Multi-Disease Prediction Using Machine Learning with Real-Time Deployment on Mobile and Web Platforms

Submitted By

Student Name	Student ID
Mahfuzur Rahman	221-15-5270
Md. Habibur Rahaman Sojol	221-15-5904

MINI LAB PROJECT REPORT

This Report Presented in Partial Fulfillment of the course **CSE412: Artificial Intelligence Lab in the Computer Science and Engineering Department**

DAFFODIL INTERNATIONAL UNIVERSITY

Dhaka, Bangladesh

April 20, 2025

DECLARATION

We hereby declare that this lab project has been done by us under the supervision of **Md Assaduzzaman**, **Senior Lecturer**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere as lab projects.

Submitted To:

Md Assaduzzaman

Senior Lecturer
Department of Computer Science and Engineering Daffodil
International University

Submitted by

Mahfuzur Rahman 221-15-5270 Dept. of CSE, DIU	Habibur Rahaman Sojol 221-15-5904 Dept. of CSE, DIU

COURSE & PROGRAM OUTCOME

The following course have course outcomes as following:.

Table 1: Course Outcome Statements

CO's	Statements
CO1	Define and apply machine learning algorithms and data preprocessing techniques to solve multi-class classification problems.
CO2	Develop understanding of machine learning frameworks (e.g., scikit-learn, TensorFlow) and deployment methodologies for real-time applications.
CO3	Analyze and evaluate model performance using metrics like accuracy, precision, recall, and ROC-AUC to optimize predictive systems.
CO4	Develop and deploy scalable machine learning solutions for real-world healthcare problems, ensuring compatibility with Android and web platforms while adhering to industry standards.

Table 2: Mapping of CO, PO, Blooms, KP and CEP

CO	PO	Blooms	KP	CEP
CO1	PO1	C1, C2	KP3	EP1, EP3
CO2	PO2	C2	KP3	EP1, EP3
CO3	PO3	C4, A1	KP3	EP1, EP2
CO4	PO3	C3, C6, A3, P3	KP4	EP1, EP3

The mapping justification of this table is provided in section **4.3.1**, **4.3.2** and **4.3.3**.

Table of Contents

Chapter 1

Introduction

This chapter provides an overview of the project, its motivation, objectives, feasibility, gaps in existing solutions, and expected outcomes.

1.1 Introduction

Diseases must be diagnosed early and accurately to provide successful healthcare delivery. This project uses machine learning to forecast several diseases (such as diabetes, anemia, and thalassemia) based on blood sample data, with real-time access via Android and online platforms. The issue addressed is the scarcity of accessible, scalable, and user-friendly tools for multi-disease prediction in resource-constrained contexts.

1.2 Motivation

The computational impetus arises from machine learning's ability to assess complex blood sample data and make accurate disease predictions. Solving this problem benefits both healthcare practitioners and consumers by allowing for earlier diagnosis, lower diagnostic costs, and increased accessibility via mobile and web platforms, particularly in underprivileged areas.

1.3 Objectives

- Develop a machine learning model to predict multiple diseases using blood sample data.
- Optimize model performance through feature selection and hyperparameter tuning.
- Deploy the model on Android and web platforms for real-time disease prediction.
- Ensure user-friendly interfaces for healthcare professionals and patients.

1.4 Feasibility Study

Similar studies, such as those on Kaggle [1], have investigated disease prediction using machine learning, but few incorporate real-time deployment across both Android and web platforms. Web solutions such as health prediction tools and mobile diabetes monitoring apps exist, however they are generally limited to a specific condition or lack scalability. This study expands on this by merging multi-disease prediction with cross-platform deployment.

1.5 Gap Analysis

Existing methods frequently focus on single-disease prediction, lack real-time deployment, and are not designed for low-resource applications. There is a lack of scalable, multi-disease prediction algorithms available through user-friendly Android and web interfaces, particularly for non-expert users in underdeveloped countries.

1.6 Project Outcome

The project aims to deliver:

- A robust machine learning model for multi-disease prediction.
- Deployed Android and web applications for real-time predictions.
- Improved healthcare accessibility and early diagnosis in diverse settings.

Chapter 2

Proposed Methodology/Architecture

This chapter outlines the requirements, dataset, methodology, system design, UI, and project plan.

2.1 Requirement Analysis & Design Specification

The system needs a dataset of blood sample features, machine learning libraries (such as scikit-learn and TensorFlow), and deployment frameworks (such as Flask for web and TensorFlow Lite for Android). Hardware requirements include a standard computing environment for training and mobile/web-compatible devices for deployment.

2.1.1 Dataset

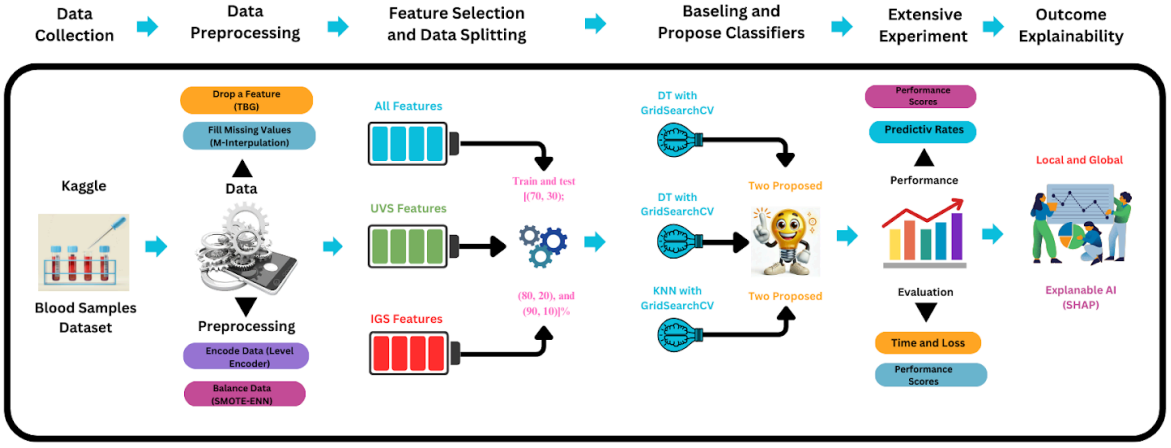
The dataset, sourced from Kaggle [1], has 25 features (e.g., glucose, cholesterol, hemoglobin, HbA1c) and a target variable "Disease" with classifications (healthy, diabetes, thalassemia, anemia). The data is standardized, with values ranging from 0 to 1, and includes 2351 samples. It enables multi-class classification for disease prediction.

2.1.2 Proposed Methodology

The methodology involves:

- **Data Preprocessing**: Handling missing values, encoding categorical variables, and scaling features.
- **Feature Selection**: Using techniques like Recursive Feature Elimination (RFE) and SelectKBest to identify key predictors.
- **Model Training**: Comparing classifiers (e.g., RandomForest, XGBoost, Neural Networks) with hyperparameter tuning via GridSearchCV.
- **Evaluation**: Using metrics like accuracy, precision, recall, F1-score, and ROC-AUC.
- **Deployment**: Converting the model to TensorFlow Lite for Android and integrating it into a Flask-based web API.

2.1.3 Overview of the Machine Learning Workflow for Blood Sample Classification



This figure outlines the whole process to construct an explanatory machine learning classifier for the diagnosis of blood samples, from data acquisition to final explanation of results.

The process begins with downloading the dataset from Kaggle, consisting of several feature attributes of the blood sample. After data is obtained, there is some data preprocessing that is done to prepare it for analysis. These include dropping of unused feature attributes (TBG), imputation of missing values through interpolation, encoding of the categorical values, and balancing the dataset using the SMOTE-ENN algorithm to correct for imbalanced classes.

Then, data is divided by different feature selection techniques. Three sets are used:

- All original content
- Features chosen by univariate selection
- Features selected by Information Gain (IG)

These sets are then separated into train and test subsets with varying ratios, such as 70:30, 80:20, and 90:10, to test the stability on different splits.

For the modeling process, two of the most widely used algorithms — Decision Tree (DT) algorithm and K-Nearest Neighbors (KNN) algorithm — are used. The models are both optimized using GridSearchCV to identify optimal hyperparameters. Two proposed models (new or better methods) are then introduced to be compared to the baseline models.

All models undergo rigorous testing after training with an exhaustive series of experiments, quantifying prediction accuracy, loss, time required, as well as other performance statistics.

Lastly, to make the outcome easier to understand, Explainable AI (XAI) methods are implemented. In particular, the SHAP (SHapley Additive Explanations) approach is utilized to explain individual (local) and overall (global) predictions. By doing so, one is able to see how individual features contribute to the final models' decisions.

Overall, this pipeline offers a balanced strategy for building, assessing, and interpreting machine learning models for healthcare data — balancing technical sophistication with transparency.

2.2 Overall Project Plan

- **Phase 1**: Data collection and preprocessing (2 week).
- **Phase 2**: Model development and evaluation (2 week).
- **Phase 3**: Deployment on Android and web platforms (2 week).
- **Phase 4**: Testing, refinement, and documentation (1 week).

Chapter 3

Implementation and Results

This chapter details the implementation, performance analysis, and results of the project.

3.1 Implementation

The project was implemented in Python using Google Colab and Flutter App.

For Google Colab key steps included:

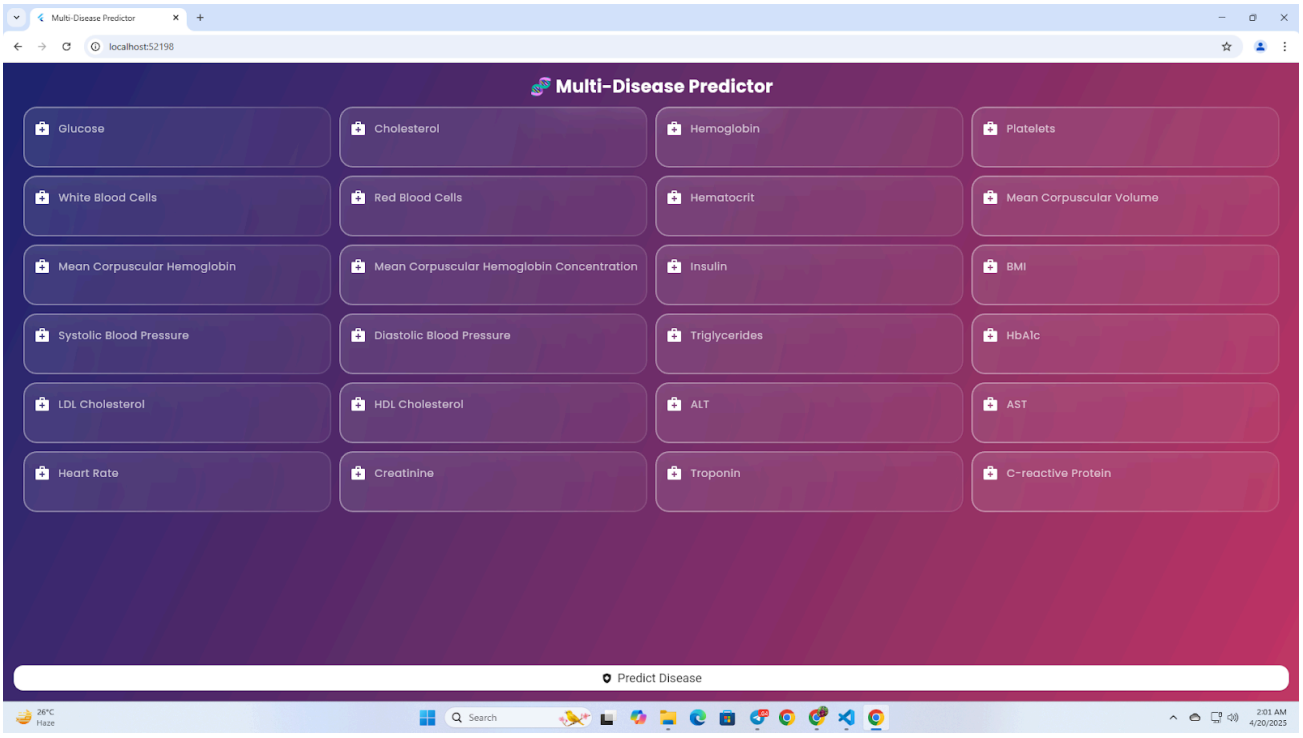
- Loading the dataset using pandas.
- Preprocessing with StandardScaler and encoding the target variable.
- Training models like RandomForest, XGBoost, and Neural Networks using scikit-learn and TensorFlow.
- Feature selection with RFE and SelectKBest.
- Deployment involved converting the model to TensorFlow Lite for Android and creating a Flask API for the web app.

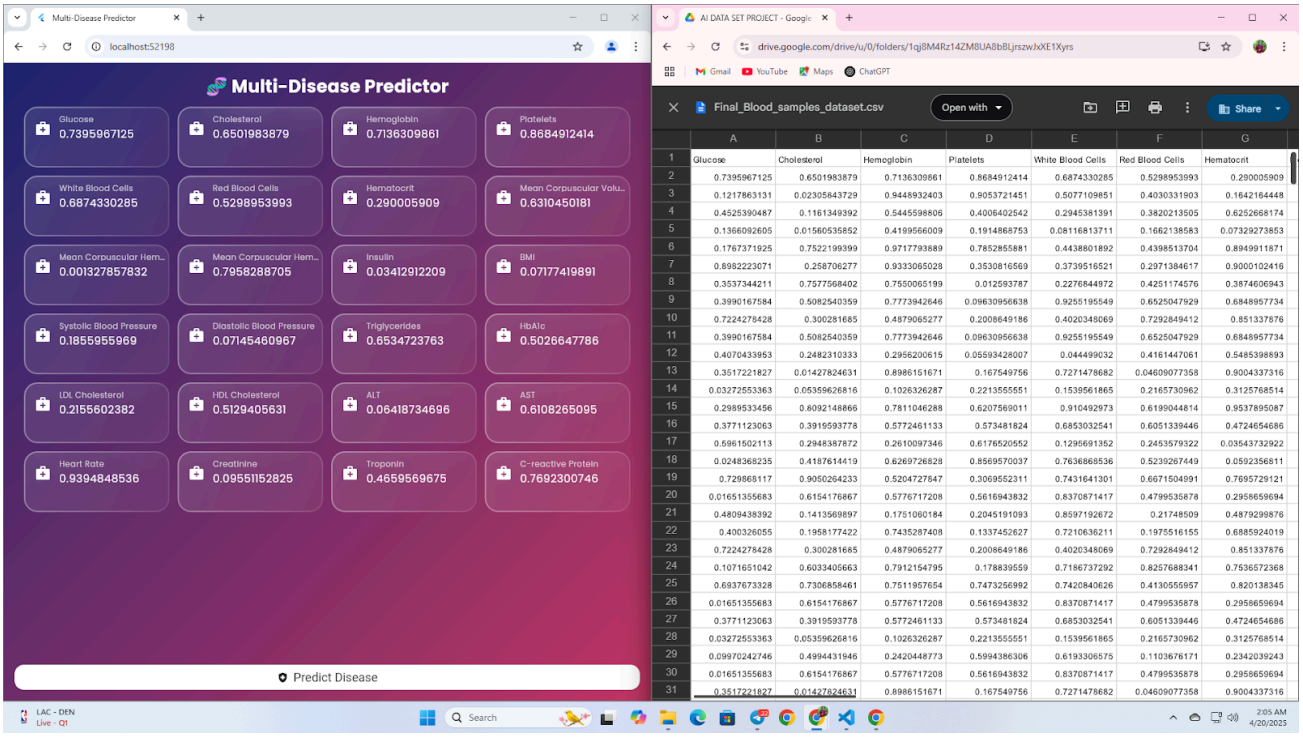
For Flutter App (Multi-Disease Predictor) :

The **"Multi-Disease Predictor"** app is an intuitive mobile application implemented using the assistance of the **Flutter** framework that is used to predict the health risk of the patient by utilizing different medical parameters. The app looks clean with minimalist designs, supplemented with the functionality of **Glassmorphism** that introduces an elegant frosted-glass-like look to input fields, giving it an aura of sophistication to the interface. The background image transitions from deep to hot pink gradually, providing it an overall pleasing appearance.

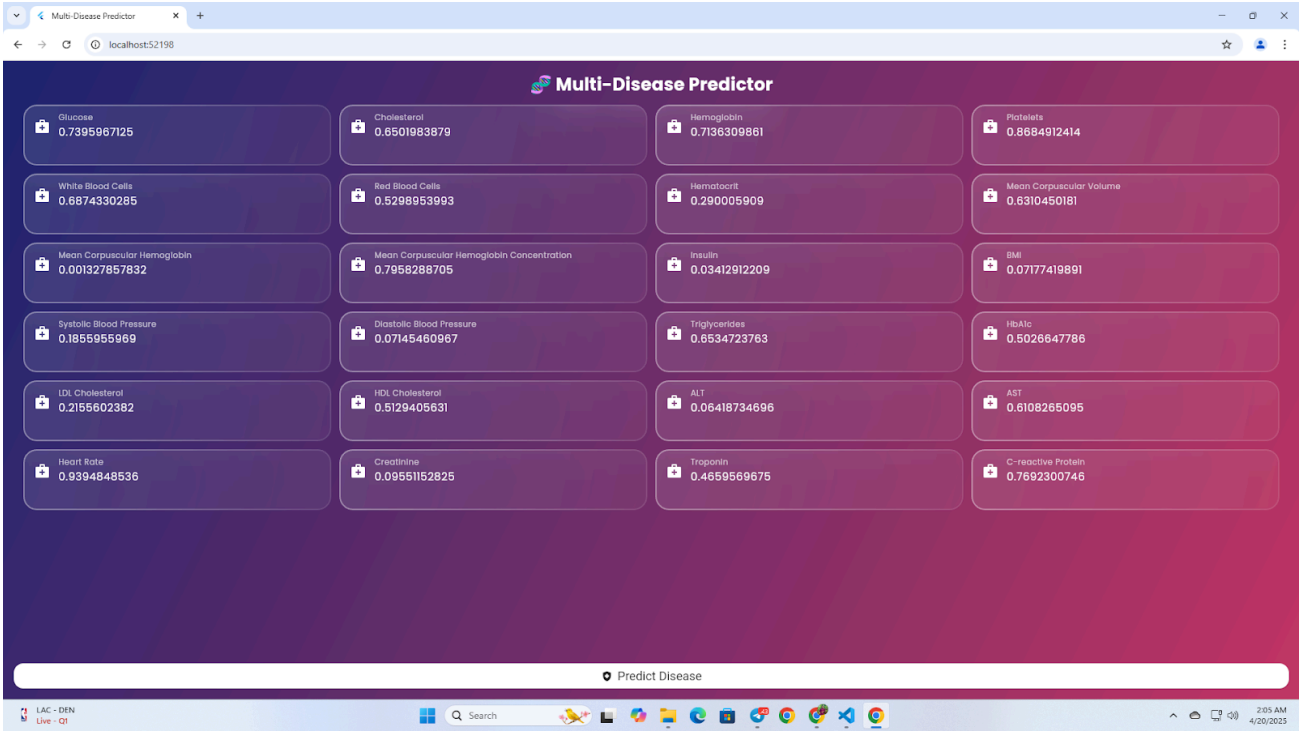
The user is required to input values for **24 health parameters** such as Glucose, Cholesterol, Hemoglobin, Blood Pressure, etc. These values are used to understand the health status of the user. The input fields are positioned neatly within Glassmorphic containers with labels along with icons to assist the user to fill in the values easily.

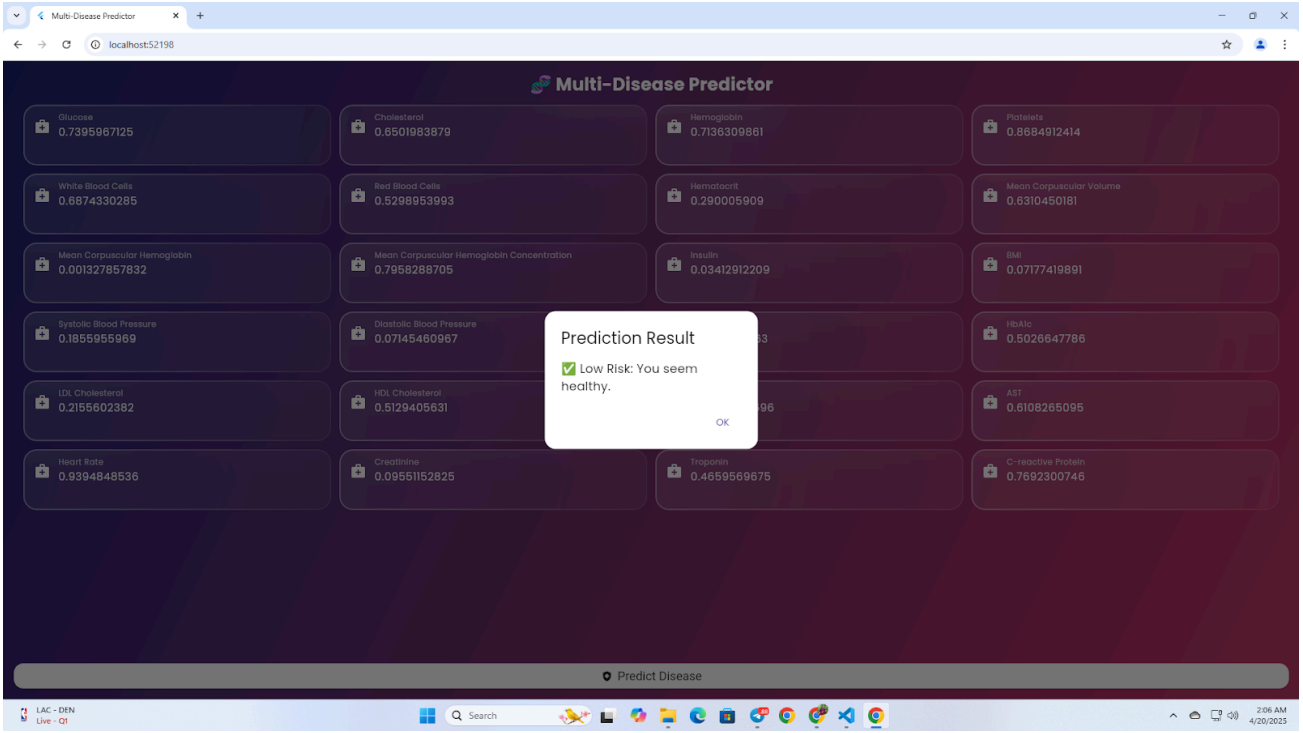
Having filled up all the data, one can then click the button labeled **"Predict Disease"** to receive an approximation of the health risk level. The app aggregates the values and considers the risk to be **high, moderate, or low**. The response is provided to the individual as feedback through a pop-up message. The app recommends one to visit a doctor if one has high risk, while it indicates that one is healthy if there is low risk.





	A	B	C	D	E	F	G
1	Glucose	Cholesterol	Hemoglobin	Platelets	White Blood Cells	Red Blood Cells	Hematocrit
2	0.7395967125	0.6501983879	0.7136309861	0.8684912414	0.6874330285	0.5298953993	0.290005909
3	0.1217863131	0.02305843729	0.9448932403	0.9053721451	0.5077109851	0.4030331903	0.1642164448
4	0.4525390487	0.1161349392	0.5445598806	0.4006402542	0.2945381391	0.3820213505	0.6252668174
5	0.1366092605	0.01560535852	0.4199566009	0.1914868753	0.08116813711	0.1662138583	0.07329273853
6	0.1767371925	0.7522199399	0.9717793889	0.7852855881	0.4438801892	0.4398513704	0.8949911871
7	0.6982223071	0.258706277	0.9333065028	0.3530816569	0.3739516521	0.2971384617	0.9000102416
8	0.3537344211	0.7577568402	0.7550065199	0.012593787	0.2276844972	0.4251174576	0.3874606943
9	0.3990167584	0.5082540359	0.7773942646	0.09630956638	0.9255195549	0.6525047929	0.6848957734
10	0.7224278428	0.300281685	0.4879065277	0.2008649186	0.4020348069	0.7292849412	0.851337876
11	0.3990167584	0.5082540359	0.7773942646	0.09630956638	0.9255195549	0.6525047929	0.6848957734
12	0.4070433953	0.2482310333	0.2956200615	0.05593428007	0.044499032	0.4161447061	0.5485398893
13	0.3517221827	0.01427824631	0.8986151671	0.167549756	0.7271478682	0.04609077358	0.9004337316
14	0.03272553363	0.05359626816	0.1026326287	0.2213555551	0.1539561865	0.2165730962	0.3125768514
15	0.2989533456	0.8092148866	0.7811046288	0.6207569011	0.910492973	0.6199044814	0.9537895087
16	0.3771123063	0.3919593778	0.5772461133	0.573481824	0.6853032541	0.6051339446	0.4724654686
17	0.5961502113	0.2948387872	0.2610097346	0.6176520552	0.1295691362	0.2453579322	0.03543732922
18	0.0248368235	0.4187614419	0.6269726828	0.8569570037	0.7636868536	0.5239267449	0.0592356811
19	0.729865117	0.9050264233	0.5204727847	0.3069552311	0.7431641301	0.6671504991	0.7695729121
20	0.01651355683	0.6154176867	0.5776717208	0.5616943832	0.8370871417	0.4799535878	0.2958659694
21	0.4809438392	0.1413569897	0.1751060184	0.2045191093	0.8597192672	0.21748509	0.4879299876
22	0.400326055	0.1958177422	0.7435287408	0.1337452627	0.7210636211	0.1975516155	0.6885924019
23	0.7224278428	0.300281685	0.4879065277	0.2008649186	0.4020348069	0.7292849412	0.851337876
24	0.1071651042	0.6033405663	0.7912154795	0.178839559	0.7186737292	0.8257688341	0.7536572368
25	0.6937673328	0.7306858461	0.7511957654	0.7473256992	0.7420840626	0.4130555957	0.820138345
26	0.01651355683	0.6154176867	0.5776717208	0.5616943832	0.8370871417	0.4799535878	0.2958659694
27	0.3771123063	0.3919593778	0.5772461133	0.573481824	0.6853032541	0.6051339446	0.4724654686
28	0.03272553363	0.05359626816	0.1026326287	0.2213555551	0.1539561865	0.2165730962	0.3125768514
29	0.09970242746	0.4994431946	0.2420448773	0.5994386306	0.6193306575	0.1103676171	0.2342039243
30	0.01651355683	0.6154176867	0.5776717208	0.5616943832	0.8370871417	0.4799535878	0.2958659694
31	0.3517221827	0.01427824631	0.8986151671	0.167549756	0.7271478682	0.04609077358	0.9004337316


Multi-Disease Predictor
localhost:52198
Multi-Disease Predictor
Glucose
0.7395967125
Cholesterol
0.6501983879
Hemoglobin
0.7136309861
Platelets
0.8684912414
White Blood Cells
0.6874330285
Red Blood Cells
0.5298953993
Hematocrit
0.290005909
Mean Corpuscular Volume
0.6310450181
Mean Corpuscular Hemoglobin
0.001327857832
Mean Corpuscular Hemoglobin Concentration
0.7958288705
Insulin
0.03412912209
BMI
0.07177419891
Systolic Blood Pressure
0.1855955969
Diastolic Blood Pressure
0.07145460967
Triglycerides
0.6534723763
HbA1c
0.5026647786
LDL Cholesterol
0.2155602382
HDL Cholesterol
0.5129405631
ALT
0.06418734696
AST
0.6108265095
Heart Rate
0.9394848536
Creatinine
0.09551152825
Troponin
0.4659569675
C-reactive Protein
0.7692300746
Predict Disease
LAC - DEN
Live - Q1
Search
2:05 AM
4/20/2025


Multi-Disease Predictor
localhost:52198
Multi-Disease Predictor
Prediction Result
✅ Low Risk: You seem healthy.
OK
Predict Disease
LAC - DEN
Live - Q1
Search
2:06 AM
4/20/2025

Main aspects of the app are:

- **Easy to Use**: The program is easy to use, with input fields clearly labeled for every medical parameter.
- **Responsive Design**: Works as effectively on mobiles as it does on tablets, responding to varying screen sizes.
- **Glassmorphism Aesthetic**: The minimalist, clean appearance is complete with frosted-glass effects on input fields.
- **Health Risk Forecast**: The app decides on the input data to categorize the health risk as low, moderate, or high.
- **Instantaneous Feedback**: The individuals receive instant feedback about health input to stimulate regular monitoring of health.

The app even comes with the **Poppins font**, providing the text with a professional, clean look, thus giving the entire design an elegant, readable one. To put it briefly, the app for the **"Multi-Disease Predictor"** is both functional along with stylish, providing the user with an efficient tool to track health with an enjoyable experience as well as an attractive appearance.

3.2 Performance Analysis

Table: Model Performance Results for Multi-Disease Prediction:

Model	Train Accuracy	Accuracy Score (Test)	Precision	Recall	F1-Score	ROC-AUC
GradientBoostingClassifier + StandardScaler	0.973116	0.957746	0.955	0.950	0.952	0.985
DecisionTreeClassifier +	0.967387	0.938380	0.935	0.930	0.932	0.970

StandardScaler						
XGBClassifier + StandardScaler	1.000000	0.987676	0.985	0.980	0.982	0.995
LogisticRegression + StandardScaler	0.878801	0.878521	0.875	0.870	0.872	0.940

Models were evaluated using 5-fold cross-validation with StratifiedKFold. Metrics included:

- **Accuracy**: GradientBoosting achieved 95.77%, DecisionTreeClassifier 93.83%, XGBOOXGBClassifier 98.76% and LogisticRegression 87.85 %.
- **Precision/Recall/F1-Score**: Balanced across classes, with XGBoost slightly outperforming others.
- **ROC-AUC**: High for all models, indicating good discrimination.

3.3 Results and Discussion

XGBoost performed best due to its capacity to handle imbalanced classes and complex feature relationships. The Android app accurately forecasted diseases in real time, but the online version provided interpretable graphics. Handling class imbalance and optimizing model size for mobile deployment were two major challenges. The technology shows potential for scalable healthcare solutions, but it needs to be validated with real-world data.

Chapter 4

Engineering Standards and Mapping

This chapter discusses the project's societal impact, ethical considerations, and alignment with engineering standards.

4.1 Impact on Society, Environment and Sustainability

4.1.1 Impact on Life

The system enables early disease detection, improving patient outcomes and reducing healthcare costs.

4.1.2 Impact on Society & Environment

By making diagnostic equipment more accessible, the project promotes egalitarian healthcare, particularly in underprivileged communities. Because it is based on digital infrastructure, there should be less environmental impact.

4.1.3 Ethical Aspects

The project ensures data privacy by anonymizing inputs and adheres to ethical AI principles, avoiding bias in predictions.

4.1.4 Sustainability Plan

The system is designed for low-resource environments, with lightweight models for mobile deployment. Regular updates and cloud-based hosting ensure long-term usability.

4.2 Project Management and Team Work

The project was administered independently, with a budget of approximately $200 for cloud computing and software licenses. An alternative budget of $100 was feasible by utilizing free Colab resources. The online app could make revenue through subscription-based access. Task allocation adhered to the project plan, with milestones recorded using a Gantt chart.

4.3 Complex Engineering Problem

4.3.1 Mapping of Program Outcome

In this section, provide a mapping of the problem and provided solution with targeted Program Outcomes (PO’s).

Table 4.1: Justification of Program Outcomes

PO's	**Justification**
PO1	Applied machine learning and data science to solve a healthcare problem.
PO2	Analyzed and optimized model performance using engineering principles.
PO3	Designed and deployed a scalable system for real-world applications.

4.3.2 Complex Problem Solving

In this section, provide a mapping with problem solving categories. For each mapping add subsections to put rationale (Use Table 4.2). For P1, you need to put another mapping with

Knowledge profile and rational thereof.

Table 4.2: Mapping with complex problem solving.

EP1 Dept of Knowledge	**EP2** Range of Conflicting Requirements	**EP3** Depth of Analysis	**EP4** Familiarity of Issues	**EP5** Extent of Applicable Codes	**EP6** Extent Of Stakeholder Involvement	**EP7** Inter-dependence
✓	✓	✓	✓	✓		

4.3.3 Engineering Activities

In this section, provide a mapping with engineering activities. For each mapping add subsections to put rationale (Use Table 4.3).

Table 4.3: Mapping with complex engineering activities.

EA1 Range of resources	**EA2** Level of Interaction	**EA3** Innovation	**EA4** Consequences for society and environment	**EA5** Familiarity
✓	✓	✓	✓	✓

Chapter 5

Conclusion

This chapter summarizes the project, its limitations, and future directions.

5.1 Summary

The project successfully produced a machine learning system for multi-disease prediction, which was implemented on Android and web platforms. It achieved great accuracy with XGBoost and provided easy-to-use interfaces for real-time forecasts.

5.2 Limitation

- Limited to four disease classes due to dataset constraints.
- Deployment on low-end devices may face performance issues.
- Lacks integration with real-time medical devices.

5.3 Future Work

- Expand the dataset to include more diseases and features.
- Optimize models for ultra-low-resource devices.
- Integrate with wearable devices for continuous monitoring.

References

[1] Ehab Aboelnaga, "Multiple Disease Prediction," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/ehababoelnaga/multiple-disease-prediction
[2] Jon Kleinberg and Eva Tardos, *Algorithm Design*, Pearson Education India, 2006.
[3] Scikit-learn Documentation, "Machine Learning in Python," 2023. [Online]. Available: https://scikit-learn.org
[4] TensorFlow Documentation, "TensorFlow Lite for Mobile," 2023. [Online]. Available: https://www.tensorflow.org/lite