Disease Prediction & Medical Information Providing System

Shivam Kumar Singh
210150022
shivam.ks@iitg.ac.in

***Abstract*—The healthcare domain is undergoing a transformative era, fueled by rapid technological advancements and the exponential growth of patient data. Managing this vast and complex data landscape is a formidable challenge. Big Data Analytics emerges as a vital solution, with Electronic Health Records (EHRs) standing as one of the prime examples of its application in healthcare. Big data analytics not only streamlines data management but also empowers healthcare professionals with previously unattainable insights.**

The following project report details the development of a Disease Prediction System that integrates frontend technologies like HTML, CSS, JavaScript, and Bootstrap with Python Flask as the backend. The system employs machine learning algorithms for disease prediction, web scraping for disease description retrieval, and prescription extraction, aiming to offer accurate disease predictions and relevant medical information.

I. Introduction & Background

A. Background

The healthcare sector faces growing challenges in accurately predicting diseases and providing comprehensive medical information to healthcare professionals and patients. Manual diagnosis methods often fall short in leveraging the abundance of healthcare data, leading to delays, misdiagnosis, and suboptimal patient care. Consequently, there is a critical need for advanced technologies that harness this data for accurate disease prediction and medical information retrieval.

Leveraging Big Data Analytics and Machine Learning, this project aims to address these challenges by developing a Disease Prediction System. The system will utilize machine learning algorithms, including Multinomial Naive Bayes, Decision Tree, and Random Forest, to predict diseases based on symptoms. Additionally, it will incorporate web scraping techniques to retrieve detailed disease descriptions and medicine prescriptions, empowering healthcare providers and individuals with comprehensive information for informed decision-making.

B. Motivation

The motivation behind this project stems from the potential transformative impact it can have on healthcare practices. By

harnessing the power of machine learning and web scraping technologies, the project aims to enhance disease prediction accuracy, thereby enabling early diagnosis, appropriate treatment planning, and improved patient outcomes. The seamless integration of these technologies seeks to bridge the gap in existing healthcare systems, offering a unified platform for disease prediction and medical information dissemination.

This project's motivation also lies in the empowerment of healthcare professionals, patients, and individuals by providing them with accurate disease predictions and detailed medical information. The ultimate goal is to contribute to the advancement of healthcare systems by leveraging cutting-edge technologies for the benefit of society.

II. Scope of the Project

This project's scope is comprehensive and encompasses several key areas:

A. Data Preprocessing: Multihot Encoding of Symptoms

During the data preprocessing phase, the raw dataset containing symptoms underwent transformation into a multihot encoded format. This process involved several key steps:

- **Symptom Extraction:** Initially, the dataset consisted of raw symptom data associated with various diseases. Each instance represented a list of symptoms linked to a particular ailment.
- **Creating Symptom Dictionary:** To perform multihot encoding, a symptom dictionary was created, comprising unique symptoms extracted from the dataset. This step involved identifying and compiling all distinct symptoms present across different instances.
- **Encoding Symptoms:** Using the symptom dictionary, each instance was encoded into a multihot format. For every disease entry, the presence or absence of symptoms was represented by binary values (0 or 1) in an array, where 1 indicated the presence of a symptom and 0 denoted its absence.
- **Final Multihot Encoded Dataset:** The processed dataset was transformed into a multihot encoded representation, where each row represented a disease and columns corresponded to symptoms. This encoded format facilitated the training of machine learning models by converting categorical symptom data into a numerical format suitable for analysis.

The multihot encoded representation of symptoms provided a structured format for training the machine learning models, enabling them to learn associations between symptoms and diseases effectively.

B. Machine Learning Model Development

This project will leverage Python to develop and implement machine learning models. These models will be constructed using a suite of diverse machine learning algorithms to ensure a robust and accurate disease prediction system.

- **Naive Bayes:** Multinomial Naive Bayes, known for its simplicity and effectiveness in text classification, will be employed for disease prediction based on symptom data.
- **Decision Tree:** The Decision Tree algorithm, offering interpretable decision rules, will be utilized to create a predictive model based on a hierarchy of features.

- **Random Forest:** A collection of Decision Trees forming a forest will be used to enhance prediction accuracy and handle overfitting issues.

The development of these machine learning models will involve extensive training and fine-tuning to ensure optimal performance in disease prediction based on symptoms.

C. Medical Information Extraction

Medical information extraction involves two critical steps:

1. Extracting Disease Description:

To acquire comprehensive disease descriptions corresponding to the predicted disease, BeautifulSoup, a Python library, is utilized. The system extracts information from the following link: https://www.diseaseinfosearch.org/search?term=disease. BeautifulSoup parses the HTML content of the webpage and retrieves detailed disease descriptions, aiding in providing valuable insights into the identified disease.

2. Extracting Medicine Information:

The system extracts medicine-related information associated with the predicted disease through a two-step process involving Selenium and Chrome WebDriver. Selenium facilitates dynamic content extraction by automating web interactions. The system navigates to the following link: https://search.medscape.com/search/?q=%22disease%22&plr=ref&contenttype=Drugs+%26+Neutraceuticals&page=1. By dynamically interacting with the webpage elements using Chrome WebDriver, it retrieves medicine-related details, including dosages, uses, adverse effects, and other pertinent information, aiding healthcare professionals and individuals in understanding the recommended medicines for the identified disease.

D. Model Evaluation

The performance of the disease prediction model will be rigorously evaluated using diverse test datasets. Comprehensive metrics, including accuracy, precision, recall, F1-score, and area under the curve (AUC), will be employed to gauge the model's efficacy.

E. User Interface Development

The User Interface (UI) was a pivotal component of the project, designed to provide an intuitive and interactive platform for users to input symptoms and receive disease predictions. The UI was constructed using a combination of HTML, CSS, JavaScript, and Bootstrap, offering a seamless experience for both healthcare professionals and individuals interacting with the system.

- **HTML Structure:** HTML served as the foundation of the UI, defining the structure and layout of the web-based interface. It outlined various sections, such as symptom input fields, prediction results display area, and navigation elements.
- **CSS Styling:** CSS was employed to enhance the visual appeal and aesthetics of the UI. It provided customization options for colors, fonts, layouts, and overall styling, ensuring a visually appealing and user-friendly interface.
- **JavaScript Functionality:** JavaScript was utilized to incorporate interactive elements and functionality within the UI. It enabled dynamic behavior, such as real-time validation of symptom inputs, form

submission handling, and asynchronous communication with the backend server.

- **Bootstrap Framework:** Bootstrap, a front-end framework, was leveraged to expedite the UI development process. It offered pre-designed components, responsive layout utilities, and CSS styles, allowing for rapid prototyping and ensuring compatibility across various devices and screen sizes.

The collaboration of HTML, CSS, JavaScript, and Bootstrap in constructing the UI facilitated a seamless and user-friendly experience. Healthcare professionals and users were able to input symptoms effortlessly and obtain disease predictions efficiently, contributing to the accessibility and usability of the system.

F. Backend Development with Flask

The backend includes a robust infrastructure to support the seamless integration of the frontend with the Flask framework. Flask provides the necessary routes to handle requests from the user interface, ensuring smooth data transmission between the frontend and the backend components.

Utilizing Flask, we have designed specific endpoints to receive symptom inputs from the user interface. These endpoints process the received data, allowing the backend to perform disease predictions using machine learning algorithms.

The Flask backend efficiently handles requests and orchestrates the execution of prediction algorithms based on the symptom data received. This ensures a user-friendly experience by providing quick and accurate disease predictions in response to user inputs.

Additionally, Flask's flexibility allows for easy integration of the backend with different frontend technologies, ensuring a versatile and adaptable system architecture.

The robustness of Flask in managing routes, processing data, and interfacing with machine learning models facilitates the seamless functioning of the entire Disease Prediction System, making it accessible to healthcare professionals and individuals alike.

G. Data Source

We will utilize a publicly available dataset for this project, which can be found at the following link: https://www.kaggle.com/datasets/kaushil268/disease-prediction-using-machine-learning/.

This dataset will serve as a valuable resource for training and evaluating our disease prediction models. This dataset has 132 parameters on which 42 different types of diseases can be predicted.

III. Problem Statement

The healthcare industry is confronted with the challenge of accurate disease prediction and comprehensive disease information dissemination. Traditional diagnostic methods often rely on manual analysis and may not effectively utilize the wealth of available healthcare data. As a result, healthcare professionals may face difficulties in accurately predicting diseases based on symptoms and providing detailed information about those diseases.

Existing healthcare systems may not fully harness the potential of modern technologies, such as machine learning and web

scraping, to improve disease prediction accuracy and facilitate easy access to disease-related information. Furthermore, the absence of an integrated system that effectively combines these technologies impedes the efficient provision of disease predictions, descriptions, and medicine prescriptions to healthcare providers and individuals.

The inadequacy of current healthcare systems in leveraging the power of data-driven approaches leads to delayed or inaccurate diagnoses, potentially impacting patient outcomes and healthcare costs. There is a pressing need for an integrated solution that incorporates machine learning algorithms for precise disease prediction and web scraping techniques for obtaining detailed disease descriptions and medicine prescriptions, empowering healthcare professionals with accurate and accessible information.

The primary challenges addressed by this project include:

- **Accuracy in Disease Prediction**: Developing a system that accurately predicts diseases based on symptoms, leveraging machine learning algorithms trained on diverse datasets.
- **Comprehensive Disease Information Retrieval**: Implementing web scraping techniques to gather detailed disease descriptions and medicine prescriptions from external sources to provide comprehensive information.
- **Integration of Technologies**: Integrating machine learning models and web scraping functionalities seamlessly into a unified backend system to offer accurate predictions and information retrieval through a user-friendly interface.
- **Efficient Healthcare Support**: Creating a system that assists healthcare providers and individuals in making informed decisions regarding disease identification, treatment, and management by providing reliable predictions and detailed disease-related information.

The absence of a unified system combining machine learning-based disease prediction and web scraping-based information retrieval poses significant challenges in healthcare, necessitating the development of an integrated Disease Prediction System to address these shortcomings.

IV. Architectural Details

The Disease Prediction System incorporates a robust architecture that seamlessly integrates frontend technologies with Python Flask serving as the backend. This system is designed to facilitate user interaction and efficient backend processing, ensuring accurate disease prediction and detailed disease-related information retrieval.

A. Frontend Technologies Integration

The system's frontend is developed using HTML, CSS, JavaScript, and Bootstrap, providing a user-friendly interface for symptom input and interaction. These technologies collectively enable the creation of an intuitive and responsive graphical user interface (GUI) that allows users to input symptoms effortlessly.

B. Backend with Python Flask

Python Flask serves as the backend framework for the Disease Prediction System. Flask's versatility and ease of use make it an ideal choice for handling various routes and functionalities required for disease prediction, information retrieval, and seamless

communication between the frontend and backend.

1) Predict Disease Route: The Predict Disease route within Flask integrates machine learning algorithms, including Multinomial Naive Bayes, Decision Tree, and Random Forest. These algorithms, trained on diverse datasets containing symptoms and corresponding diseases, enable accurate disease prediction upon receiving symptom inputs from users.

2) Get Disease Description Route: Leveraging Flask's capabilities along with web scraping techniques using Beautifulsoup, the Get Disease Description route fetches detailed disease descriptions from external websites. Upon receiving a predicted disease, this route retrieves and delivers comprehensive information about the disease to users.

3) Medicine Prescription Route: The Medicine Prescription route utilizes Flask's functionalities in conjunction with Selenium and Chrome WebDriver for dynamic content extraction. When a predicted disease is received, this route orchestrates the extraction process and delivers detailed medicine prescriptions, including dosages & uses, adverse effects, warnings and other necessary information.

The integration of frontend technologies for user interaction and Python Flask as the backend ensures a cohesive and efficient system architecture, allowing seamless communication between the frontend interface and backend functionalities. This architecture enables the Disease Prediction System to provide accurate predictions and comprehensive disease-related information to healthcare professionals and individuals.

V. Methodology and Experiments/Demo

The Disease Prediction System employs various methodologies and techniques to achieve accurate disease prediction and comprehensive information retrieval.

A. Predict Disease

The Predict Disease route within the backend utilizes Multinomial Naive Bayes, Decision Tree, and Random Forest machine learning algorithms. These models are trained on diverse symptom datasets to predict diseases based on received symptom inputs, showcasing exceptional performance with over 95% accuracy on test data, ensuring reliable and accurate disease predictions for users.

The experiments conducted encompassed the utilization of three distinct machine learning algorithms for disease prediction:

1) Using Multinomial Naive Bayes: The prediction of diseases using Multinomial Naive Bayes involved several key steps:

1) **Data Preprocessing**: Initially, the dataset underwent preprocessing by splitting it into features and target variables.
2) **Model Training**: The Multinomial Naive Bayes classifier was trained using the preprocessed training dataset containing symptom data.
3) **Prediction**: Predictions were made on the test dataset to evaluate the model's predictive capabilities.
4) **Evaluation**: The classifier's performance metrics, including accuracy, confusion matrix, and classification report, were thoroughly evaluated.

The results from this classifier exhibited promising accuracy rates, demonstrating its potential effectiveness in predicting diseases based on symptom data.

2) Using Decision Tree: The prediction of diseases using the Decision Tree classifier involved the following procedural steps:

1) **Data Preprocessing**: The dataset was segregated into features and target variables for model development.
2) **Model Training**: Training the Decision Tree classifier using the preprocessed training dataset that comprised symptom data.
3) **Prediction**: Predictions were generated on the test dataset to assess the model's predictive performance.
4) **Evaluation**: The classifier's performance metrics, including accuracy, confusion matrix, and classification report, were comprehensively analyzed.

The Decision Tree model demonstrated promising accuracy rates, indicating its potential for disease prediction based on symptom data.

3) Using Random Forest: The prediction of diseases using the Random Forest classifier encompassed the subsequent steps:

1) **Data Preprocessing**: Partitioning the dataset into features and target variables to prepare it for model training.
2) **Model Training**: Training the Random Forest classifier using the preprocessed training dataset that contained symptom data.
3) **Prediction**: Predicting disease classes for the test dataset to evaluate the model's predictive performance.
4) **Evaluation**: Analyzing the classifier's performance metrics, including accuracy, confusion matrix, and classification report, to assess its efficacy in disease prediction.

The experiments demonstrated high accuracy rates across all three classifiers, showcasing their effectiveness in accurately predicting diseases based on symptom data. These results underscore the potential applicability of these machine learning models in healthcare for disease prediction.

B. Get Disease Description

Leveraging the capabilities of Beautifulsoup, the Get Disease Description route implements web scraping techniques. Upon receiving a predicted disease, this route scrapes disease descriptions from a designated website, ensuring the retrieval of comprehensive and detailed information about the predicted disease. This information is subsequently delivered to users to enhance their understanding of the disease.

C. Medicine Prescription

The Medicine Prescription route utilizes Selenium in conjunction with Chrome WebDriver for dynamic content extraction. Upon receiving a predicted disease, this route orchestrates the extraction process from a specific website, providing detailed and dynamic medicine prescriptions. Users can access information about the dosages & uses, adverse effects, warnings, and other necessary details of prescribed medicines associated with the predicted disease.

These methodologies and routes employed within the Disease Prediction System combine machine learning algorithms, web scraping, and dynamic content extraction

techniques to ensure accurate disease prediction and the retrieval of detailed disease-related information. The successful integration of these techniques enhances the system's capability to assist healthcare professionals and individuals in making informed decisions regarding disease identification, treatment, and management.

VI. Results and Conclusion

The experiments conducted using various machine learning algorithms for disease prediction yielded compelling results:

A. Predict Disease using Multinomial Naive Bayes

The Multinomial Naive Bayes classifier achieved remarkable performance in disease prediction, exhibiting an accuracy of 95% on the test dataset. The average precision, recall, and F1-score across multiple disease classes were also high, with precision at 96%, recall at 94%, and F1-score at 95%. These metrics indicate the classifier's effectiveness in accurately predicting diseases based on symptom data, showcasing its potential application in healthcare systems.

B. Predict Disease using Decision Tree

The Decision Tree classifier demonstrated a commendable accuracy rate of 97.6% on the test dataset. The average precision, recall, and F1-score across various disease classes were consistent, with precision at 98%, recall at 97%, and F1-score at 97%. Despite minor variations in performance across disease classes, the Decision Tree model shows promising results for disease prediction, warranting its consideration for healthcare applications.

C. Predict Disease using Random Forest

The Random Forest classifier also showcased an accuracy rate matching the Decision Tree classifier, reaching 97.6% on the test dataset. Similarly, the average precision, recall, and F1-score across different disease classes were high, with precision at 97%, recall at 96%, and F1-score at 97%. This classifier's performance closely aligns with the Decision Tree model, indicating its effectiveness in accurate disease prediction based on symptom data.

D. Conclusion

In conclusion, the experiments conducted using Multinomial Naive Bayes, Decision Tree, and Random Forest classifiers for disease prediction yielded promising outcomes. These machine learning models demonstrated high accuracy rates along with strong average precision, recall, and F1-scores, showcasing their potential applicability in healthcare for disease diagnosis based on symptoms.

The integration of machine learning techniques in healthcare systems can significantly aid medical professionals in accurately predicting diseases, facilitating prompt and effective treatment. Further research and refinement of these models could enhance their performance, contributing to improved healthcare diagnostics and patient care.