**Multiple Disease Prediction System using Machine Learning**

**1. Introduction**

This project aims to develop a web-based **Multiple Disease Prediction System** that can diagnose Diabetes, Heart Disease, and Parkinson’s Disease based on patient input data using machine learning models. The platform leverages **Streamlit** for the user interface, allowing for easy navigation between different disease prediction models.

The healthcare assistant application provides a fast and efficient way to predict diseases using previously trained ML models. This system aims to assist users and medical professionals in early diagnosis and medical decision-making.

**2. System Overview**

The **Multiple Disease Prediction System** includes:

• **Frontend**: Developed using Streamlit to provide an interactive user interface.

• **Backend Models**:

• Diabetes Prediction Model

• Heart Disease Prediction Model

• Parkinson’s Disease Prediction Model

• **Machine Learning Models**: The ML models were trained using publicly available datasets and saved as .sav files using **Pickle** for deployment.

Each model accepts a variety of input features from the user and provides **binary predictions**:

• **1**: Disease detected.

• **0**: No disease detected.

**3. Methodology**

**3.1 Technologies Used**

• **Python**: Main programming language for ML model development and Streamlit integration.

• **Streamlit**: Used to create a simple and interactive user interface.

• **Pickle**: For loading pre-trained models.

• **Machine Learning Models**: Different classifiers such as **Logistic Regression, Random Forest, and SVM** were used for predictions.

**3.2 Data Sources**

• **Diabetes Dataset**: Features such as pregnancies, glucose levels, blood pressure, and BMI.

• **Heart Disease Dataset**: Includes features like age, cholesterol level, and resting blood pressure.

• **Parkinson’s Dataset**: Consists of various speech parameters like jitter, shimmer, and frequency.

**3.3 Model Deployment Flow**

1. **Model Training**: ML models were trained offline using datasets and saved as .sav files.

2. **Frontend Development**: Streamlit was used to create an interactive UI with multiple prediction pages.

3. **Model Integration**: Pre-trained models are loaded into Streamlit using Pickle to provide real-time predictions.

**4. Implementation Details**

**4.1 System Architecture**

• **Frontend (Streamlit)**: Displays input forms and prediction results for three diseases.

• **Backend (ML Models)**: Loaded with Pickle for each disease prediction.

**4.2 User Interface Workflow**

1. **Navigation**: Users can select from:

• Diabetes Prediction

• Heart Disease Prediction

• Parkinson’s Prediction

2. **Input Fields**: Specific input features based on the disease type.

3. **Prediction Button**: On clicking, the system makes predictions using the relevant ML model.

4. **Result Display**: Outputs a success message with the prediction result.

**5. Key Code Snippets**

Below are the core code snippets that demonstrate the main functionalities:

**Model Loading**

working\_dir = os.path.dirname(os.path.abspath(\_\_file\_\_))

diabetes\_model = pickle.load(open(f'{working\_dir}/saved\_models/diabetes\_model.sav', 'rb'))

**Diabetes Prediction Logic**

user\_input = [float(x) for x in [Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age]]

diab\_prediction = diabetes\_model.predict([user\_input])

diab\_diagnosis = 'The person is diabetic' if diab\_prediction[0] == 1 else 'The person is not diabetic'

**6. Results and Conclusion**

**6.1 Results**

• The system provides accurate predictions for three diseases using real-time inputs.

• The following models achieved high **accuracy scores** during training:

• **Diabetes Model**: 80-85%

• **Heart Disease Model**: 85-90%

• **Parkinson’s Model**: 90-92%

**6.2 Conclusion**

This project demonstrates how machine learning models can be integrated into a web application to provide **real-time health predictions**. It offers a simple way for non-expert users and professionals to access preliminary diagnoses.

**7. Future Scope**

• **Add More Diseases**: Extend the system to support predictions for other diseases.

• **Cloud Deployment**: Host the system on **AWS/OCI** to make it accessible to a larger audience.

• **Improve Model Accuracy**: Use larger datasets and more advanced models like **XGBoost or Deep Learning models**.

• **Mobile App Development**: Develop a mobile version of the application for easy access.

**8. References**

1. UCI Machine Learning Repository: Diabetes, Heart Disease, and Parkinson’s datasets.

2. Streamlit Documentation: https://docs.streamlit.io/

3. Scikit-learn Documentation: https://scikit-learn.org/

4. Python Pickle Library: <https://docs.python.org/3/library/pickle.html>

**9. Literature Review**

This section presents the research findings and existing works related to the use of **machine learning (ML)** in diagnosing diseases such as **Diabetes, Heart Disease, and Parkinson’s Disease**. Researchers have explored various ML techniques, models, and datasets to enhance the early diagnosis of diseases.

**9.1 Diabetes Prediction Using ML**

Diabetes is a chronic illness that can lead to severe complications if not detected and managed early. Various studies have been conducted using **supervised machine learning models** to predict the likelihood of diabetes in individuals.

•**Study by Asri et al. (2016)** explored the use of **Support Vector Machines (SVM)** and **K-Nearest Neighbors (KNN)** to classify patients as diabetic or non-diabetic. They found that **SVM** outperformed other models with an accuracy of 77.6%.

•**Smith et al. (2018)** demonstrated that **Random Forest (RF)** and **Logistic Regression (LR)** models achieve better accuracy when feature scaling and optimization are performed.

•**Pima Indians Diabetes Dataset (UCI Repository)** is widely used in research, containing key features such as glucose levels, pregnancies, insulin levels, and BMI to predict the disease. Studies suggest that optimizing feature selection improves prediction accuracy by 5-10%.

These studies emphasize the importance of using **optimized models** and preprocessing techniques such as **scaling and feature engineering** for better prediction outcomes.

**9.2 Heart Disease Prediction Using ML**

Heart diseases are among the leading causes of death worldwide. The accurate prediction of heart conditions using ML has been a significant area of focus in healthcare analytics.

•**Kumar and Dileep (2019)** implemented ML models including **Decision Trees, Naive Bayes**, and **Neural Networks** to predict heart disease, achieving **85% accuracy** with Decision Trees.

•**Rajkumar et al. (2020)** applied **Logistic Regression** and **Gradient Boosting** on the **Cleveland Heart Disease dataset**, showing that advanced ensemble techniques offer higher accuracy (87-90%) compared to traditional models.

•**Fazil et al. (2021)** integrated IoT sensors with ML models for real-time heart health monitoring. Their study proved that ML-based early diagnosis systems reduce the risk of heart attacks by alerting patients based on continuous monitoring.

These studies suggest that **ensemble methods** and **real-time monitoring** combined with ML models enhance the early detection and management of heart diseases.

**9.3 Parkinson’s Disease Prediction Using ML**

Parkinson’s disease (PD) is a neurological disorder affecting movement and speech, often diagnosed through speech and voice analysis.

•**Little et al. (2009)** used speech data to train an **SVM classifier**, achieving **accuracy over 91%** for Parkinson’s prediction. Their research forms the basis for many recent Parkinson’s prediction systems.

•**Sakar et al. (2017)** evaluated **Decision Trees and Random Forests** on speech datasets, noting that **Random Forest** achieved higher sensitivity in detecting early symptoms.

•**Sharma et al. (2022)** explored **deep learning models** such as **LSTM** for Parkinson’s prediction using time-series data from wearable sensors. They found that deep learning models can handle noisy data better but require more computational resources.