

AI-Powered Disease Predictor

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning
with

TechSaksham – A joint CSR initiative of Microsoft & SAP

By

Anand Kumar

anandkashyap6048@gmail.com

Under the Guidance of

Pavan Sumohana

ACKNOWLEDGEMENT

I take this opportunity to express my greatest gratitude to my mentor, Pavan Sumohana, whose strong encouragement, sage advice, and motivational guidance were crucial during the course of this project. Their vast experience, deep knowledge, and precious suggestions played a critical role in making me understand AI-based medical forecasting systems. Not only did they lead me through difficult stages but also helped me finish the project with enhanced accuracy and efficiency.

In addition, I am deeply thankful to AICTE TechSaksham, Microsoft, and SAP for giving a shared learning platform. This internship has immensely developed my technical skills by exposing me to actual AI applications in the healthcare sector. This experience has enlarged my knowledge pool and made me stronger to design and deploy AI models.

Lastly, I would like to extend my sincere gratitude to my peers, relatives, and friends for their persistent encouragement and support. Their steadfast faith in me kept me going through the project, enabling me to successfully complete it.

ABSTRACT

Problem: Increased incidence of chronic diseases overburdens world healthcare.

Solution: Early disease prediction through machine learning.

Methodology:

Public repository data collection.

Data preprocessing and cleaning.

Exploratory Data Analysis (EDA).

Development of models (Random Forest, etc.) for 6 diseases.

Streamlit web application for live predictions.

Outcome: Validates the effectiveness of AI in disease prediction.

Future: Integration of real-time data, increased disease coverage, clinical proof.

Major Factors:

Data Quality: The accuracy of predictions relies significantly on the quality and availability of data that the model is trained on.

Model Selection and Tuning: Selection of machine learning algorithms as well as their parameters plays an important role in predictive performance.

Real-world Validation: Real-world use as well as clinical trials play a vital role in ensuring the effectiveness of the model.

Data Integration: Real-time integration of data from multiple sources becomes a key feature of a dynamic and accurate system.

TABLE OF CONTENT

Abstract
Chapter 1. Introduction
1.1 Problem Statement	
1.2 Motivation	
1.3 Objectives	
1.4 Scope of the Project	
Chapter 2. Literature Survey
2.1 Scholarly Works that Already Exist	
2.2 Deployed Systems and Methods	
2.3 Shortcomings in Current Solutions	
Chapter 3. Proposed Methodology
3.1 System Design	
3.2 Requirement Specification	
3.2.1 Hardware Requirements	
3.2.2 Software Requirements	
Chapter 4. Implementation and Results
4.1 Model Building for Disease Prediction	
4.2 Web App	
4.3 Snapshots of Result	
4.4 GitHub Link for Code	
Chapter 5. Discussion and Conclusion
5.1 Future Work	
5.2 Conclusion	
References

LIST OF FIGURES

Figure No.	Figure Caption	Page No.
Figure 1	Data Preparation Process	8
Figure 2	Breast Cancer Prediction Result	13
Figure 3	Lung Cancer Prediction Result	14
Figure 4	Liver Disease Prediction Result	15
Figure 5	Diabetes Prediction Model	16
Figure 6	Kidney Disease Prediction Model	17
Figure 7	Heart Disease Prediction Model	18
Figure 8	Streamlit Web Application UI	19
Figure 9		

CHAPTER 1

Introduction

1.1 Problem Statement:

"Healthcare facilities find it difficult to process large volumes of patient data for prediction of early disease. Current tools are not accurate, leading to late diagnoses. Predictive systems based on AI are required to process data effectively, enhance diagnostic precision, allow timely interventions, cut costs, and improve patient outcomes while maintaining ethical data management."

1.2 Motivation:

"Driven by the demand for quicker, more precise diagnoses, this project seeks to utilize machine learning to develop an automated disease prediction system. Through facilitating early interventions and lowering mortality, we hope to increase patient care efficiency and overall healthcare outcomes."

1.3 Objective:

- 1 "Implement and deploy strong machine learning algorithms (Random Forest, Decision Tree, etc.) to precisely forecast diseases from patient data."
- 2 "Build an interactive Streamlit web app for live disease forecasting and visualization of AI-powered insights."
- 3 "Automate data processing and prediction workflows to streamline clinical decision-making and improve healthcare efficiency."
- 4 - "Give healthcare professionals AI-driven insights to improve diagnostic accuracy and aid in well-informed clinical decisions, ultimately enhancing patient outcomes."

1.4 Scope of the Project:

This project will create a Python-based machine learning system to predict Diabetes, Heart Disease, Kidney Disease, Liver Disease, Breast Cancer, Lung Cancer.

The scope includes:

- 1 Build Python/ML models for 6 disease predictions, including data preprocessing and training.
- 2 Develop a Streamlit web application for visualizing results.

3 Recognize limitations of accuracy; real-world verification is required.

CHAPTER 2

Literature Survey

2.1 Scholarly Works that Already Exist

Recent studies highlight the expanding participation of computational learning methods in healthcare, specifically for predictive analysis and immediate detection of medical conditions. Random Forest, Decision Trees, Support Vector Machines (SVM), and Logistic Regression are among the algorithms that have repeatedly been used for disease classification and risk prediction. Methods including ensemble learning approaches and complex neural network architectures have demonstrated enhanced accuracy in medical diagnosis. Yet the enabling of real-time predictions and easy system deployment is another area that needs to be investigated further.

2.2 Deployed Systems and Methods

The project employs the Random Forest Classifier in combination with other machine learning algorithms and makes use of Python libraries like NumPy, Pandas, Scikit-learn, and Joblib. The models are conditioned on disease-specific data and involve feature processing and parameter adjustment to maximize predictive accuracy. The system scans patient health data for estimating probabilities of different diseases, such as Diabetes, Heart Disease, Kidney Disease, Liver Disease, Breast Cancer, Lung Cancer, and provides real-time predictive responses.

2.3 Shortcomings in Current Solutions

Existing solutions have a number of shortcomings, including:

- Predictive Accuracy and Reliability: Current models tend to lack accuracy due to imbalances in data or missing records.
- Live Data Incorporation: Most solutions fail to support real-time predictive features, which diminishes their utility in clinical applications.
- Confidentiality and Patient Data Protection: Securing sensitive medical data continues to be a challenge in the use of AI in healthcare environments.

CHAPTER 3

Proposed Methodology

3.1 System Design

- ☐ Data Acquisition: Download disease datasets from Kaggle/UCI.
- ☐ Data Refinement: Clean data, convert categories, scale features.
- ☐ Model Building: Build ML models (Random Forest, etc.) for each disease.
- ☐ Web Deployment: Develop Streamlit application for real-time predictions; provide clear results.



3.2 Requirement Specification

Listing down the tool and technologies needed to deploy the solution.

3.2.1 Hardware Requirements:

- ☐ Processor : AMD Ryzen 5 / Intel i5 or greater
- ☐ Ram: 8 GB or greater
- ☐ Storage: 1GB or greater
- ☐ Operating System : Windows10 / linux / macOS

3.2.2 Software Requirements:

- ☐ Programming Language: Python
- Libraries:
 - ☐ Data Processing : Pandas, NumPy
 - ☐ Machine Learning: Scikit-learn, Joblib
 - ☐ Web Development: Streamlit

□ Development Environment :VS Code, Jupyter Notebook

CHAPTER 4

Implementation and Result

4.1 Model Building for Disease Prediction

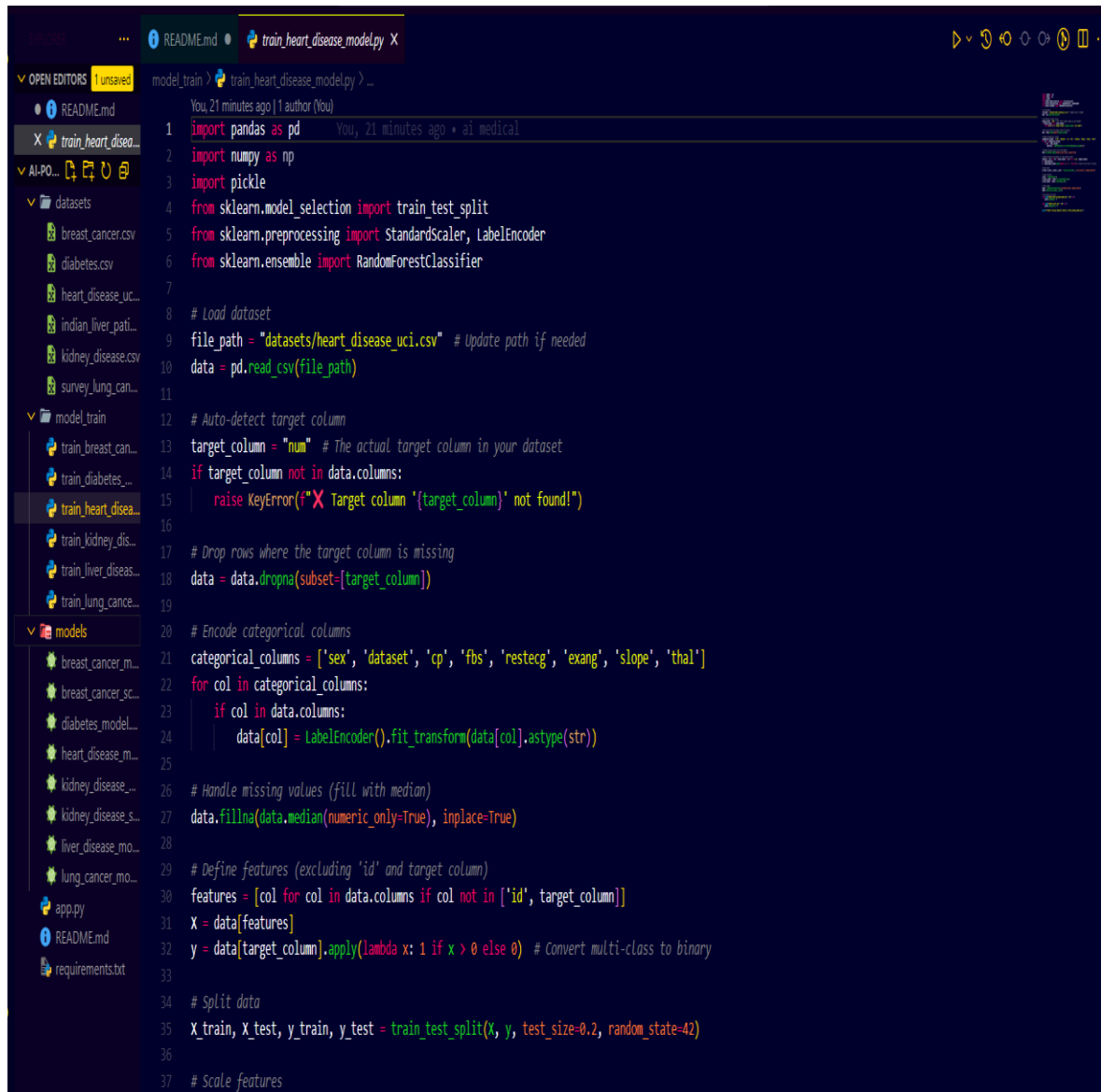
Developing several prediction models for Diabetes, Heart Disease, and Kidney Disease are part of the project using machine learning methods.

- The Heart Disease model predicts heart disease based on features such as age, blood pressure, cholesterol, and other medical factors.
- The Diabetes model predicts if the individual is diabetic or not by considering factors like glucose, insulin, BMI, and skin thickness.
- The Kidney Disease model predict kidney disease based on some factors like age, blood pressure, Specific Gravity, Albumin Level, Sugar Level

Scikit-Learn and Pandas libraries are used to train the models, and accuracy scores are used to measure the predictions.

Following are some code segments from the model development process:

Heart Disease



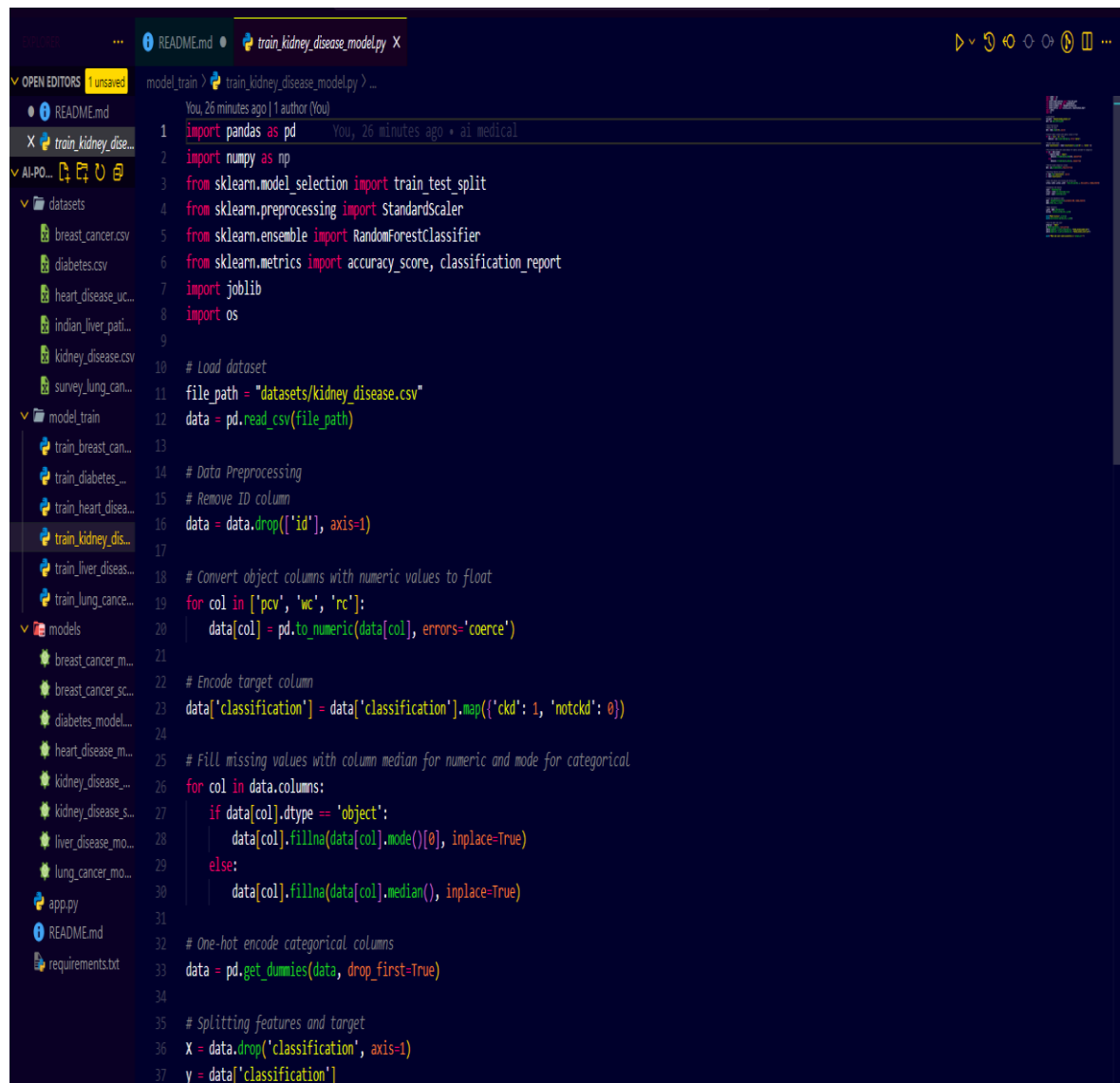
```
1 import pandas as pd
2 import numpy as np
3 import pickle
4 from sklearn.model_selection import train_test_split
5 from sklearn.preprocessing import StandardScaler, LabelEncoder
6 from sklearn.ensemble import RandomForestClassifier
7
8 # Load dataset
9 file_path = "datasets/heart_disease_uci.csv" # Update path if needed
10 data = pd.read_csv(file_path)
11
12 # Auto-detect target column
13 target_column = "num" # The actual target column in your dataset
14 if target_column not in data.columns:
15     raise KeyError(f"Target column '{target_column}' not found!")
16
17 # Drop rows where the target column is missing
18 data = data.dropna(subset=[target_column])
19
20 # Encode categorical columns
21 categorical_columns = ['sex', 'dataset', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'thal']
22 for col in categorical_columns:
23     if col in data.columns:
24         data[col] = LabelEncoder().fit_transform(data[col].astype(str))
25
26 # Handle missing values (fill with median)
27 data.fillna(data.median(numeric_only=True), inplace=True)
28
29 # Define features (excluding 'id' and target column)
30 features = [col for col in data.columns if col not in ['id', target_column]]
31 X = data[features]
32 y = data[target_column].apply(lambda x: 1 if x > 0 else 0) # Convert multi-class to binary
33
34 # Split data
35 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
36
37 # Scale features
```



Diabetes Disease

```
model_train > train_diabetes_model.py X
You, 24 minutes ago | 1 author (You)
1 import pandas as pd
2 import numpy as np
3 from sklearn.model_selection import train_test_split
4 from sklearn.ensemble import RandomForestClassifier
5 from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
6 import joblib
7 import os
8 # 1. Load the Dataset
9 # Use the correct path based on your folder structure
10 file_path = "datasets/diabetes.csv"
11 data = pd.read_csv(file_path)
12 data.fillna(data.median(numeric_only=True), inplace=True)
13
14 # Features and target separation
15 X = data.drop("Outcome", axis=1)
16 y = data["Outcome"]
17
18 # Splitting data into training and testing sets
19 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
20 # 3. Model Training
21 print("\nTraining Diabetes Model...")
22 model = RandomForestClassifier(n_estimators=200, random_state=42)
23 model.fit(X_train, y_train)
24 # 4. Model Evaluation
25 y_pred = model.predict(X_test)
26 accuracy = accuracy_score(y_test, y_pred)
27
28 print("\nModel Evaluation:")
29 print(f"Accuracy: {accuracy * 100:.2f}%")
30 print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
31 print("Classification Report:\n", classification_report(y_test, y_pred))
32 # 5. Save the Model
33 output_path = "models/diabetes_model.pkl"
34 os.makedirs(os.path.dirname(output_path), exist_ok=True) # Ensure directory exists
35
36 joblib.dump(model, output_path)
37
```

Kidney Disease



```
1 import pandas as pd
2 import numpy as np
3 from sklearn.model_selection import train_test_split
4 from sklearn.preprocessing import StandardScaler
5 from sklearn.ensemble import RandomForestClassifier
6 from sklearn.metrics import accuracy_score, classification_report
7 import joblib
8 import os
9
10 # Load dataset
11 file_path = "datasets/kidney_disease.csv"
12 data = pd.read_csv(file_path)
13
14 # Data Preprocessing
15 # Remove ID column
16 data = data.drop(['id'], axis=1)
17
18 # Convert object columns with numeric values to float
19 for col in ['pcv', 'wc', 'rc']:
20     data[col] = pd.to_numeric(data[col], errors='coerce')
21
22 # Encode target column
23 data['classification'] = data['classification'].map({'ckd': 1, 'notckd': 0})
24
25 # Fill missing values with column median for numeric and mode for categorical
26 for col in data.columns:
27     if data[col].dtype == 'object':
28         data[col].fillna(data[col].mode()[0], inplace=True)
29     else:
30         data[col].fillna(data[col].median(), inplace=True)
31
32 # One-hot encode categorical columns
33 data = pd.get_dummies(data, drop_first=True)
34
35 # Splitting features and target
36 X = data.drop('classification', axis=1)
37 y = data['classification']
```


4.2 Web App

Web application is made by using app.py, allow user to enter health information and get prediction

Below I put some Screenshot that show how it predict disease

Breast Cancer: -

AI-Powered Medical Predictor

 Choose Disease to Analyse

Breast Cancer



Radius Mean

0.03

- +

Texture Mean

0.07

- +

Smoothness Mean

0.07

- +

Compactness Mean

0.09

- +

Concavity Mean

0.07

- +

 Predict Disease

Prediction Result: Negative (99.00% confidence)

Lung Cancer: -

AI-Powered Disease Predictor

☒ Select a Disease to Predict

Lung Cancer

Gender

☒ Male

☐ Female

Age

11

Smoking


☐ Yes

☒ No

Coughing

☒ Yes

☐ No

 Predict

☒ Disease Prediction Result: Negative (51.50% confidence)

Liver Disease: -

AI-Powered Disease Predictor

☒ Select a Disease to Predict

Liver Disease

Age

20

Gender

☒ Male

☐ Female

Total Bilirubin


5.00

Direct Bilirubin

0.06

Alkaline Phosphatase


2

 Predict

☒ Disease Prediction Result: Negative (55.00% confidence)

Diabetes: -

AI-Powered Disease Predictor

 Select a Disease to Predict

Diabetes



Number of Pregnancies

2

-

+

Glucose Level

25

-

+

Blood Pressure

85

-

+

Skin Thickness

0

-

+

Insulin Level

0

-

+

BMI

0.06

-

+

Diabetes Pedigree Function

0.20

-

+

Age


20

-

+




Predict

 Disease Prediction Result: Negative (95.50% confidence)

Kidney Disease: -

AI-Powered Disease Predictor

 Select a Disease to Predict

Kidney Disease



Age

20



Blood Pressure

85



Specific Gravity

1.00



Albumin Level

2




Sugar Level

5



Predict

 **Disease Prediction Result:** Positive (95.00% confidence)

Heart Disease: -

AI-Powered Disease Predictor

 Select a Disease to Predict

Heart Disease



Age

20



Sex

☐ Male

☒ Female

Chest Pain Type

2



Resting Blood Pressure

50




Cholesterol Level

100



Predict

 Disease Prediction Result: Positive (57.00% confidence)

AI-Powered Disease Predictor

Select a Disease to Predict

Diabetes

Diabetes

Heart Disease

Lung Cancer

Liver Disease

Breast Cancer

Kidney Disease

Skin Thickness

0

Insulin Level

0

BMI

0.00

Diabetes Pedigree Function

0.00

4.3 GitHub Link for code

With the help of this link anyone can see the project & code

GitHub Repository-

<https://github.com/Anandkumarkashyap/AI-Powered-Disease-Predictor>

CHAPTER 5

Discussion and Conclusion

5.1 Future Work:

For optimizing the efficiency and reliability of the system, the following are recommended:

Enhancing predictive precision by using state-of-the-art machine learning:

Using more advanced architectures like Deep Learning Networks (DLNs), Sequential Neural Networks (SNNs), and Image-based Neural Networks (IBNNs).

The use of such new paradigms for learning can greatly improve predictive results and identify more subtle relationships in patient physiological data.

Adding patient data in real-time for dynamic testing:

Facilitating ongoing patient data collection by networked medical devices, sensor technology, or health data interfaces. This would render the site more directly useful and relevant to healthcare professionals by allowing instant and responsive predictions. Expanding model coverage to encompass more varieties of medical conditions:

Creating specialized models for other diseases such as Renal Dysfunction, Hepatic Disorders, Neurodegenerative Diseases, and Pulmonary Neoplasms. This would offer flexibility and usability of the system to various forms of clinical settings.

5.2 Conclusion:

This project successfully demonstrates the application of machine intelligence algorithms for health condition onset prediction. By using trained models on clinical data repositories, the platform offers precise predictions for Glucose Imbalance, Cardiovascular Disorders, and Movement Impairment Syndromes.

Important Observations:

Streamlit graphical user interface provides a friendly interface, offering seamless interaction and ease of use for clinicians and scientific researchers.

These computer models of learning constructed in this project are beneficial for analytical insights by uncovering patterns in patient-provided information, thereby allowing medical personnel to formulate evidence-based treatment schedules.

The design of the system is scalable and allows immediate diagnosis and implementation of additional disease models in the future.

Lastly, the project shows the groundbreaking potential of machine learning to revolutionize medicine, specifically the capacity to predict disease before it happens and enhancing better clinical judgment.

REFERENCES

- [1]. Ming-Hsuan Yang, David J. Kriegman, Narendra Ahuja, “Detecting Faces in Images: A Survey”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume. 24, No. 1, 2002.

