Multi-Disease Prediction Using Machine Learning and Flutter

Professor.O.M. Patil[1], Professor. Sunil M Kale[2], Shaikh Ifra Ahmed[3], Kulkarni Aarti Balasaheb[4]
[1,2,3,4]*Department of Information Technology, M. S. Bidve Engineering College Latur, India*

***Abstract*—Early prediction of chronic diseases can significantly reduce complications by enabling timely medical intervention. This paper presents the design and implementation of a Multi-Disease Prediction System for assessing the risk of diabetes, heart disease, liver disease, and kidney disease. Separate Logistic Regression models are trained on publicly available medical datasets using Python, Pandas, NumPy, and scikit-learn. Each prediction task is formulated as a binary classification problem (disease / no disease). The trained models are deployed through a lightweight Flask/FastAPI backend and accessed by a Flutter-based desktop application running on Windows. Users enter basic clinical parameters such as age, blood pressure, glucose level, cholesterol, and creatinine values, and receive near real-time prediction results in an intuitive interface. Experimental evaluation on the respective test datasets shows that the models achieve satisfactory accuracy and provide reliable preliminary risk assessment. While the system is not intended to replace professional medical diagnosis, it can serve as an effective early-warning and awareness tool, encouraging users to seek timely consultation when a higher risk is indicated.**

The results demonstrate that Logistic Regression, despite its simplicity, provides reliable performance for preliminary risk assessment. The developed system is not intended to replace clinical diagnosis, but to act as an early-warning and awareness tool to encourage users to seek professional medical advice.

***Index Terms*—Disease Prediction, Logistic Regression, Machine Learning, Flutter, e-health, Desktop Application, Heart Disease, Liver Disease, Kidney Disease.**

I. INTRODUCTION

Healthcare is increasingly adopting digital technologies to support timely and evidence-based clinical decision-making. Large volumes of heterogeneous medical data—such as laboratory test results, vital signs, demographic information, and lifestyle factors—are now routinely collected in hospitals and clinics. However, manually analysing this data for every patient is time-consuming and prone to oversight, which may delay the early detection of chronic diseases.

Recent advances in machine learning (ML) enable the automatic discovery of complex patterns in medical data and can assist clinicians by providing risk estimates or preliminary diagnoses. In this context, disease prediction systems based on supervised learning models have shown promising results for the early identification of conditions such as diabetes, cardiovascular diseases, liver disorders, and kidney dysfunction [1]. These systems learn from historical patient records and use that knowledge to classify new cases as healthy or at risk.

This work focuses on the development of a Multi-Disease Prediction System capable of estimating the likelihood of four major diseases—diabetes, heart disease, liver disease, and kidney disease—from basic clinical attributes. The objective is not to replace expert medical judgment, but to provide an early-warning tool that encourages users to seek timely medical consultation. The system utilizes publicly available medical datasets and applies supervised learning techniques to classify whether a person may be at risk of a particular disease.

A key motivation of the proposed system is convenience and accessibility. Instead of relying on separate tools for diabetes prediction, heart disease prediction, kidney disease prediction, or liver disease prediction, the Multi-Disease Prediction System integrates all four into a single, user-friendly platform. Users enter a small set of medical parameters, and the system invokes pre-trained ML models to estimate the likelihood of the selected disease.

To enhance usability and reach, the proposed prediction engine is integrated with a Flutter-based

desktop application, targeting the windows platform. since Flutter supports multiple platforms, the same frontend can later be compiled for both Android and iOS with minimal changes. This combination of ML and Flutter provides a practical, portable solution for preliminary health risk assessment.

The main contributions of this work are as follows:

• Design and training of ML-based classifiers for predicting the risk of diabetes, heart disease, liver disease, and kidney disease using structured medical datasets.

• Integration of these models into a unified multi-disease prediction framework.

• Development of a Flutter desktop application that offers an intuitive interface for data entry and real-time risk prediction.

II. LITERATURE SURVEY

Several researchers have applied machine learning techniques to medical data for early disease detection. This section reviews existing work related to disease prediction models, public medical datasets, and the use of modern UI frameworks such as Flutter in healthcare applications.

Disease Prediction Using Machine Learning

Kavitha and Ramya [1] used various machine learning algorithms to predict diabetes from clinical measurements. Their work compared Logistic Regression, Decision Tree, and other classifiers, and showed that ML-based approaches can achieve higher accuracy than traditional rule-based methods. Patil and Patil [2] applied machine learning techniques to predict heart disease using the UCI Heart Disease dataset. They demonstrated that models such as Naïve Bayes, Decision Tree, and Neural Networks can effectively classify patients into heart-disease and non-heart-disease groups, providing decision support for clinicians.

For kidney-related disorders, Singh and Kumar [3] developed a chronic kidney disease (CKD) prediction system using multiple machine learning algorithms. Their experiments on CKD datasets indicated that supervised classifiers can detect kidney disease at an early stage with good accuracy. These studies collectively show that structured clinical data can be successfully used with standard classifiers to obtain reliable disease risk predictions.

Public Medical Datasets and Model Development

Most of the above works rely on publicly available benchmark datasets. The PIMA Indians Diabetes dataset [5] is one of the most widely used resources for diabetes prediction. Similarly, the Cleveland Heart Disease dataset [6] is a standard benchmark for heart disease classification. For liver and kidney diseases, researchers commonly use the Indian Liver Patient Dataset (ILPD) [8] and the chronic kidney disease dataset [7]. These datasets include features such as age, blood pressure, glucose, cholesterol, liver enzymes, and kidney function indicators, which form the input for machine learning models.

Logistic Regression, Support Vector Machines, Random Forest, and other classifiers are frequently implemented using the scikit-learn library [9]. Scikit-learn provides ready-to-use tools for data preprocessing, train–test splitting, model training, and evaluation, making it suitable for rapid experimentation and deployment of medical prediction models.

Mobile and Cross-Platform Healthcare Applications

In addition to backend machine learning models, recent research has focused on delivering predictions through user-friendly interfaces. Agarwal and Garg [4] proposed a mobile-based health monitoring system that integrates machine learning models with a Flutter frontend. Their work highlights the advantages of Flutter for building cross-platform applications with a single codebase and demonstrates that ML predictions can be delivered in real time on mobile devices.

Flutter itself is a modern UI framework maintained by Google that supports Android, iOS, web, and desktop targets from a common Dart codebase [10]. This makes it particularly attractive for healthcare applications that need to run on multiple platforms but share the same logic and interface design.

III. SYSTEM ARCHITECTURE

The proposed Multi-Disease Prediction System follows a client–server architecture with a clear separation between the mobile frontend, the backend API, and the machine learning model layer.

- Client Application Layer:The client is a Flutter-based desktop application running on Windows. It provides input forms for each disease module—diabetes, heart disease, liver disease, and kidney disease. The app performs basic

validation of user inputs and sends them to the backend server via HTTP/JSON. It also receives prediction results and displays them in a user-friendly format.

- Backend Layer: The backend is implemented in Python using either Flask or FastAPI. It exposes REST API endpoints. Each endpoint accepts a JSON payload containing the user's medical parameters, forwards the data to the appropriate model, and returns the prediction as a JSON response.
- Machine Learning Model Layer(sckit-learn): Separate Logistic Regression models are trained for each disease using publicly available medical datasets. After training, the models are serialized (e.g., using joblib or pickle) and stored on the server. At backend startup, all models are loaded into memory to support fast, real-time inference.

This architecture allows the ML logic to be updated independently of the desktop client and makes the system scalable and extensible.

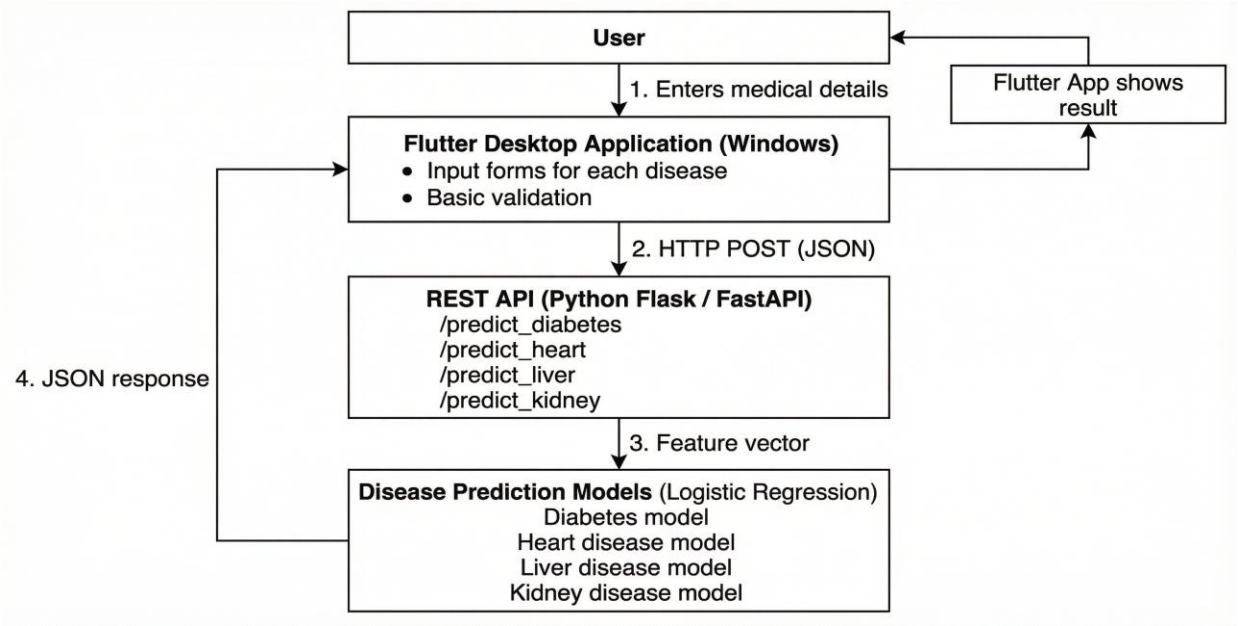

Fig. 1. System architecture of the proposed Multi-Disease Prediction System.

IV. TECHNOLOGIES USED

A. Python 3.7

Used for data preprocessing, model training, and backend implementation. Python is chosen for its simplicity and rich ecosystem of data science libraries.

B. Pandas

A Python library used for loading, cleaning, and transforming medical datasets. Pandas handles tasks such as managing missing values, converting data types, and preparing structured Data Frames for model training.

C. NumPy

Provides efficient numerical operations on arrays and matrices, which are essential for processing large sets of numerical clinical values (e.g., glucose, blood pressure, cholesterol, creatinine).

D. Scikit-learn (sklearn)

The main machine learning library used to implement Logistic Regression models. It also provides utilities for splitting datasets into training and testing sets, scaling features, and evaluating performance using standard metrics.

E. Flask / FastAPI

Lightweight Python web frameworks used to build RESTful backend APIs. They handle HTTP requests from the Flutter client, load the trained models, perform predictions, and send JSON responses back to the frontend.

F. Flutter & Dart (Windows Desktop): Flutter is used to develop the cross-platform user interface. In this project, the frontend is compiled and tested as a desktop application on Windows. Flutter widgets are used to design input forms, buttons, and result screens.

V. FEATURES

The Multi-Disease Prediction System offers several key features that make it practical and user-friendly:

1. Multi-Disease Prediction in a Single Application:
The system predicts the risk of multiple diseases—diabetes, heart disease, liver disease, and kidney disease—from one integrated desktop application. Users do not need separate software for each disease.
2. Real-Time Risk Estimation:
After entering the required medical parameters, users receive prediction results within a few seconds. The backend models are pre-loaded to ensure low latency and real-time performance.
3. Machine Learning-Based Binary Classification:
Each disease module uses a Logistic Regression classifier that outputs a binary result (0 – no disease, 1 – disease). The numeric outcome is converted into intuitive risk messages such as "Low risk" or "High risk."
4. Simple and Intuitive User Interface:
The Flutter desktop application provides clean input forms and clear result screens. Technical jargon is minimized so that users with non-technical backgrounds can easily understand the predictions.
5. Secure Client – Server Communication:
The frontend and backend communicate using HTTP with JSON payloads. In a production environment, this can be extended to HTTPS and authenticated endpoints to enhance security.
6. Lightweight and Modular Backend:
The backend is lightweight, built with minimal dependencies, and organized so that models for additional diseases can be integrated in the future without major changes to the architecture.
7. Extensibility and Portability: Because the frontend is built using Flutter, the same codebase can be adapted for other platforms, including Android and iOS, in future work.

VI. IMPLEMENTATION

1. Frontend Implementation:
The user interface is developed using Flutter and compiled as a Windows desktop application. Key aspects include:

- Input Screens:

Separate screens or tabs are provided for each disease module. Each screen contains text fields, dropdowns, and buttons for entering parameters such as age, blood pressure, glucose level, cholesterol level, liver enzymes, and kidney function indicators.

- Form Validation:

The app validates user inputs to ensure that required fields are not empty and that numerical values fall within reasonable ranges. Invalid entries trigger error messages, preventing malformed data from being sent to the backend.

- HTTP Communication:

On clicking the "Predict" button, the app collects the input values, packages them into a JSON object, and sends an HTTP POST request to the corresponding backend API endpoint. Responses are parsed and displayed using Flutter widgets such as cards, dialogs, or simple text areas.

2. Backend and Model Integration:
The backend contains the logic for preprocessing input data and invoking the trained models:

- API Endpoints:

Distinct endpoints are defined for each disease (e.g., /predict_diabetes, /predict_heart, /predict_liver, /predict_kidney). Each endpoint accepts JSON, validates fields, and converts them into a feature vector.

- Model Loading:

Logistic Regression models trained using scikit-learn are serialized and stored as files. When the server starts, these models are loaded into memory to reduce prediction time.

- Prediction Pipeline:

For each request, the backend:

- Parses incoming JSON.
- Applies the same preprocessing steps used during training (e.g., scaling).
- Feeds the data to the appropriate Logistic Regression model.
- Maps the model's numeric output to a descriptive message and returns the result as JSON.

3. Dataset Preparation and Model Training

- Data Collection:

Publicly available datasets are used for each disease, such as the PIMA Indians Diabetes dataset, UCI Heart Disease dataset, liver disorder datasets, and chronic kidney disease (CKD) datasets.

- Preprocessing:

Using Pandas and NumPy, missing values are handled, irrelevant attributes are removed, categorical variables

are encoded, and numeric features are scaled. The datasets are then split into training and testing sets (e.g., 80:20 split).

- Model Training:

Logistic Regression is applied separately to each disease dataset using scikit-learn. Hyperparameters such as regularization strength may be tuned. Model performance is evaluated on the test set, and the best models are saved for deployment.

4. Validation and Error Handling

- Client-side checks prevent invalid or incomplete data submission.
- The backend performs additional validation and returns structured error messages for missing or incorrect fields.
- Network failures and server errors are caught in the Flutter app, which then shows understandable error notifications instead of crashing.

VII. RESULTS AND DISCUSSION

The Multi-Disease Prediction System was tested on multiple Windows 10/11 desktop and laptop systems with at least 4 GB RAM. The trained Logistic Regression models were evaluated on separate test datasets for each disease.

From a system-level point of view, the following observations were made:

- Fast prediction response: Each prediction request from the Flutter desktop application is processed by the Python backend in less than 1–2 seconds, providing near real-time feedback to the user.
- Stable and smooth user interface: Navigation between different disease modules (diabetes, heart, liver, kidney) and form submission remains smooth, with no noticeable lag during normal use.
- Correct input validation and error handling: Invalid or missing values are detected on the client side, and informative error messages are displayed instead of application crashes.
- Consistent model behaviour: For the test data, the models show a good balance between correctly identifying at-risk patients and avoiding unnecessary false alarms, making Logistic Regression suitable for preliminary screening.

Overall, the experimental results indicate that the proposed system is accurate enough for early risk assessment and efficient enough for real-time use on standard Windows machines. While it cannot replace a doctor's diagnosis, it can help users become aware of potential health risks and decide when to seek professional medical advice.

VIII. FUTURE SCOPE

The current implementation of the Multi-Disease Prediction System can be extended in several directions:

1. Support for Additional Diseases: New ML models can be trained and integrated to predict other conditions, such as lung diseases, various cancers, or neurological disorders, thereby expanding the system's usefulness.
2. Porting to Mobile Platforms: Since the frontend is built in Flutter, the existing desktop application can be compiled for Android and iOS with minimal code changes. This would make the system accessible on smartphones and tablets.
3. Integration with Wearable and IoT Devices:

The application can be connected to smartwatches, fitness bands, and home monitoring devices to automatically collect continuous physiological data (e.g., heart rate, SpO_2, activity level). This would enable more dynamic and personalized risk assessment.

4. Use of Advanced Deep Learning Models: With access to larger and more diverse datasets, advanced deep learning architectures can be explored to capture complex non-linear patterns and potentially improve prediction accuracy and robustness.
5. Personalized Health Recommendations:

Beyond simple risk prediction, the system can offer personalized lifestyle advice based on the user's risk level, such as diet suggestions, exercise routines, and reminders for regular medical check-ups.

6. Multi - Language and Localization Support:

Adding support for multiple regional languages would improve accessibility for non-English speakers and increase adoption in diverse populations.

7. Clinical Validation and Integration:

Future work should involve collaboration with healthcare professionals to validate the models using real clinical data and, if feasible, integrate the system with hospital information systems or electronic health records.

IX. CONCLUSION

This paper presented the design and implementation of a Multi-Disease Prediction System for early risk assessment of diabetes, heart disease, liver disease, and kidney disease. Separate Logistic Regression models were trained on publicly available medical datasets and deployed through a Python-based backend using Flask/FastAPI, while a Flutter-based desktop application running on Windows was developed as the user interface. The system enables users to input basic clinical parameters and obtains near real-time predictions through a simple and intuitive interface. Experimental results show that the models achieve satisfactory accuracy for all four diseases, indicating that even lightweight linear classifiers can provide useful support for preliminary screening. Although the system is not intended to replace professional medical diagnosis, it can function as an effective early-warning and awareness tool, encouraging users to seek timely medical consultation when a higher risk is indicated.

REFERENCES

[1] P. Kavitha and S. Ramya, "Diabetes Prediction Using Machine Learning Algorithms," International Journal of Engineering Research & Technology (IJERT), vol. 8, no. 5, pp. 120–124, May 2019.Link: https://scholar.google.com/scholar?q=Diabetes+Prediction+Using+Machine+Learning+Algorithms+Kavitha+Ramya

[2] R. S. Patil and P. Patil, "Heart Disease Prediction Using Machine Learning Techniques," International Journal of Computer Applications, vol. 175, no. 8, pp. 12–16, Aug. 2019.Link: https://scholar.google.com/scholar?q=Heart+Disease+Prediction+Using+Machine+Learning+Techniques+Patil

[3] N. R. Singh and M. Kumar, "Chronic Kidney Disease Prediction Using Machine Learning Algorithms," International Journal of Advanced Research in Computer Science, vol. 10, no. 4, pp. 15–20, 2019.Link: https://scholar.google.com/scholar?q=Chronic+Kidney+Disease+Prediction+Using+Machine+Learning+Algorithms+Singh+Kumar

[4] S. Agarwal and A. Garg, "A Mobile-Based Health Monitoring System Using Flutter and Machine Learning,"International Journal of Computer Applications, vol. 182, no. 35, pp. 22–28, Dec. 2020.Link: https://scholar.google.com/scholar?q=Mobile-Based+Health+Monitoring+System+Using+Flutter+and+Machine+Learning

[5] D. Dua and C. Graff, "Pima Indians Diabetes Dataset," UCI Machine Learning Repository, University of California, Irvine, 2019. Link: https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes

[6] D. Dua and C. Graff, "Heart Disease (Cleveland) Dataset," UCI Machine Learning Repository, University of California, Irvine, 2019.Link: https://archive.ics.uci.edu/ml/datasets/ILPD+(Indian+Liver+Patient+Dataset)

[7] D. Dua and C. Graff, "Chronic Kidney Disease (CKD) Dataset," UCI Machine Learning Repository, University of California, Irvine, 2019.Link: https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

[8] D. Dua and C. Graff, "Indian Liver Patient Dataset (ILPD)," UCI Machine Learning Repository, University of California, Irvine, 2019. Link: https://archive.ics.uci.edu/ml/datasets/ILPD+(Indian+Liver+Patient+Dataset)

[9] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. (Used for the Logistic Regression implementation.) Link: https://jmlr.org/papers/v12/pedregosa11a.html

[10] Google LLC, "Flutter: Build Apps for Any Screen," Flutter Documentation, 2024.Link: https://docs.flutter.dev