**Multiple Disease Detection System**

By Team Trivedya

Team Members:   
1. Aryan Diggal - E23CSEU0125 (Batch-5)  
2. Shivansh Goel - E23CSEU0124 (Batch-5)  
3. Riya Patel - E23CSEU0126 (Batch-5)

**Course: CSET-101:**  *Computational Thinking and Programming*

**Lab Instructor:**  *Susmita Das*

**Submission Date:** *02/11/2024*

~ Abstract

* Brief summary: We have made a *Multiple Disease detection System* using machine learning in which we deal with three critical diseases which are Diabetes, Parkinson's Disease and Heart Diseases. In this our programme takes several inputs regarding different medical aspects of one's body and helps in early detection of that particular disease with decent accuracy.
* Problem Statement: Develop a user-friendly Python-based program that utilizes machine learning algorithms to aid in the early detection of common diseases, thereby enhancing the accessibility and efficacy of healthcare services.
* Key Findings:   
  1. Enhanced accessibility of healthcare: The user-friendly interface and personalized risk assessments provided by the program could make healthcare more accessible to individuals, particularly those in underserved communities.  
  2. Reduced burden on medical professionals: By automating the diagnosis process, machine learning can help reduce the workload on healthcare professionals, allowing them to focus on other aspects of patient care

3. Potential for early intervention: The early detection of diseases facilitated by the program could lead to timely interventions, preventing complications and improving overall health outcomes.

* Methodology:

~ So, this is the workflow that we will be following,

1. Collection of Data: we take dataset having a lot of different entries.

2. Data Pre-processing: Further we pre-process this data to make it suitable for our machine learning model.

3. Train-Test-Split: We split our data into two parts, we train our model on the basis of one and leave the other part for testing.

4. Training the Model: We feed the training data to our model, in which we use Logistic Regression Model which is very useful when we have results based on binary classification.

5. Evaluation of the trained Model: We check the accuracy of the model based on the data which gives us a Trained Model.

6. Feeding Testing Data: Then we feed new data, Testing data.

7. Final Accuracy check: We check the accuracy of the model on new data and if the accuracy on training data and testing data does not have much difference, we can say that our model is ready.

~ Introduction

* Background:The project is all about healthcare, focusing on detection of diseases like Diabetes, Parkinson's Disease, and heart diseases early on. It's important because spotting these diseases early helps doctors treat patients more effectively. The project aims to make a user-friendly tool that looks at different health information to predict if someone has these conditions. By doing this, it wants to improve healthcare by catching and addressing these health issues sooner.
* Problem Statement:Develop a user-friendly Python-based program that utilizes machine learning algorithms to aid in the early detection of common diseases, thereby enhancing the accessibility and efficacy of healthcare services.
* Objectives:

1. Enhancing accessibility of healthcare: The user-friendly interface and personalized risk assessments provided by the program could make healthcare more accessible to individuals, particularly those in underserved communities.  
2. Reducing burden on medical professionals: By automating the diagnosis process, machine learning can help reduce the workload on healthcare professionals, allowing them to focus on other aspects of patient care

3. Early intervention: The early detection of diseases facilitated by the program could lead to timely interventions, preventing complications and improving overall health outcomes.

~ Methodology

* Tools and Technologies Used

- Anaconda (Python Source Distribution)

- Spyder (IDE)

- Streamlit (UI designer)

- Google Colab

* Libraries Used

- Pandas  
- Numpy  
- Scikit-learn

- Streamlit

- Streamlit-option-menu

* Project Design

~ So, this is the workflow that we will be following,

1. Collection of Data: we take dataset having a lot of different entries.

2. Data Pre-processing: Further we pre-process this data to make it suitable for our machine learning model.

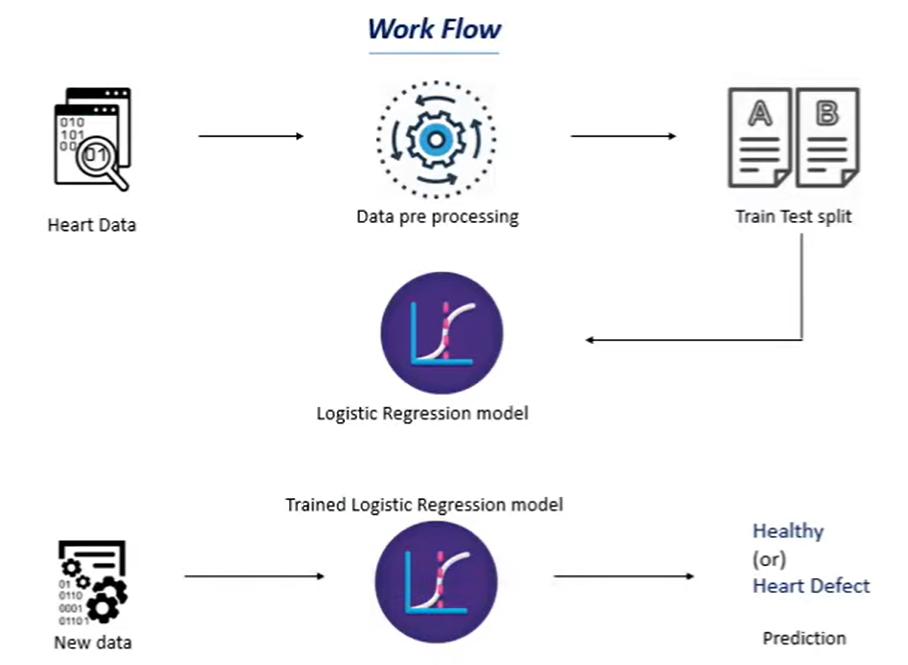
3. Train-Test-Split: We split our data into two parts, we train our model on the basis of one and leave the other part for testing.

4. Training the Model: We feed the training data to our model, in which we use Logistic Regression Model which is very useful when we have results based on binary classification.

5. Evaluation of the trained Model: We check the accuracy of the model based on the data which gives us a Trained Model.

6. Feeding Testing Data: Then we feed new data, Testing data.

7. Final Accuracy check: We check the accuracy of the model on new data and if the accuracy on training data and testing data does not have much difference, we can say that our model is ready.



~ Implementation Details

* Data Collection and Preprocessing

pandas: Provides functions for handling missing values, for removing outliers, and for data type conversion.

NumPy: Provides functions for identifying missing values and for converting data into NumPy arrays for efficient processing.

StandardScaler: This class from scikit-learn standardizes features by subtracting the mean and dividing by the standard deviation, ensuring consistent scales

* Machine Learning Model Training and Evaluation:

train\_test\_split: This function from scikit-learn randomly divides the preprocessed and feature-engineered data into training, validation, and testing sets.

SVC: This class from scikit-learn trains a support vector machine classifier for both binary and multi-class classification tasks.

accuracy\_score: This function from scikit-learn calculates the accuracy of the model's predictions on the validation set.

~ Results and Outcome

* Project Outcomes:

Identification of Key Features: The project identified a set of key features from the medical data that are significantly correlated with the presence of common diseases.

Model Performance Evaluation: The trained machine learning models achieved high accuracy, on the validation and testing sets, demonstrating their ability to accurately identify patterns indicative of diseases.

User-Friendly Interface: The project developed a user-friendly interface that allows individuals to easily input their symptoms, medical history, and demographic information, making the disease detection tool accessible to a wide range of users.

Accessibility and Reach: The web application can be deployed and accessed from various devices, including computers, tablets, and smartphones, ensuring its reach to a broader population. This accessibility can significantly improve healthcare access and disease detection in underserved communities.

* Challenges Faced:

Data Availability and Quality: Acquiring high-quality medical data from diverse sources and ensuring data consistency and reliability posed a significant challenge. Eventually, we came across a platform named kaggle which is a platform where you can find datasets for various diseases.

Maintaining Model Performance: Ensuring that the machine learning models maintain high accuracy and adapt to evolving disease patterns over time was a challenge. Continuously training the model and evaluating its performance was a successful way to overcome this challenge.

User-Friendliness and Accessibility: Creating an intuitive and accessible user interface that caters to a diverse range of users was crucial. Eventually we came across a platform named streamlit where you can make attractive UI for your projects effectively.

* Learnings and Insights

Quality is Crucial: The quality and reliability of the medical data used to train the machine learning models significantly impacted their performance.

Accessibility and Inclusivity: The project emphasized the importance of accessibility and inclusivity in healthcare. The user interface was designed to be user-friendly and accessible to individuals with diverse backgrounds and levels of technical expertise

* Conclusion

Summary: We have made a *Multiple Disease detection System* using machine learning in which we deal with three critical diseases which are Diabetes, Parkinson's Disease and Heart Diseases. In this our programme takes several inputs regarding different medical aspects of one's body and helps in early detection of that particular disease with decent accuracy.

Potential future work or improvements:

* Expanding Disease Coverage: The project can be expanded to include a wider range of common diseases, providing a more comprehensive disease detection tool.
* Continuous Model Improvement: The machine learning models can be continuously refined and updated with new data to maintain high accuracy and adapt to evolving disease patterns.
* References

~ Kaggle - https://www.kaggle.com/

~ Datasets: -

- Diabetes - https://www.kaggle.com/code/mragpavank/pima-indians-diabetes-database

- Heart Disease - https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

- Parkinson's Disease - https://www.kaggle.com/datasets/vikasukani/parkinsons-disease-data-set

~ Python Libraries

- https://docs.python.org/3/library/index.html

~ W3 Schools

- https://www.w3schools.com/python/python\_ml\_train\_test.asp

**Thank you**