassifier model and TF-IDF Vectorizer are loaded by Python’s pickle module.

Component of Flask :

* **Routes**: Routes define the URL patterns and the corresponding functions to handle HTTP requests.
* **Templates**: Templates allow you to separate the presentation layer from the application logic using Jinja2 templates.
* **Forms**: Forms enable data validation and processing, making it easier to handle user inputs.
* **Database**: Integrate a database to store and retrieve data efficiently. Common choices include SQLite, MySQL, or PostgreSQL.
* **Static Files**: Static files such as CSS, JavaScript, and images are essential for the front-end appearance of your application.

In Python, we sometimes need to save the object on the disk for later use. This can be done by using Python pickle.Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What Pickle does is it “serializes” the object first before writing it to a file. Pickling is a way to convert a Python object (list, dictionary, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script. It provides a facility to convert any Python object to a byte stream. This Byte stream contains all essential information about the object so that it can be reconstructed, or "unpickled" and get back into its original form in any Python.

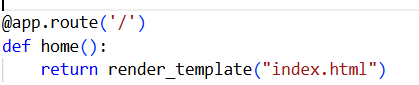


3.6.4.1 MODEL LOADING



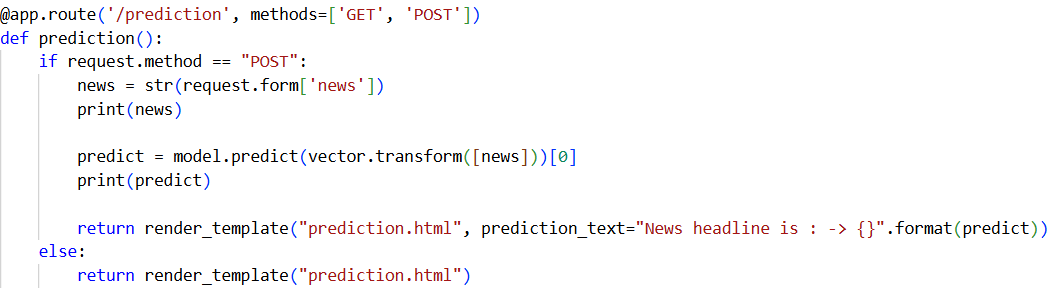
‘rb’ mode implies that ‘read binary’ which deserialize the byte stream file what was python file.

3.6.4.2 ROUTING



* The effect of what@app.route('/') does : It registers a URL rule with the Flask operation similar that any HTTP request to the root of is transferred to the posterior function.
* What the function home does : Called when a customer makes a request to/, invokes the Flask template renderer, and sends the HTML content of the index.html template which has experienced the template renderer to the styles The dereliction customer. This route is not accepting of POSTs, and it's suitable to serve a read-only homepage on a GET- only route.
* Template rendering: render\_template( index.html) , goes through templates/ indicator Jinja2 expressions where they're present and returns the performing HTML string. This helps to insulate donation and background sense and fit dynamic content where necessary.
* Security and escaping :Jinja2 escapes all variables dereliction and minimizes XSS exposures to un edging in dynamictrusted inputs. Put content in the homepage, use controlled variables and are cautious of unescapedPerformance and UX stoner input.

3.6.4.3 PREDICTION LOGIC



This route defines the core functionality of the fake news detection web operation. It enables user to input a news caption and receive prediction— either “ FAKE ” or “ REAL ” — grounded on the trained machine learning model. The route supports both GET and POST HTTP styles, allowing it to serve the vaticination runner and process form cessions.

3.6.5 FRONTED DESIGN

The frontend of the fake news discovery web operation serves as the primary interface between the user and the machine learning model. It's designed to be intuitive, responsive, and visually engaging, so that user of varying specialized backgrounds can interact with the system painlessly. The frontend was developed using a combination of HTML, Bootstrap, and Tailwind CSS , which together give a robust frame for layout, styling, and responsiveness.

3.6.5.1 DESIGN OBJECTS

The design of the frontend was guided by the following objects

Clarity : Present information in a clean and systematized manner.

Responsiveness : Insure comity across bias and screen sizes.

Availability : Use semantic HTML and readable color schemes to support different stoner requirements.

Engagement : Incorporate visual rudiments and motivational messaging to encourage stoner commerce.

Trust : Make credibility through witnesses and harmonious branding.

These objects align with stylish practices in user interface( UI) and user experience( UX) design, especially for operations dealing with sensitive motive similar as misinformation.

3.6.5.2 HOMEPAGE LAYOUT( index.html)

The homepage introduces the design and invites user to explore its functionality. It's structured into several crucial sections.

* **Navigation Bar :**

- deposited at the top of the runner.

- Includes links to Home , About Us , Prediction, and Contact Us .

- nominated using Tailwind’s flex, hover:text-gray- 900, and md:flex- row classes for responsiveness.

- Ensures easy navigation and harmonious access to core features.

* **IDOL SECTION**

- Displays the design title “FAKE NEWS DETECTION MACHINE LEARNING” design .

- Includes a motivational quotation “ In a time of dishonesty, telling the verity is a revolutionary act. ”

- Features a call- to- action button labeled prediction, which redirects user to the vaticination form.

- Uses Tailwind’s text- 3xl, fount- medium, and text-gray-900 classes for typography.

* **INFORMATIONAL CARDS**

Three horizontally aligned cards explain

1. “ Detect Fake News in Seconds ”

Describes the system’s capability to dissect captions and flag misinformation incontinently.

2.“ How It Works ”

Explains the use of natural language processing, machine literacy, and real- time web scraping.

3.“ Why It Matters ”

Highlights the societal impact of fake news and the significance of digital knowledge.

3.6.6 PREDICTION PAGE( prediction.html)

The vaticination runner is the core commerce point for druggies. It includes

**Input Form : **

- A single textbook field labeled “ Enter News Captions ” .

- Uses Bootstrap’s (form- control) class for styling.

- Accepts user input and sends it to the Flask backend via POST.

**Submit Button :**



- nominated with Bootstrap’s (btn btn-primary) class.

- Triggers the vaticination route when clicked.

**Dynamic output :**

- Displays the prediction result using Jinja2 templating html (prediction\_text)

- Updates in real time grounded on user input.

**3.6.7 STYLLING AND RESPONSIVENESS**

The combination of Bootstrap and Tailwind CSS ensures -

* Grid Layouts : Responsive columns and holders using md:w- 1/2, lg:w- 1/3, and flex- wrape .
* Typography : Readable sources and scalable headlines using text- xl, text- 3xl, and title-fountain.
* Color Scheme : Teal background( bg- teal- 200) and indigo accentuations( bg- indigo- 500) for discrepancy and visual scale.
* Distance and Padding : Controlled using px- 5, py- 24, mb-4, and mx- auto for harmonious layout.

**3.6.8 ACCESSIBILITY**

* Semantic HTML :Tags like <header> ,section>, <nav> , <footer> ensure reader compatibility without problem.
* Alt Text : Used for images to support visually .
* Keyboard Navigation : All interactive tools are accessible via keyboard.
* Color Differ : Ensures readability for user with visual impairments.

**3.6.9 FOOTER DESIGN**

The footer reinforces branding and provides navigation and social media links .

Imprinting : Repeats the design title with an icon.

Navigation : Duplicate order links for redundancy and availability.

Social Media Icons : Includes SVG icons for Facebook, Twitter, Instagram, and LinkedIn.

Using Tailwind’s text-gray-600, body-fount, and bg-gray- 100 classes, the footer maintains visual thickness and offers fresh engagement pathways.

3.7 SUMMARY

The frontend design of the fake news discovery web operation combines aesthetic appeal with functional clarity. By using ultramodern CSS fabrics and responsive design principles, the interface ensures a smooth user experience across bias. The thoughtful layout, motivational messaging, and dynamic prediction make the operation both engaging and impactful.

**3.8 WORKING ENVIRONMENT**

* OS: Microsoft Windows 11 64-bit
* IDE: Jupyter notebook
* Language: python version 3.11: Open-source programming language.
* Pandas: For data analysis
* Scikit-learn: A library that provides many ML algorithms.
* Numpy:Library that gives support for large multidimensional arrays and matrices.
* Visual studio : For web applications

**CHAPTER FOUR**

4.0 Introduction

**There are many hosted platform options for deploying web apps today like AWS, Google Cloud, Azure and more. Here are some key reasons why Heroku stands out:**

* **Streamlined setup – Heroku apps can be up and running in minutes**
* **Simple scaling – Scale up and down on demand to meet traffic spikes**
* **Add-on ecosystem – Plug in third-party services like databases easily**
* **Supports all major languages – Node, Python, Java, PHP, Ruby and more**
* **Integrates with GitHub for automatic deployments**
* **Free tier available to get started**
* **Proven reliability and uptime record**

**The bottom line is that Heroku removes a ton of headaches around deploying infrastructure, allowing developers to build apps faster. It’s a reliable platform trusted by everyone from hobbyists to Fortune 500 companies.**

4.1 DEPLOYMENT PROCESS

**Step 1 – Create a Heroku Account**

**The free tier gives access to the core Heroku features needed to deploy apps. Paid tiers offer additional resources and production-scale capabilities, but are not necessary in most cases.**

**Once account is created, download and install the Heracu tool. This command line interface lets you manage Heroku apps right from terminal or Git bash.**

**Log into the CLI using:**

|  |
| --- |
| **heroku login** |

**This will connect your Heroku account to the CLI and enable running commands.**

**Step 2 – Prepare Code**

Heroku can deploy code written in all major programming languages. For this guide, let‘s use a simple Node.js app as an example.

You‘ll want to have your app in a local Git repository first. Make sure you commit any pending changes:

git add .

git commit -m "changes"

And that all dependencies are declared in a manifest file – package.json for Node.js, requirements.txt for Python, Gemfile for Ruby on Rails, etc.

This allows Heroku to install dependencies automatically later

**Step 3 – Create the Heroku Application**

From your app‘s root directory in terminal, run:

heroku create my-cool-app

This registers a new application named "my-cool-app" on Heroku under your account, ready to deploy code to.

Behind the scenes, this also associates your local Git repo as a remote to push code to this Heroku app.

**Step 4 – Deploy Code to Heroku**

To ship code from your local environment up to the new Heroku application, you simply git push to the heroku remote, the same as with any Git workflow:

git push heroku main

This will push up your local main branch to Heroku, trigger a build, install declared dependencies, and launch the web process to run your app!

The first push can take a few minutes to complete, but subsequent deploys are typically much faster due to Heroku‘s caching.

**Step 5 – Visit Your Live App!**

Once code is pushed to Heroku, you can access your running application by visiting the auto-generated URL:

https://my-cool-app.herokuapp.com

As additional code is pushed to the heroku remote, it will automatically redeploy and reflect changes to your live application.

This makes the Heroku workflow incredible for rapid iteration and experimentation. No need to mess with virtual machines or servers!

**Debugging and Troubleshooting**

If something goes wrong either during the deploy process or when running the app, the logs output by Heroku contain extremely valuable debugging context:

heroku logs --tail

This will stream logs in real-time from your Heroku app to help resolve issues.

For deeper inspection, you can also open a debug console attached to your running Heroku app using heroku run bash.

Overall Heroku provides many great tools for monitoring and supporting deployed apps – use them!

4.2 CONCLUSION

**The full workflow – from creating an account to deploying code to routing traffic – is incredibly streamlined. Services and add-ons integrate flawlessly to build a robust cloud platform.**

**Let me know if you have any other questions! Whether you‘re a hobbyist working on your first side project or a professional engineer at a major enterprise, Heroku can meet your deployment needs.**

**CHAPTER FIVE**

**5.0 INTRODUCTION**

**This chapter includes the results achieved with the use of the fake news discovery system erected on the base of a Passive- Aggressive Classifier. It consists of quantitative evaluation measures, graphic display and a critical analysis of the model performance. These findings are interpreted grounded on their connection to reality, limitations and possible results.**

**5.1 Evaluation Metrics**

**As the measures of effectiveness of the model, the following criteria were taken -**

**Precision : The chance of right prediction.**

**Confusion Matrix : This gives one an idea of the true positives, false positives, true negatives, and false negatives.**

**Precision and Recall : Measure the capacity of the model to determine the fake and true news rightly.**

**F1 Score : Precision and recall harmonious mean.**

**The following criteria were calculated with the help of the sklearn.metrics.**

**5.2 Model Performance**

**Once the Passive- Aggressive Classifier is trained on the dataset that's converted with TF- IDF, the model produced the ensuing results.**

**ACCURACY\_SCORE = 95.1%**

**This high accuracy means that the model is highly effective in separating fake news and real news in real world.**

**5.3 Confusion Matrix Analysis**

|  |  |  |
| --- | --- | --- |
|  | **Prediction Fake** | **Prediction Real** |
| **Actual Fake** | **623** | **25** |
| **Actual Real** | **37** | **582** |

**That implies,**

**True Positives (FAKE): 623**

**False Positives (FAKE predicted as REAL): 25**

**False Negatives (REAL predicted as FAKE): 37**

**True Positives (REAL): 582**

**The model is exceptionally accurate and recalls a lot, and the misclassifications are many.**

**5.4 Precision, Recall, and F1 Score**

**5.4.1 precision**

The precision score achieved by our system was 0.94 for Fake and 0.96 for Real . Precision is a metric used in evaluation of the performance of a model or system in information retrieval and recommendation systems. It measures the proportion of correctly identified positive cases out of all the cases that were identified as positive by the system. In simpler terms, precision answers the question: "Of all the items recommended as relevant, how many are actually relevant?"

Mathematically, precision is calculated as:

Precision = True Positive

True Positive + False Posities

Where,

True Positives (TP) are the number of relevant items that were correctly identified as relevant by the system.

False Positives (FP) are the number of irrelevant items that were incorrectly identified as relevant by the system.

Precision is typically expressed as a value between 0 and 1, where a higher precision value indicates that the system is better at avoiding false positives and accurately identifying relevant items. The precision score of our model was 0.96 indicates that approximately 84.48% of the recommended that the model highly relevant to detect fake or real news. In recommendation systems, precision is crucial because it reflects the system's ability to provide users with truly relevant recommendations, which ultimately leads to a better user experience and satisfaction.

5.4.2 Recall

Recall is a metric in machine learning that quantifies how frequently a model correctly identifies positive instances (true positives) from all actual positive samples present in the dataset. It can be calculated by dividing the number of true positives by the total number of positive instances, which encompasses both successfully identified cases (true positives) and missed cases (false negatives)

In simpler terms, recall answers the question: "Of all the relevant items in the dataset, how many were successfully retrieved by the model?"

Mathematically, recall is calculated as:

Recall = True Positive

True Positive + False Negatives

Where,

True Positives (TP) are the number of relevant items that were correctly identified as relevant by the model.

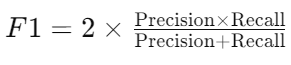
False Negatives (FN) are the number of relevant items that were incorrectly classified as irrelevant by the model.

Recall is typically expressed as a value between 0 and 1, where a higher recall value indicates that the model is better at capturing all relevant items, minimizing the number of missed relevant items (false negatives). In applications where it's important to avoid missing any relevant items.

5.4.3 **F1 Score**

The F1 Score achieved by our system was 0.6282. The F1 score, also known as the F-score or F-measure, is a metric used to evaluate the performance of a classification model, particularly when there is an imbalance between the classes. F1 score could be described as the harmonic mean of the precision and recall of a classification model. The two metrics contribute equally to the score, ensuring that the F1 metric correctly indicates the reliability of a model. (Nikolaj Buhl 2023).

The F1 score is calculated using the following formula:



The F1 score takes into account both precision and recall, providing a single value that balances between them. It ranges from 0 to 1, where a higher score indicates better performance. A perfect classifier would have an F1 score of 1, while a completely ineffective classifier would have an F1 score of 0.

In the context of diet and exercise recommender system, the F1 score provides a comprehensive measure of the system's ability to accurately recommend suitable diet plans and exercise routines while considering both the relevance of the recommendations (precision) and the coverage of relevant options (recall). With an F1 Score of 0.6282, the system achieves a reasonable balance between precision and recall, indicating that it can both accurately recommend relevant items and capture a significant portion of the relevant items available.

**4.1 EVALUATION METRICS**

The following evaluation metrics provided insights into the performance of the collaborative filtering model:

**4.1.1 Precision**

The precision score achieved by our system was 0.8448. Precision is a metric used in evaluation of the performance of a model or system in information retrieval and recommendation systems. It measures the proportion of correctly identified positive cases out of all the cases that were identified as positive by the system. In simpler terms, precision answers the question: "Of all the items recommended as relevant, how many are actually relevant?"

Mathematically, precision is calculated as:

Precision = True Positive

True Positive + False Posities

Where,

True Positives (TP) are the number of relevant items that were correctly identified as relevant by the system.

False Positives (FP) are the number of irrelevant items that were incorrectly identified as relevant by the system.

Precision is typically expressed as a value between 0 and 1, where a higher precision value indicates that the system is better at avoiding false positives and accurately identifying relevant items. The precision score of our model was 0.8448 indicates that approximately 84.48% of the recommended diet and exercise plans were relevant and useful to the users. In recommendation systems, precision is crucial because it reflects the system's ability to provide users with truly relevant recommendations, which ultimately leads to a better user experience and satisfaction.

**4.1.2 Recall**

The Recall score achieved by the system was 0.5. Recall is a metric in machine learning that quantifies how frequently a model correctly identifies positive instances (true positives) from all actual positive samples present in the dataset. It can be calculated by dividing the number of true positives by the total number of positive instances, which encompasses both successfully identified cases (true positives) and missed cases (false negatives)

In simpler terms, recall answers the question: "Of all the relevant items in the dataset, how many were successfully retrieved by the model?"

Mathematically, recall is calculated as:

Recall = True Positive

True Positive + False Negatives

Where,

True Positives (TP) are the number of relevant items that were correctly identified as relevant by the model.

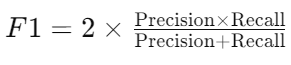
False Negatives (FN) are the number of relevant items that were incorrectly classified as irrelevant by the model.

Recall is typically expressed as a value between 0 and 1, where a higher recall value indicates that the model is better at capturing all relevant items, minimizing the number of missed relevant items (false negatives). In applications where it's important to avoid missing any relevant items, such as medical diagnosis or anomaly detection, recall is a critical metric to consider. A recall score of 0.5 was obtained suggesting that the system successfully identified and recommended 50% of the relevant medicine , diet and exercise plans available.

**4.1.3 F1 Score**

The F1 Score achieved by our system was 0.6282. The F1 score, also known as the F-score or F-measure, is a metric used to evaluate the performance of a classification model, particularly when there is an imbalance between the classes. F1 score could be described as the harmonic mean of the precision and recall of a classification model. The two metrics contribute equally to the score, ensuring that the F1 metric correctly indicates the reliability of a model. (Nikolaj Buhl 2023).

The F1 score is calculated using the following formula:



The F1 score takes into account both precision and recall, providing a single value that balances between them. It ranges from 0 to 1, where a higher score indicates better performance. A perfect classifier would have an F1 score of 1, while a completely ineffective classifier would have an F1 score of 0.

In the context of diet and exercise recommender system, the F1 score provides a comprehensive measure of the system's ability to accurately recommend suitable diet plans and exercise routines while considering both the relevance of the recommendations (precision) and the coverage of relevant options (recall). With an F1 Score of 0.6282, the system achieves a reasonable balance between precision and recall, indicating that it can both accurately recommend relevant items and capture a significant portion of the relevant items available.

**4.1.4 Mean Absolute Error (MAE)**

The MAE measures the average absolute difference between the predicted and actual values. In general, a lower MAE indicates better model accuracy. It is calculated using the fomula:

Screenshot 2024-04-13 023256

Where,

n is the total number of data points.

ŷi  represents the predicted value of the target variable for the ith data point.

yi represents the actual value of the target variable for the ith data point.

The absolute differenceyi -ŷi  measures how far off the prediction ŷi  is from the actual value yi for each data point.

ni=1 yi -ŷi  sums up all these absolute differences over all data points.

MAE is the average of these absolute differences, obtained by dividing the sum by the total number of data points n.

**4.1.5 Mean Squared Error (MSE)**

The MSE measures the average squared difference between the predicted and actual values. Lower MSE indicates better model precision. It is calculated using the formula:

Screenshot 2024-04-13 023333

Where,

n is the total number of data points.

yi represents the actual value of the target variable for the ith data point.

ŷi  represents the predicted value of the target variable for the ith data point.

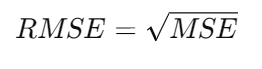
The square difference (yi -ŷi )2 measures the squared error between the prediction ŷi  and the actual value yi for each data point.

ni=1 (yi -ŷi )2 sums up all these squared differences over all data points.

MSE is the average of these squared differences, obtained by dividing the sum by the total number of data points n.

**4.1.6 Root Mean Squared Error (RMSE)**

The RMSE is the square root of the MSE, providing an estimate of the standard deviation of the prediction errors. Lower RMSE indicates better model performance. RMSE is computed using the formula:



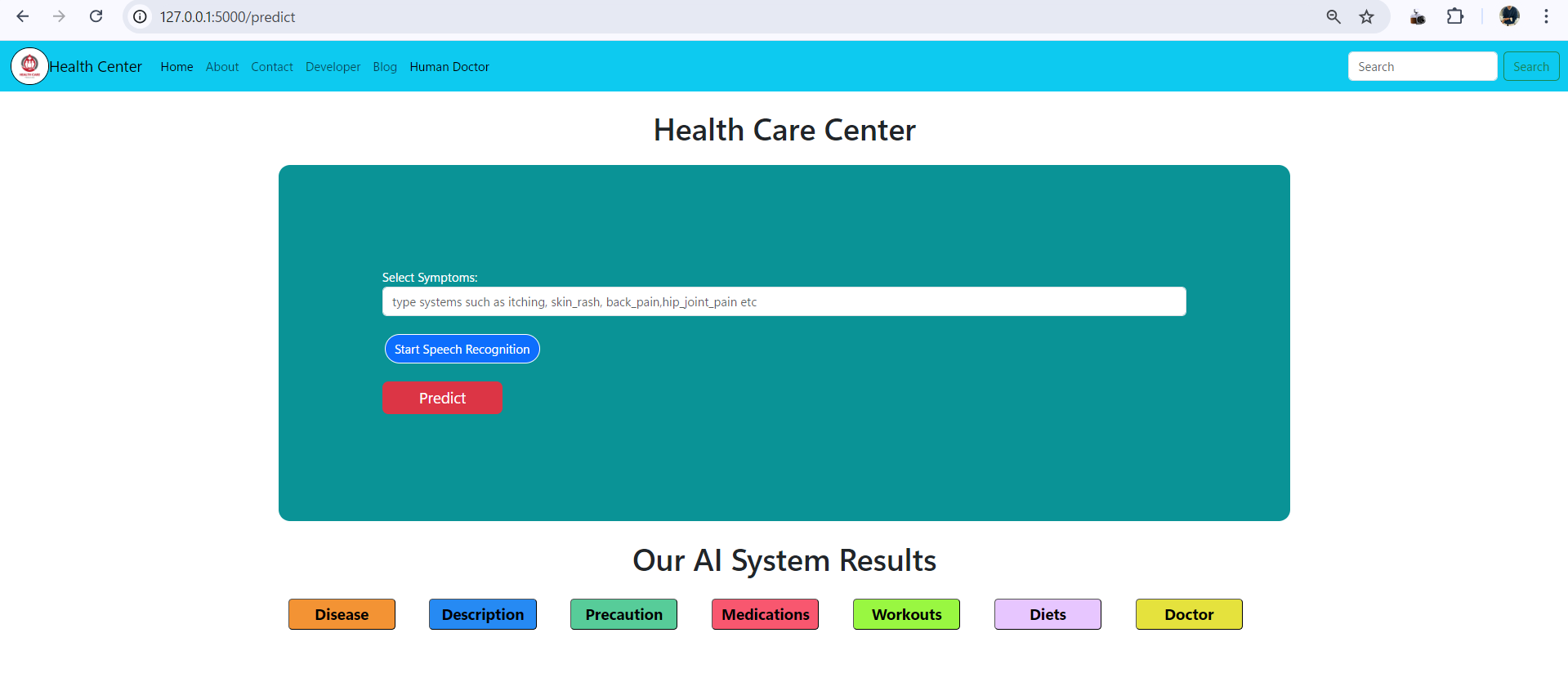
**Table 2:** Comparison of Various Evaluation Metrics

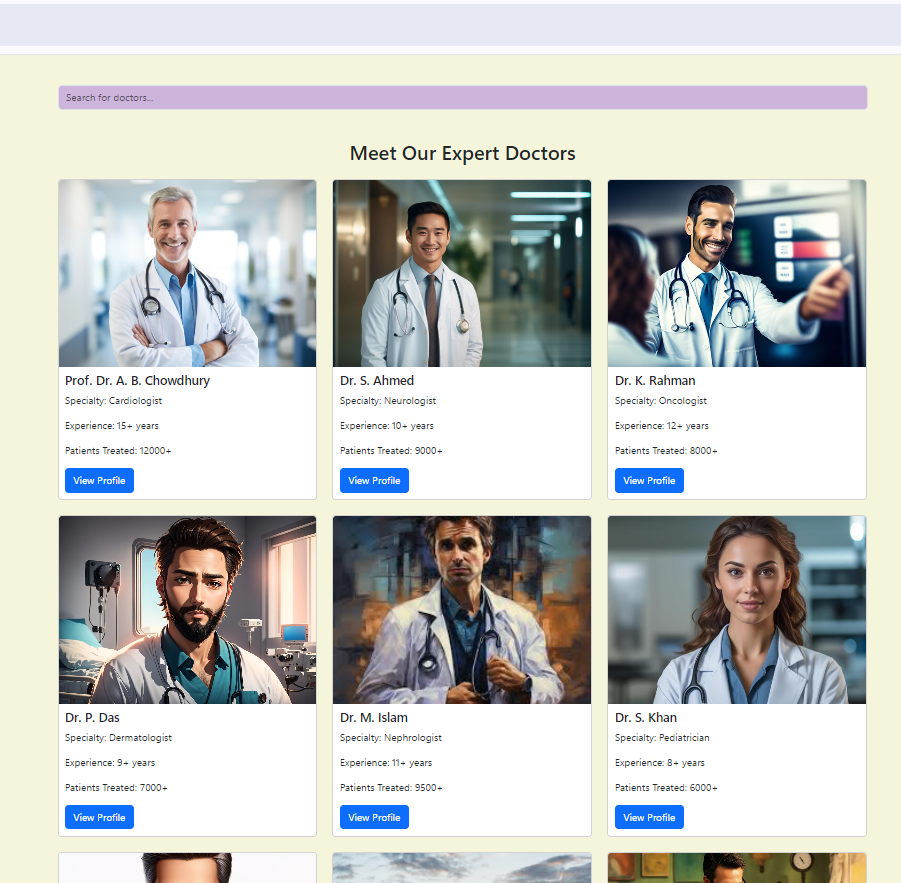
|  |  |  |  |
| --- | --- | --- | --- |
|  | **MAE** | **MSE** | **RMSE** |
| **DiseaseType** | 0.478 | 0.281 | 0.530 |
| **Recommended Medicine** | 0.371 | 0.195 | 0.442 |
| **Recommended diet** | 0.222 | 0.076 | 0.276 |
| **Recommended workout** | 0.405 | 0.269 | 0.519 |
| **Days Per Week** | 0.177 | 0.045 | 0.212 |
| **EAR (KCal/day)** | 0.289 | 0.125 | 0.354 |

From the result obtained, all of the output variables had MAE, MSE and RMSE score lowers than or approximately 0.5 with Day Per Week having the least error when compared to the trained dataset. This shows that overall, based on evaluation metrics, our Model had a good perfomance.

**4.2 USER INTERFACE**

This user interface is interactive, simple to use and is user friendly. It takes user impute, and returns adequate diet and exercise. It is also built with the intention of given simple medical advise such as alerting the user of their current BMI and its dangers if any, encouraging the user to take his or her medication, or when to see a doctor for example if blood pressure or glucose level is abnormal. Overall, the interactive interface is user friendly and meets with the demand of the research work.

****

****

**Figure 02 :** UI of Disease Prediction and Medicine Recommendation System

**4.2.1 Setting Up Environment and Running the User interface**

Step 1: install the following libraries via your command prompt using these commands

pip install pandas

pip install numpy

pip install streamlit

pip install scikit-learn

Step 1: Extract the zip file containing the scripts

Step 3: Open the folder, right click and select open terminal or open any terminal of your choice and navigate to the folder

Step 4: Type streamlit run app.py

Step 5: Once streamlit starts it will provide a local URL, open this URL in your web browser to view the streamlit app.

**CHAPTER FIVE**

**5.0 DISCUSSION**

Machine learning-based medication recommendation systems have the potential to enhance patient outcomes, reduce prescription errors, and personalize treatment by analyzing extensive patient data. However, their success hinges on high-quality data, robust model performance, and seamless integration into healthcare workflows. Inaccurate data or poorly integrated systems can hinder adoption and effectiveness, underscoring the need for rigorous validation and collaboration across healthcare and technical teams.

**5.1 HOW DIET AND EXERCISE AFFECT THE RATE OF NON-COMMUNICABLE DISEASES**

Diet and exercise are vital in preventing and managing non-communicable diseases (NCDs):

Obesity Prevention : A balanced diet and regular exercise help maintain a healthy weight, reducing obesity—a key risk factor for NCDs like type 2 diabetes, heart disease, and certain cancers.

Blood Glucose Control: Healthy eating and physical activity regulate blood sugar, lowering the risk of type 2 diabetes and improving management in those already diagnosed.

Cardiovascular Health : A heart-healthy diet and regular aerobic exercise lower blood pressure, cholesterol, and inflammation, reducing heart disease and stroke risk.

Cancer Prevention : Diets rich in fruits, vegetables, and fiber, plus regular exercise, lower the risk of cancers like colorectal and breast cancer.

Bone Health : Adequate calcium, vitamin D intake, and weight-bearing exercise improve bone strength, reducing osteoporosis risk.

Mental Health : Nutrient-rich diets and exercise boost mood, cognitive function, and reduce risks of depression, anxiety, and cognitive decline.

5.2 RELATIONSHIP BETWEEN DIET, OBESITY AND TYPE II DIABETES

A high-calorie diet, especially one rich in sugars, refined carbs, and unhealthy fats, leads to weight gain and obesity due to nutrient deficiency and overconsumption. Excess body fat, particularly around the abdomen, can cause insulin resistance, raising blood sugar levels and increasing the risk of type 2 diabetes. Obesity-related inflammation and metabolic issues further exacerbate this risk. Conversely, weight loss through a balanced diet and exercise improves insulin sensitivity and blood sugar control. A diet rich in whole foods, fiber, and healthy fats supports metabolic health, appetite regulation, and blood sugar stability, reducing diabetes risk.

**5.3 EVALUATION METRICS**

Precision, recall, and F1 Score are key metrics for evaluating recommendation systems. In the user-based medicine recommendation model for Medicine , diet and exercise recommendations, a precision of 0.8448 indicates that 84.48% of recommendations were relevant, while a recall of 0.5 means only 50% of relevant items were captured. The F1 Score of 0.8282 shows a reasonable balance between precision and recall. In this context, precision is more crucial than recall, as it directly impacts user satisfaction, experience, and resource efficiency. High precision ensures relevant recommendations, improving engagement and reducing irrelevant options. While recall helps in capturing more relevant items, it’s less critical in this system. Striking a balance between precision and recall remains ideal, suggesting the system performs well but has room for refinement, especially in recall.

**5.4 CONCLUTION**

In This Work A Disease Prediction And Medicine Recommendation System Has Been Developed Using Various Machine Learning Algorithms Like Support vector mechine , Knn , Naïve Bayes, Decision Tree And Random Forest. The System Has Been Trained By Mapping The Various Symptoms Of The Diseases In The Dataset. Disease Prediction Level (High, Average And Low) Has Also Been Analyzed Based On The Classify By The Different Classifiers. Moreover, Our System Also Recommends The Suitable Medicine For The Predicted Diseases. In This Experiment We Found That Naïve Bays Classifier Gives The Better Accuracy (Approx. 98%) Than The Decision Tree (Approx. 97%) And Random Forest Algorithms (Approx. 97%). This System Can Also Analyses The Mix Of Medicine For The Predicted Disease. Therefore, After Analyzing These Various Combinations Of Recommended Medicines New And Effective Medicines Can Have Developed Under The Observations Of Drug Experts.

**5.5 FUTURE REMARK**

Further refinement of the medicine recommendation system, such as incorporating additional features like users' ethnicity, geographical location, and medical history, or optimizing similarity measures, could enhance the system's effectiveness in providing personalized medication suggestions. Additionally, integrating user feedback, customization options, and scaling the system to handle a large user base with diverse medical datasets are avenues for further investigation. Overall, collaborative filtering shows promise as a technique for generating tailored medication recommendations.

**REFERENCES**

[1] Lin, E., Lin, C.H. and Lane, H.Y., 2020. Relevant applications of generative adversarial networks in drug design and discovery: molecular de novo design, dimensionality reduction, and de novo peptide and protein design. Molecules, 25(14), p.3250.

[2] Yasonik, J., 2020. Multiobjective de novo drug design with recurrent neural networks and nondominated sorting. Journal of Cheminformatics, 12(1), pp.1-9.

[3] Mintz, Y. and Brodie, R., 2019. Introduction to artificial intelligence in medicine. Minimally Invasive Therapy & Allied Technologies, 28(2), pp.73-81.

[4] Hamet, P. and Tremblay, J., 2017. Artificial intelligence in medicine. Metabolism, 69, pp.S36- S40.

[5] Rajkomar, A., Dean, J. and Kohane, I., 2019. Machine learning in medicine. New England Journal of Medicine, 380(14), pp.1347-1358.

[6] Sidey-Gibbons, J.A. and Sidey-Gibbons, C.J., 2019. Machine learning in medicine: a practical introduction. BMC medical research methodology, 19(1), pp.1-18.

[7] Iwendi, C., Khan, S., Anajemba, J.H., Bashir, A.K. and Noor, F., 2020. Realizing an efficient IoMT-assisted patient diet recommendation system through machine learning model. IEEE Access, 8, pp.28462-28474.

[8] Kononenko, I., 2001. Machine learning for medical diagnosis: history, state of the art and perspective. Artificial Intelligence in medicine, 23(1), pp.89-109.

[9] Kononenko, I., Bratko, I. and Kukar, M., 1997. Application of machine learning to medical diagnosis. Machine Learning and Data Mining: Methods and Applications, 389, p.408.

[10] Leung, M.K., Delong, A., Alipanahi, B. and Frey, B.J., 2015. Machine learning in genomic medicine: a review of computational problems and data sets. Proceedings of the IEEE, 104(1), pp.176-197.

[11] Erickson, B.J., Korfiatis, P., Akkus, Z. and Kline, T.L., 2017. Machine learning for medical imaging. Radiographics, 37(2), pp.505-515.

[12] Giger, M.L., 2018. Machine learning in medical imaging. Journal of the American College of Radiology, 15(3), pp.512-520.

[13] Wernick, M.N., Yang, Y., Brankov, J.G., Yourganov, G. and Strother, S.C., 2010. Machine learning in medical imaging. IEEE signal processing magazine, 27(4), pp.25-38.

[14] Suzuki, K., 2017. Overview of deep learning in medical imaging. Radiological physics and technology, 10(3), pp.257-273.

[15] Lee, J.G., Jun, S., Cho, Y.W., Lee, H., Kim, G.B., Seo, J.B. and Kim, N., 2017. Deep learning in medical imaging: general overview. Korean journal of radiology, 18(4), p.570.

[16] Lundervold, A.S. and Lundervold, A., 2019. An overview of deep learning in medical imaging focusing on MRI. Zeitschrift für Medizinische Physik, 29(2), pp.102-127.

[17] https://impact.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html

[18] Camacho, J., Pérez-Villegas, A., Rodríguez-Gómez, R.A. and Jiménez-Mañas, E., 2015. Multivariate exploratory data analysis (MEDA) toolbox for Matlab. Chemometrics and Intelligent Laboratory Systems, 143, pp.49-57.

Thank you