

Project Title

Multiple Disease Prediction

Objective

To build a scalable and accurate system that:

- Assists in early detection of diseases.
 - Improves decision-making for healthcare providers.
 - Reduces diagnostic time and cost by providing quick predictions.
-

System Architecture

1. **Frontend:** User interface for inputting data (e.g., symptoms, test results).
 - Technologies: Streamlit
 2. **Backend:** Processes user inputs and interacts with the prediction model.
 - Framework: Python
 3. **Machine Learning Models:** Predict diseases based on input data.
 - Algorithms: Logistic Regression, Random Forest, XGBoost, etc.
-

Features

1. **Multi-disease Prediction:** Predicts the likelihood of multiple diseases (e.g., Kidney disease, liver disease, Parkinsons).
2. **User-friendly Interface:** Simplified input forms and clear prediction results.
3. **Interactive Visualizations:** Graphs and charts to explain predictions.
4. **Secure Data Handling:** Ensures user privacy and compliance with data protection regulations.
5. **Scalable System:** Supports a large number of users and diseases.

Workflow

1. **Input Data:**
 - User enters symptoms, demographic details, and test results.
2. **Data Preprocessing:**
 - Handles missing values, encodes categorical data, and scales numerical features.
3. **Model Inference:**
 - Predictive models process the input data and provide probabilities for each disease.
4. **Output:**
 - Displays predicted diseases with their respective probabilities and risk levels

Implementation

Data Collection

- **Sources:** DataSet: 1. parkinsons
- 2. kidney_disease
- 3. indian_liver_patient
- **Features:** Symptoms, test results, demographic data.

Data Preprocessing

- Handling missing data (e.g., imputation).
- Encoding categorical variables.
- Feature scaling (e.g., MinMaxScaler, StandardScaler).

Model Training

- Train separate models for each disease or a multi-output classifier.
- Cross-validation to ensure robustness.

Model Evaluation

- Metrics: Accuracy, Precision, Recall, F1-score, ROC-AUC.
- Confusion Matrix for detailed analysis.

Deployment: (Optional)

- Deploy on platforms like AWS(**Optional**)
-

Tools and Technologies

- **Programming Language:** Python
 - **Libraries:** Scikit-learn, Pandas, NumPy, Matplotlib, Seaborn
 - **Frontend:** Streamlit
-

Results

The Multiple Disease Prediction System is a promising tool for enhancing healthcare accessibility and efficiency. By integrating advanced machine learning techniques with user-friendly interfaces, the system bridges the gap between technology and healthcare. Continuous improvements in data quality and model sophistication can make this tool indispensable for early disease detection.

Project Evaluation Metrics

- **Regression:**
 - Mean Absolute Error (MAE).
 - Root Mean Squared Error (RMSE).
 - **Classification:**
 - Accuracy.
 - Precision, Recall, and F1-Score.
 - **Application:**
 - Responsiveness and usability of the Streamlit app.
 - Quality of visualizations and insights provided.
-

Technical Tags

- **Streamlit, Python, Machine Learning, Visualization.**
-

Project Deliverables

1. **Source Code:** Python scripts for preprocessing, modeling, and Streamlit app.
 2. **Streamlit Application:** Fully functional web application deployed locally or on the web.
 3. **Documentation:**
 - Explanation of models and methodologies.
 - Instructions for running and deploying the app.
 4. **Presentation:** Summary of results and business insights.
-

Project Guidelines

1. Follow Python coding standards.
 2. Use version control (GitHub/GitLab) for code management.
 3. Ensure modularity (separate data, models, and app logic).
 4. Optimize the Streamlit app for performance (e.g., caching results where needed).
 5. Regularly validate models and app functionality.
-

