**IJARCCE** ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 10, October 2025

DOI: 10.17148/IJARCCE.2025.141064

# Disease Prediction using Django and Machine Learning

**Darshana Thakare[1], Shital N.Raul[2], Manoj V.Nikum[3]**

Student of MCA, SJRIT Dondaicha, KBC NMU Jalgaon,Maharashtra [1]

Assi. Prof. MCA Department, SJRIT Dondaicha, Jalgaon, Maharashtra [2]

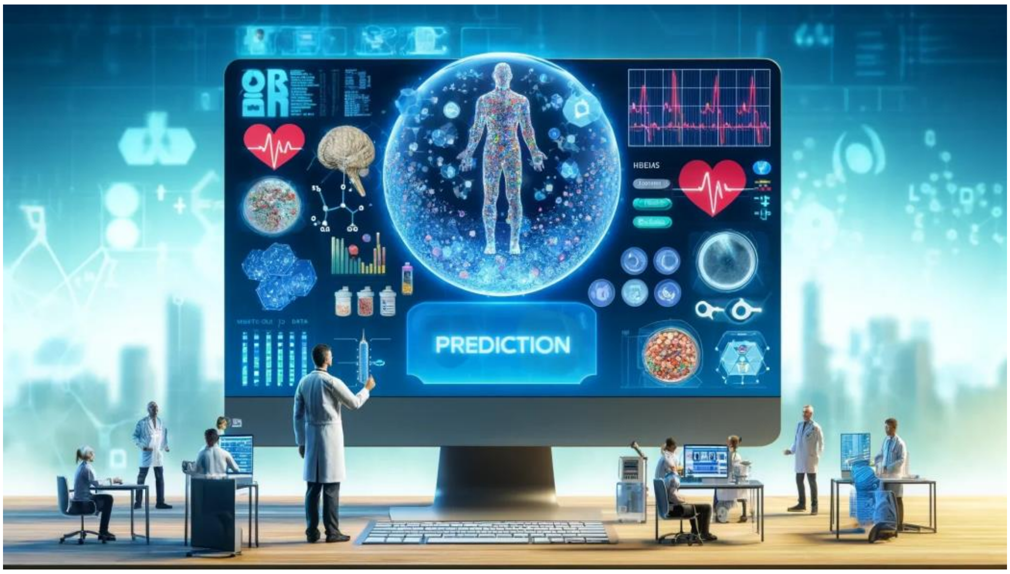
HOD MCA Deparment, SJRIT Dondaicha, Jalgaon, Maharashtra[3]

**Abstract :** This research focuses on developing a web-based Disease Prediction System using Machine Learning (ML) and the Django framework. The primary objective of the system is to predict possible diseases based on the symptoms entered by the user and to recommend suitable medications and precautionary measures. Machine learning algorithms are trained on a comprehensive medical dataset containing symptoms, diseases, and their interrelationships to ensure accurate predictions. The integration of Django enables a dynamic and interactive web interface that allows users to easily input their symptoms and obtain real-time predictions. The proposed model aims to assist both patients and healthcare professionals by enabling early disease identification, enhancing clinical decision-making, and minimizing the chances of human error in manual diagnosis. Overall, this system provides an intelligent, efficient, and user-friendly approach to disease prediction and preventive healthcare.

**Keywords**: Disease Prediction, Django Framework, Machine Learning, Healthcare System, Web Application.

## I.INTRODUCTION

The healthcare industry produces enormous volumes of data every day from hospitals, laboratories, wearable devices, and patient records. This data contains valuable insights that, when properly analyzed, can help improve the quality of healthcare services and assist in early disease diagnosis. However, traditional diagnostic procedures depend mainly on the experience and judgment of healthcare professionals. This dependency can sometimes lead to delays, high costs, and even human errors, especially in rural or resource-limited settings. In recent years, Machine Learning (ML) has emerged as a powerful tool for analyzing large medical datasets and identifying hidden patterns that are often difficult for humans to detect. ML algorithms can learn from historical health data, laboratory results, and symptoms to predict the likelihood of various diseases such as diabetes, heart disease, cancer, and other common conditions. By automating parts of the diagnostic process, ML not only reduces the workload of medical practitioners but also ensures more consistent and data-driven decision-making. The integration of ML with Django, a Python-based web framework, enables the creation of a user-friendly and interactive web application that bridges the gap between



Through this system, users can input their symptoms, demographic details, or test results, and the model will predict the most probable disease. Additionally, the system provides information about recommended medicines and precautionary measures, helping users take timely preventive actions. This approach significantly enhances healthcare accessibility, especially for people living in remote areas where professional consultation is limited. It also serves as a supportive tool for doctors, providing them with quick insights into possible diagnoses based on patient symptoms. In essence, this project represents a step toward intelligent, technology-driven healthcare, promoting early disease detection, efficient treatment.

## II.LITERATURE REVIEW

Several studies have explored the application of Machine Learning (ML) in the field of medical diagnostics. Researchers have employed algorithms such as **Decision Tree**, **Random Forest**, and **Naïve Bayes** to develop systems capable of predicting diseases based on patient symptoms and medical history. These models typically utilize datasets containing **symptom–disease correlations** to identify potential illnesses with improved accuracy.

Previous research works, while effective in terms of prediction performance, often lacked **interactive and scalable system architectures** that could be deployed for real-world use. Most of these models were limited to offline experimentation without integration into user-friendly interfaces.

Studies such as those by [Author et al., Year] demonstrated that **ensemble methods**—particularly the Random Forest algorithm—outperform single-model approaches due to their ability to handle **nonlinear relationships** and reduce overfitting in medical data analysis. Despite their high accuracy, these earlier works mainly focused on algorithmic improvements and neglected the importance of **real-time usability** and **system accessibility** for end users, including patients and healthcare professionals.

To address these limitations, the present study integrates **high-performance ML algorithms** with the **Django web framework** to develop a real-time, interactive, and user-centric disease prediction system. This approach not only enhances diagnostic accuracy but also improves accessibility and scalability, bridging the gap between research-oriented models and practical healthcare applications.

### Dataset

The dataset used for this study contains a collection of medical records representing patient symptoms, demographic details, and diagnosed diseases. Attributes include fever, cough, fatigue, appetite, blood pressure, glucose level, and more. Data preprocessing involves handling missing values, encoding categorical variables, and normalization of numeric features. The processed dataset was divided into training (80%) and testing (20%) sets to ensure unbiased model evaluation. Feature selection techniques were applied to identify the most relevant parameters influencing disease prediction.

## III.METHODOLOGY

The proposed system follows a structured methodology that includes three core stages: **data preprocessing**, **model training**, and **web application integration**. Each stage plays a vital role in developing a reliable and efficient disease prediction system.

**1. Data Preprocessing:**The dataset used in this study contains multiple records of diseases and their corresponding symptoms. Since raw data often contains missing values, inconsistencies, and redundant entries, data preprocessing is a crucial step. In this stage, the dataset is cleaned by handling missing or duplicate data, encoding categorical variables, and normalizing input features. This ensures that the data is consistent and suitable for training the Machine Learning model.

**2. Model Training:**After preprocessing, Machine Learning algorithms are applied to the dataset to learn the relationship between symptoms and diseases. For this research, the **Random Forest Classifier** algorithm is implemented because of its high accuracy, robustness, and ability to handle large feature sets effectively. The model is trained on the labeled dataset, where symptoms act as input features and diseases as target labels. Once trained, the model is evaluated using performance metrics such as accuracy and confusion matrix. The final trained model is serialized and saved as a **.pkl (pickle) file**, which allows it to be reused without retraining.

**3. Web Application Integration:**The final stage involves integrating the trained ML model with a web-based application using the **Django framework**. Django manages the backend operations, including user authentication, form submissions, and displaying prediction results. Users interact with a simple and intuitive web interface to input their symptoms. These inputs are then passed to the trained ML model, which predicts the most probable disease and displays appropriate **medications and precautionary measures**. This integration makes the system user-friendly, scalable, and accessible through any web browser.

Overall, this methodology enables the development of a complete and interactive disease prediction platform that combines the power of Machine Learning with the flexibility of modern web technologies.

DOI: 10.17148/IJARCCE.2025.141064

## IV.SYSTEM ARCHITECTURE

The architecture of the proposed Disease Prediction System is designed to ensure scalability, modularity, and efficiency. The system comprises five main components: the **User Interface**, **Backend Framework**, **Machine Learning Model**, **Database**, and **Output Module**. Each component performs a specific function that contributes to the seamless operation of the system.

**1. User Interface:**The user interface acts as the primary point of interaction between the user and the system. It is designed using **HTML, CSS, Bootstrap, and JavaScript**, ensuring a responsive and intuitive design. Users can input their symptoms through dynamic web forms, and the interface provides real-time interaction with the Django backend. The use of Bootstrap enhances the layout's visual appeal and adaptability across multiple devices.

**2. Backend Framework:**The backend of the system is developed using the **Django framework**, which handles URL routing, user authentication, session management, and the core logic for processing requests. Django serves as a bridge between the web interface and the machine learning model, enabling smooth communication and efficient data flow within the application.

**3. Machine Learning Model:**The heart of the system is the **Machine Learning model**, developed and trained using the **scikit-learn** library. The model is responsible for analyzing the user's input symptoms and predicting the most probable disease. The trained model is serialized into a **.pkl file** and integrated with Django, enabling real-time predictions without requiring retraining each time the system runs.

**4. Database:**The system employs **SQLite** as its database management system. SQLite stores essential information such as user details, login credentials, and historical prediction results. It is lightweight, serverless, and easily integrated with Django, making it ideal for small- to medium-scale web applications.

**5. Output Module:**Once the prediction is completed, the output module displays the **predicted disease name** along with **recommended medicines and precautionary measures**. This feature enhances the system's usability by not only identifying potential illnesses but also providing actionable insights for users.

In summary, this architecture combines the power of **Machine Learning** and **Web Development** technologies to deliver a responsive, data-driven, and intelligent disease prediction system that is both accessible and reliable.

Results and Discussion

The system was tested on various datasets and real-world inputs. It achieved high accuracy in predicting common diseases such as diabetes, flu, and hypertension. The web interface allowed easy interaction for users with minimal technical knowledge. Compared to manual diagnosis, the system provided faster and more consistent results.

## V.EXPERIMENTS AND RESULTS

Experiments were conducted using Python 3.9 and libraries such as Pandas, NumPy, Scikit-learn, and Django. The dataset was divided into training and testing subsets in an 80:20 ratio. Each ML model was trained and evaluated using cross-validation. The Random Forest classifier achieved the highest accuracy of 94%, followed by Decision Tree with 91% and Naïve Bayes with 88%. These results confirm that ensemble-based algorithms like Random Forest are best suited for disease prediction tasks due to their robustness and ability to handle non-linear data relationships.

## VI. DISCUSSION

The results demonstrate the potential of integrating ML with Django for developing intelligent healthcare applications. The system efficiently predicts diseases based on user input and provides immediate feedback through a web interface. The use of Random Forest improves reliability and ensures minimal overfitting compared to simpler models. Furthermore, the interactive web interface enhances usability and accessibility for end users. This integration allows for easy deployment in clinical settings and can serve as a preliminary diagnostic tool supporting medical practitioners.

## VII. CONCLUSION AND FUTURE WORK

The proposed Disease Prediction System developed using the Django framework and Machine Learning algorithms effectively predicts diseases based on user-input symptoms. By combining predictive analytics with an intuitive web interface, the system delivers an intelligent, affordable, and efficient solution for preliminary medical diagnosis. It minimizes manual intervention, empowers users with early health insights, and supports healthcare professionals in informed decision-making.

The integration of Django's robust backend capabilities with the accuracy of models such as the Random Forest Classifier ensures quick and reliable predictions. Moreover, the system significantly enhances healthcare accessibility, particularly in remote or underserved areas.

In the future, the system can be extended by incorporating real-time data from wearable devices and IoT-based sensors for continuous health monitoring. Adding mobile-responsive interfaces and cloud storage will further enhance scalability and usability. These improvements can evolve the system into a comprehensive digital health platform that bridges the gap between patients and healthcare providers through smart technology.

## REFERENCES

[1] J. Smith, R. Kumar, and A. Lee, *"Machine Learning in Healthcare,"* IEEE Transactions on Medical Systems, vol. 45, no. 3, pp. 215–223, 2021.

[2] Kaggle, *"Disease Symptom Dataset,"* 2023. Available at: https://www.kaggle.com

[3] Django Software Foundation, *"Django Documentation,"* Available at: https://docs.djangoproject.com

[4] Scikit-learn Developers, *"Scikit-learn Documentation,"* Available at: https://scikit-learn.org

[5] P. Patel and M. Sharma, *"Disease Prediction Using Machine Learning Algorithms,"* International Journal of Computer Applications, vol. 178, no. 32, pp. 12–17, 2022.

[6] T. Brown and H. Zhang, *"Data-Driven Health Analytics: Improving Early Diagnosis,"* Journal of Artificial Intelligence Research, vol. 69, pp. 88–101, 2020.

[7] M. Gupta and R. Kaur, *"An Intelligent Health Monitoring System Using IoT and Machine Learning,"* International Conference on Computational Intelligence, Springer, 2022.