MINISTRY OF SCIENCE AND TECHNOLOGY

POSTS AND TELECOMMUNICATIONS INSTITUTE OF TECHNOLOGY

HO CHI MINH CITY CAMPUS

PROJECT REPORT
MACHINE LEARNING COURSE

Project Title:
A MACHINE LEARNING PROGRAM FOR PREDICTING THREE TYPES OF DISEASES

Ho Chi Minh City, June 2025

MINISTRY OF SCIENCE AND TECHNOLOGY

POSTS AND TELECOMMUNICATIONS INSTITUTE OF TECHNOLOGY
HO CHI MINH CITY CAMPUS

PROJECT REPORT
MACHINE LEARNING COURSE

Project Title:
“A Machine Learning Program For Predicting Three Types Of Diseases”

Instructor: Dr. Nguyễn Hồng Sơn
Group Members:
Võ Nguyên Hồng Diệp – N21DCVT015
Nguyễn Tấn Phúc – N21DCDK22

Class: E21CQCNTT01-N
School year: 2021-2026
Major: Artificial Intelligence
Program: High-Quality Information Technology

Ho Chi Minh City, June 2025

Acknowledgement!

*We, the project team, selected the topic "**A Program for Predicting Three Types of Diseases Based on Dataset and Machine Learning Algorithms**" as our capstone project for the Machine Learning course during our studies in Artificial Intelligence at the Posts and Telecommunications Institute of Technology – Ho Chi Minh City Campus.*

We would like to express our sincere gratitude to our instructor, Dr. Nguyễn Hồng Sơn from the Faculty of Information Technology 2, for his support, guidance, and feedback throughout the process of selecting and implementing this project.

Although we have put significant effort into the project, there are still some limitations due to our limited experience and knowledge. We welcome constructive feedback to help us improve and gain more practical insights for future applications.

Once again, we sincerely thank our instructor and everyone who supported us.

Ho Chi Minh City, June 2025

CONTENTS

LIST OF FIGURES AND TABLES

WORK DISTRIBUTION AMONG TEAM MEMBERS

Full Name	Student ID
Nguyễn Tấn Phúc	N21DCDK22
Võ Nguyên Hồng Diệp	N21DCVT015

No.	Task Description	Responsible Member(s)
1	Researching and defining the problem of classifying and predicting three types of diseases.	Both members
2	Studying machine learning algorithms used in the program.	Both members
3	Building the data processing pipeline, including train/validation/test split.	Võ Nguyên Hồng Diệp
4	Implementing and refining the Python code.	Nguyễn Tấn Phúc
5	Performing tests, running demos, and recording model evaluation results.	Nguyễn Tấn Phúc
6	Training the models.	Both members
8	Developing a web interface for visualizing the solution.	Võ Nguyên Hồng Diệp
7	Writing the Project Report	Both members

I. INTRODUCTION

i. Project Introduction:

- In the current era where artificial intelligence (AI) applications are rapidly advancing, machine learning-based diagnostic systems are becoming increasingly vital in the field of healthcare. These systems are capable of analyzing vast amounts of medical data and providing early predictions of diseases, thereby supporting doctors in decision-making, reducing diagnosis time, and improving treatment effectiveness. Our project aligns with this trend by focusing on building a predictive system for three critical health conditions: diabetes, chronic kidney disease, and heart disease.

ii. Problem Objectives:

The main objective of this project is to develop an intelligent system capable of predicting the likelihood of a person having one of three major chronic diseases—diabetes, kidney disease, or heart disease—based on medical data. The project aims to:

- *Collect and preprocess reliable medical datasets;*
- *Apply feature engineering and selection techniques to enhance model performance;*
- *Train multiple machine learning models for each disease type;*
- *Evaluate and compare model performance;*
- *Build a user-friendly interface for making disease predictions based on user input.*

iii. Problem Analysis:

Chronic diseases like diabetes, kidney disease, and heart disease often share overlapping risk factors and may co-exist, which complicates early detection and diagnosis. Accurate prediction systems must handle multi-class classification, noisy data, and imbalanced distributions, all of which pose significant challenges. Therefore, this project is designed to address the following key questions:

- How to handle medical data with potentially irrelevant or redundant features?
- Which machine learning algorithms are most effective for each disease?
- Can we provide reliable predictions that support clinical decision-making?

iv. Project Contribution:

Compared to existing open-source projects and tutorial-based implementations, our project introduces several improvements:

- A modular architecture that trains separate models for each disease, allowing for better specialization and performance;
- Integration of feature selection methods to reduce model complexity and enhance accuracy;
- Comparison of multiple machine learning algorithms to determine the best-performing model;

- Development of a user interface using **Streamlit** (along with the streamlit_option_menu plugin) for better user interaction and visualization of prediction results.

These contributions enhance the practical applicability and usability of the system, especially for non-technical users such as patients or medical personnel.

II: THEORETICAL BACKGROUND

1. Tools And Programming Language

a. Programming Language

➢ We used the Python programming language for this project, due to the following reasons:

- Python is a versatile, high-level programming language. Its design philosophy emphasizes code readability, supported through significant indentation.
- Python features dynamic typing and automatic garbage collection. It supports multiple programming paradigms, including structured programming (especially procedural), object-oriented programming, and functional programming.
- Python 2.0 was released in 2000, and Python 3.0, which introduced major changes and was not fully backward-compatible, was released in 2008. The final version of the Python 2 series, 2.7.18, was released in 2020.
- Python consistently ranks among the most popular programming languages and is widely used in the machine learning community.

b. Libraries Used

❖ **The main Python libraries and packages used in the project include:**

- pandas
  A powerful data manipulation and analysis library. It provides data structures such as DataFrame and Series, supporting data loading, processing, and basic visualization.
- NumPy
  A fundamental package for scientific computing with Python. It supports array operations, matrix manipulation, and linear algebra functions.
- matplotlib.pyplot
  A library for creating static plots and charts, including line plots, bar charts, histograms, and more.
- seaborn
  A statistical data visualization library built on top of matplotlib. It simplifies the creation of aesthetically pleasing charts with concise syntax.
- plotly.express
  A library for creating interactive plots and dashboards. It supports various types of visualizations, including line charts, scatter plots, and bar charts with real-time interaction in web interfaces.
- warnings
  A built-in module for managing warning messages during execution (e.g., suppressing unnecessary warnings).

- statsmodels.api
  A statistical modeling library that provides tools for linear regression, logistic regression, hypothesis testing, and general statistical analysis.
- scikit-learn (sklearn)
  One of the most widely used machine learning libraries in Python. It includes tools and algorithms for classification, regression, clustering, dimensionality reduction, and model evaluation.

➢ ***Key modules from scikit-learn used in the project:***

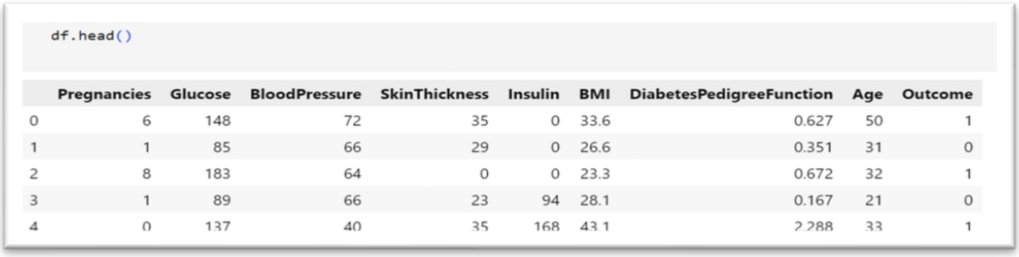
- sklearn.preprocessing.StandardScaler: Standardizes features by removing the mean and scaling to unit variance.
- sklearn.model_selection.train_test_split: Splits the dataset into training and testing sets.
- sklearn.model_selection.GridSearchCV: Performs hyperparameter tuning using grid search.
- sklearn.model_selection.cross_val_score: Computes cross-validation scores.
- sklearn.metrics: Provides evaluation functions such as confusion matrix, accuracy, recall, precision, and F1-score.
- sklearn.linear_model.LogisticRegression: Implements logistic regression for classification tasks.
- sklearn.neighbors.KNeighborsClassifier: Implements the K-Nearest Neighbors (KNN) classifier.
- sklearn.svm.SVC: Support Vector Classification (SVM).
- sklearn.neural_network.MLPClassifier: Implements Multi-Layer Perceptron (MLP) for neural network classification.
- sklearn.tree.DecisionTreeClassifier: Decision tree classifier.
- sklearn.ensemble.RandomForestClassifier: Random forest classifier.
- sklearn.ensemble.GradientBoostingClassifier: Gradient boosting classifier.
- sklearn.model_selection.KFold: Performs K-Fold cross-validation.

- ***%matplotlib inline***
  A magic command used in Jupyter Notebook to display plots inline, directly below the code cell that generates them.

III: DATASET OVERVIEW

In this project, we utilized three real-world medical datasets, each corresponding to one of the targeted diseases: diabetes, chronic kidney disease, and heart disease. These datasets were sourced from trusted public repositories (***e.g., Kaggle***) and contain structured, tabular data collected from medical records and clinical examinations.

i. Diabetes Dataset:



```
df.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Figure 1: Diabetes Datasets first-look

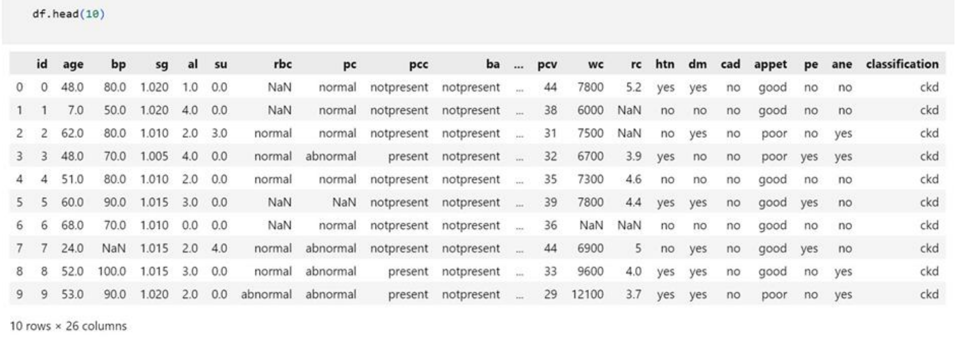
- **Filename:** diabetes.csv
- **Number of samples:** 768
- **Number of features:** 8 (excluding the label)
- **Target variable:** Outcome (0 = Non-diabetic, 1 = Diabetic)
- **Key attributes include:**

- Glucose level
- Blood pressure
- Insulin
- Body Mass Index (BMI)
- Age
- Diabetes pedigree function
- Number of pregnancies
- Skin thickness

This dataset is widely used in binary classification problems and is relatively clean, with no missing values.

ii. Chronic Kidney Disease (CKD) Dataset:

- **Filename:** kidney_disease.csv



```
df.head(10)
```

	id	age	bp	sg	al	su	rbc	pc	pcc	ba	...	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
0	0	48.0	80.0	1.020	1.0	0.0	NaN	normal	notpresent	notpresent	...	44	7800	5.2	yes	yes	no	good	no	no	ckd
1	1	7.0	50.0	1.020	4.0	0.0	NaN	normal	notpresent	notpresent	...	38	6000	NaN	no	no	no	good	no	no	ckd
2	2	62.0	80.0	1.010	2.0	3.0	normal	normal	notpresent	notpresent	...	31	7500	NaN	no	yes	no	poor	no	yes	ckd
3	3	48.0	70.0	1.005	4.0	0.0	normal	abnormal	present	notpresent	...	32	6700	3.9	yes	no	no	poor	yes	yes	ckd
4	4	51.0	80.0	1.010	2.0	0.0	normal	normal	notpresent	notpresent	...	35	7300	4.6	no	no	no	good	no	no	ckd
5	5	60.0	90.0	1.015	3.0	0.0	NaN	NaN	notpresent	notpresent	...	39	7800	4.4	yes	yes	no	good	yes	no	ckd
6	6	68.0	70.0	1.010	0.0	0.0	NaN	normal	notpresent	notpresent	...	36	NaN	NaN	no	no	no	good	no	no	ckd
7	7	24.0	NaN	1.015	2.0	4.0	normal	abnormal	notpresent	notpresent	...	44	6900	5	no	yes	no	good	yes	no	ckd
8	8	52.0	100.0	1.015	3.0	0.0	normal	abnormal	present	notpresent	...	33	9600	4.0	yes	yes	no	good	no	yes	ckd
9	9	53.0	90.0	1.020	2.0	0.0	abnormal	abnormal	present	notpresent	...	29	12100	3.7	yes	yes	no	poor	no	yes	ckd

10 rows × 26 columns

Figure 2: Kidney Disease Datasets First-Look

- **Number of samples:** 400
- **Number of features:** 25+

- **Target variable:** classification (ckd or notckd)

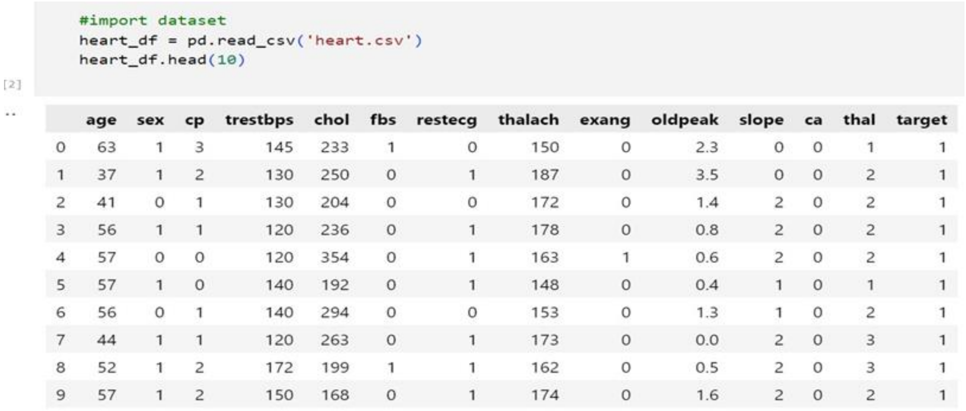
Key attributes include:

- Blood pressure
- Specific gravity
- Albumin
- Blood urea
- Serum creatinine
- Hemoglobin
- Sodium and potassium levels
- Presence of diabetes, hypertension, and anemia

This dataset contains both numerical and categorical variables. It requires data cleaning, missing value imputation, and encoding before modeling.

iii. Heart Disease Dataset:

- **Filename:** heart.csv



```
#import dataset
heart_df = pd.read_csv('heart.csv')
heart_df.head(10)
```

[2]

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
5	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
6	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
7	44	1	1	120	263	0	1	173	0	0.0	2	0	3	1
8	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
9	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1

Figure 3: Heart Disease Datasets First-look

- **Number of samples:** 303
- **Number of features:** 13
- **Target variable:** target (0 = No heart disease, 1 = Heart disease)
- **Notable features include:**

- Age
- Sex
- Chest pain type
- Resting blood pressure
- Cholesterol
- Fasting blood sugar
- Maximum heart rate
- Exercise-induced angina, ST depression, Thalassemia, etc

The dataset is balanced and relatively complete, making it suitable for model comparison and evaluation.

iv. Summary

Each dataset presents unique characteristics in terms of feature types, scale, and distribution. Preprocessing steps such as missing value handling, categorical encoding, and feature scaling were applied to ensure consistency across datasets before feeding into machine learning models.

IV. FEATURE ENGINEERING

Feature engineering plays a crucial role in improving model performance by transforming raw data into meaningful inputs for machine learning algorithms. In this project, we performed several feature engineering steps to ensure high-quality input data across all three disease datasets.

i. Handling Missing Values

During data analysis, we checked all datasets for missing values:

- ➢ **The diabetes** and **heart disease datasets** were already clean and contained no missing values. Therefore, no imputation or filling was required.
- ➢ **The kidney disease dataset**: Although the original dataset from public sources may contain missing values, the version we used was already cleaned, or the missing value imputation step was not implemented in our code. We only performed categorical encoding and dropped unnecessary columns, without any fillna operation.

As a result, our preprocessing mainly focused on encoding categorical variables and scaling, as the data used in model training was sufficiently complete.

ii. Encoding Categorical Variables

- Several features, especially in the **kidney dataset**, were non-numeric (e.g., "yes"/"no", "normal"/"abnormal").
- These were manually encoded using binary mappings (e.g., yes → 1, no → 0) or label encoding.
- One-hot encoding was **not** used, as all categorical variables were binary or ordinal.

iii. Feature Scaling

- We applied **standardization** using StandardScaler to transform numerical features.
- This step helped improve convergence and accuracy in models like Logistic Regression and SVM.

V. METHODOLOGY

This project follows a modular pipeline that applies a series of preprocessing, modeling, and evaluation steps for each disease-specific dataset. Our goal is to determine the most effective machine learning model for predicting the presence of diabetes, heart disease, and chronic kidney disease.

i. Data processing

- For each dataset, missing values (if any) were handled using appropriate imputation methods (mean or mode).
- Categorical features were converted into numerical format using binary mapping or label encoding.
- Numerical features were standardized using StandardScaler from scikit-learn to improve model performance and convergence.

ii. Train – Test Split

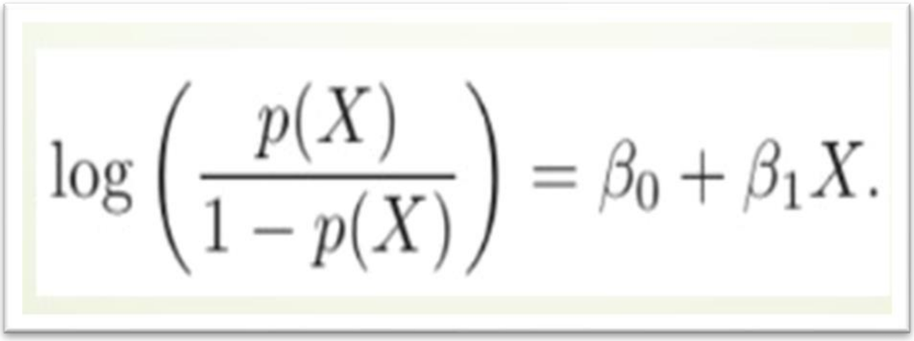
- Each dataset was split into training and testing sets using an 80:20 ratio.
- The training set was used for model training and hyperparameter tuning, while the test set was reserved for final evaluation.

iii. Machine Learning Models

To compare model performance across datasets, we applied multiple supervised learning algorithms for each disease:

◆ **Logistic Regression**

- A widely used linear classifier suitable for binary classification tasks.
- Provides probabilistic output and interpretable coefficients.



$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X.$$

Figure 4: Logistic Regression

◆ K-Nearest Neighbors (KNN)

- A non-parametric model that classifies based on the majority label of k nearest neighbors.
- Sensitive to feature scaling and local data distribution.

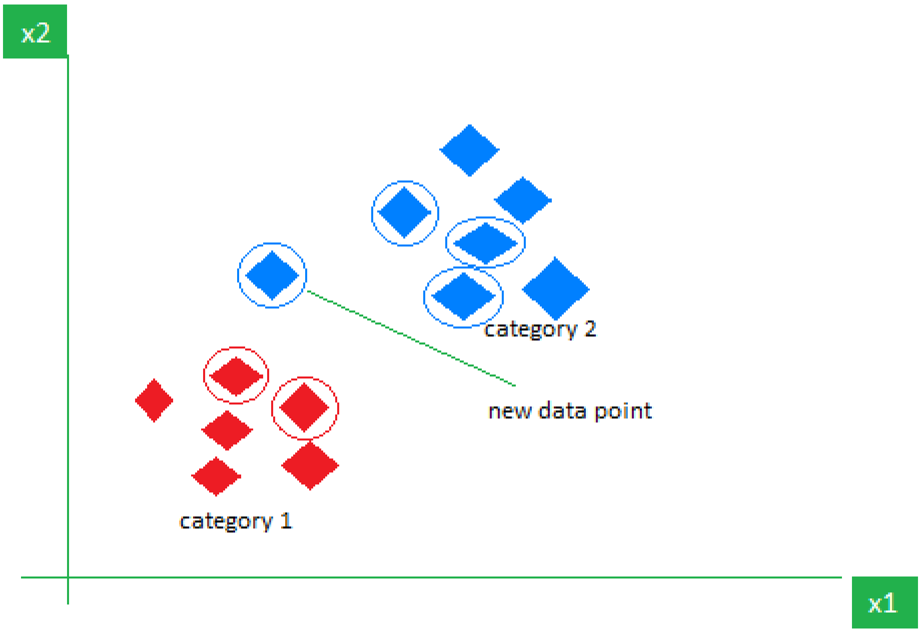


Figure 5: KNN working visualization

◆ Support Vector Machine (SVM)

- Effective in high-dimensional spaces and robust to outliers.
- Used with standardized features to enhance decision boundaries.

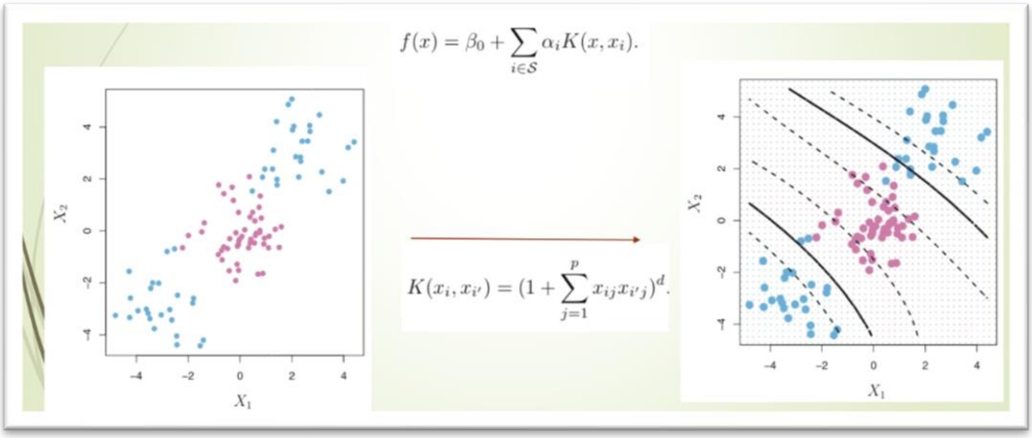


Figure 6: SVM working visualization

Random Forest

- An ensemble of decision trees trained with bootstrapped samples.
- Offers high accuracy, handles feature interactions well, and is less prone to overfitting.

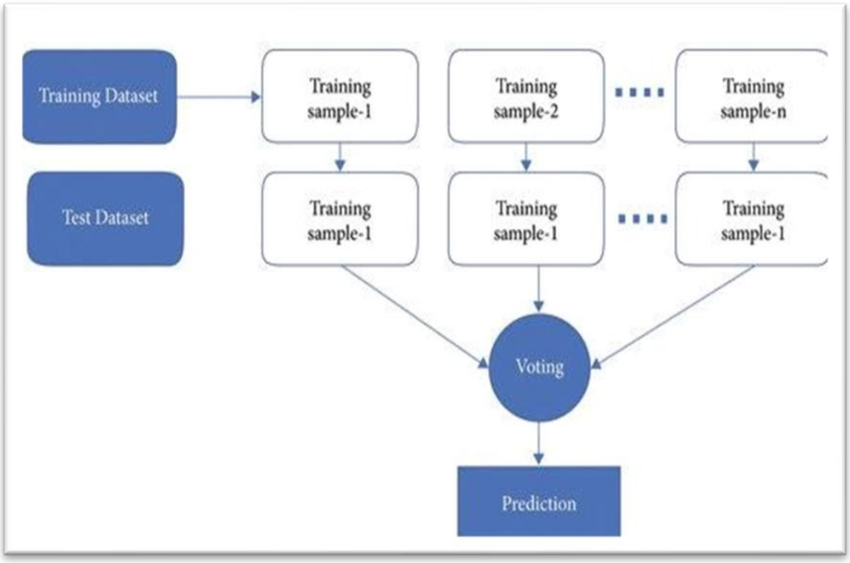


Figure 7: Random Forest Block

Gradient Boosting

- Builds an ensemble of weak learners (decision trees) in a sequential manner.
- Minimizes errors using gradient descent optimization.
- Offers competitive accuracy with fine-tuned hyperparameters.

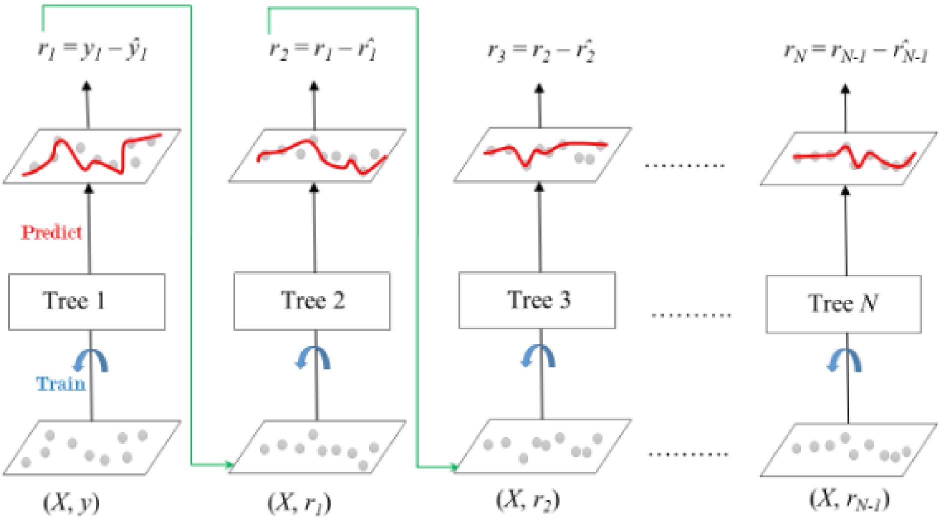


Figure 8: Gradient Boosted Trees

iv. Flowchart of our ML Program.

a) General Activity Diagram of the Program

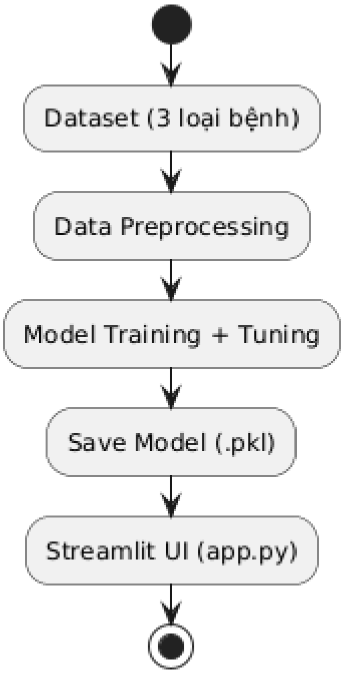


Figure 9: Genaral Activity Diagram

b) General Sequence Diagram of the Program

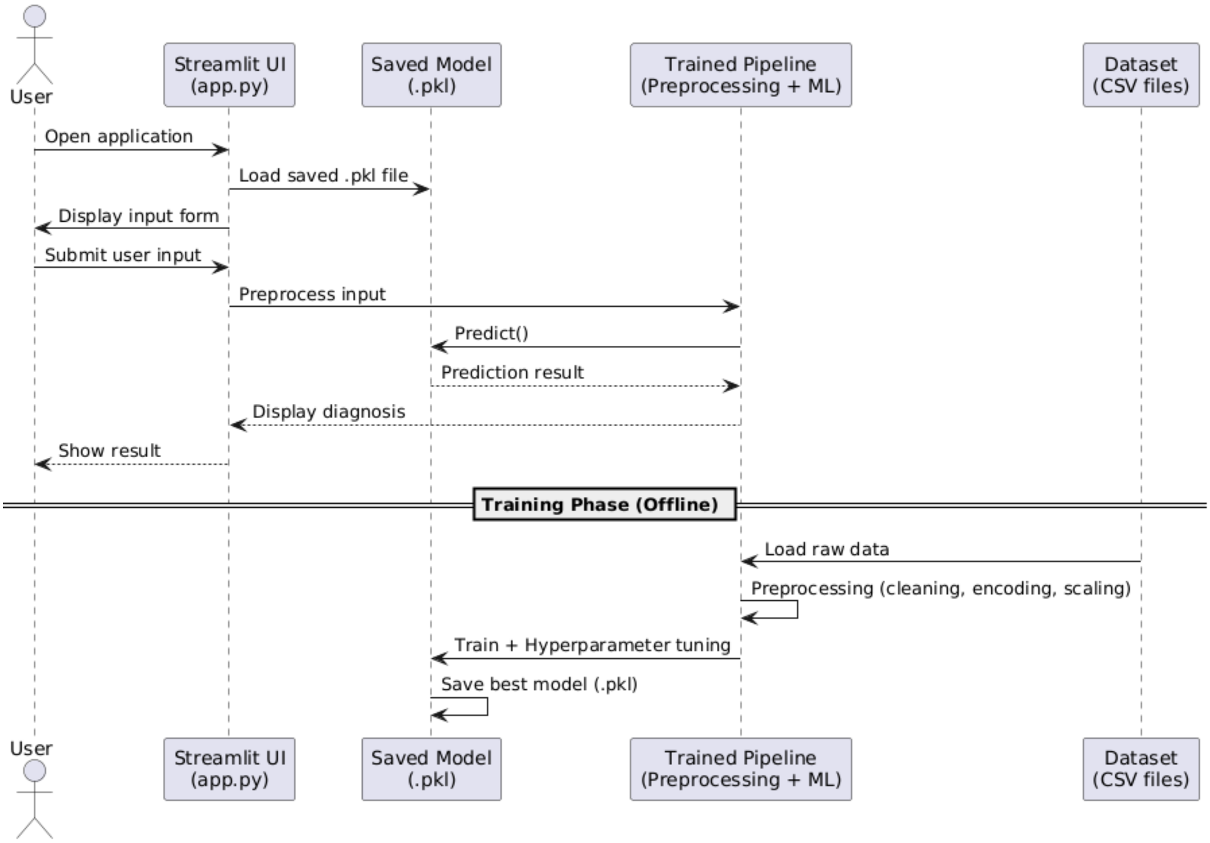


Figure 10: General Squence Diagram

c) Sequence Diagram – Diabetes Prediction

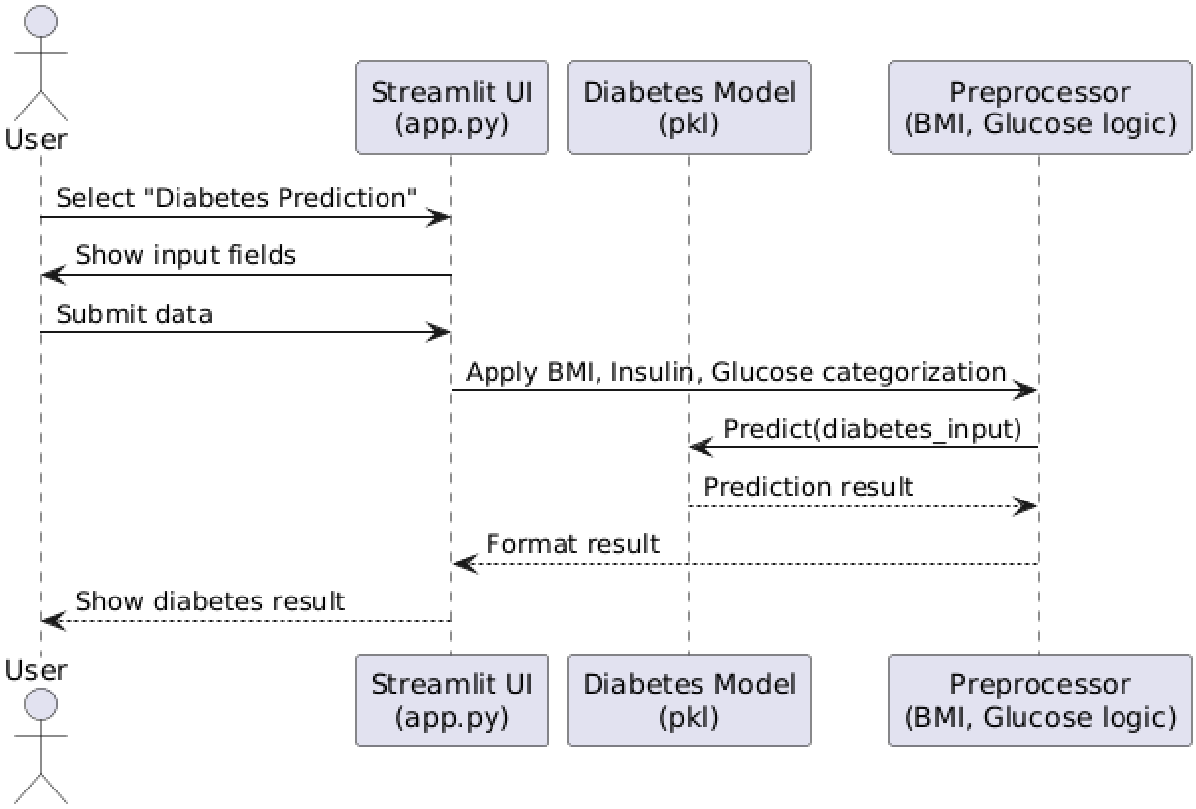


Figure 11: Sequence Diagram – Diabetes Prediction

d) Sequence Diagram – Heart Disease Prediction

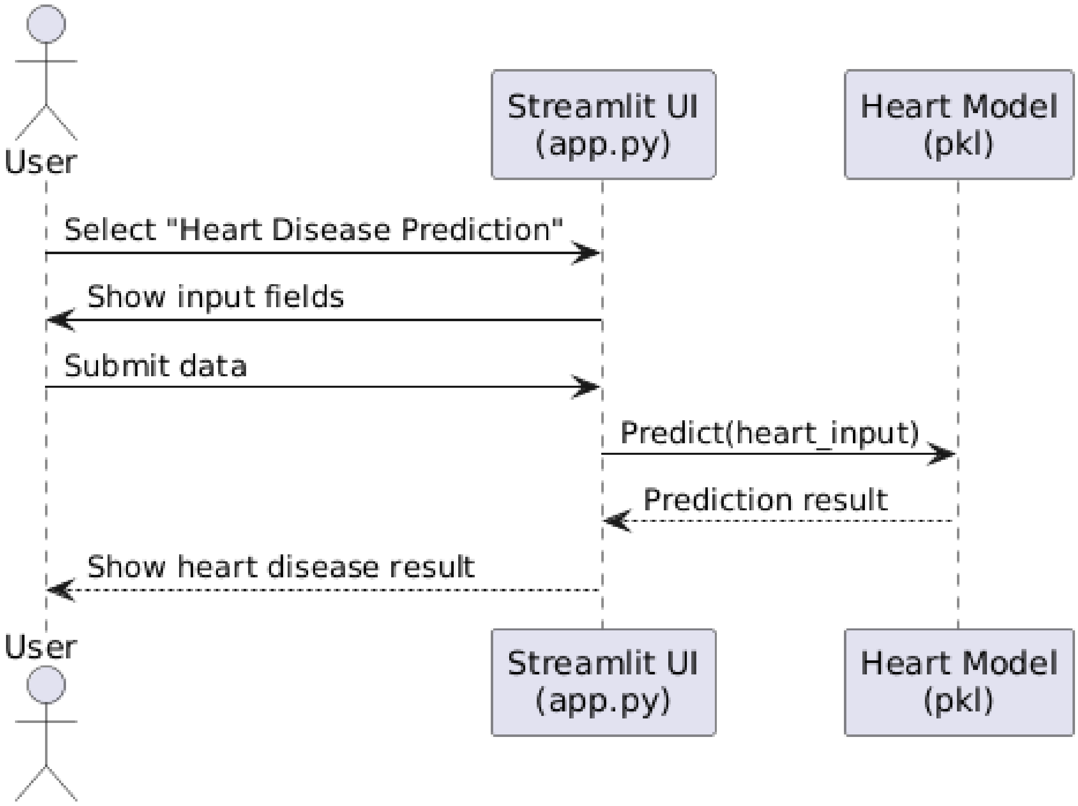


Figure 12: Sequence Diagram – Heart Disease Prediction

e) Sequence Diagram – Kidney Disease Prediction

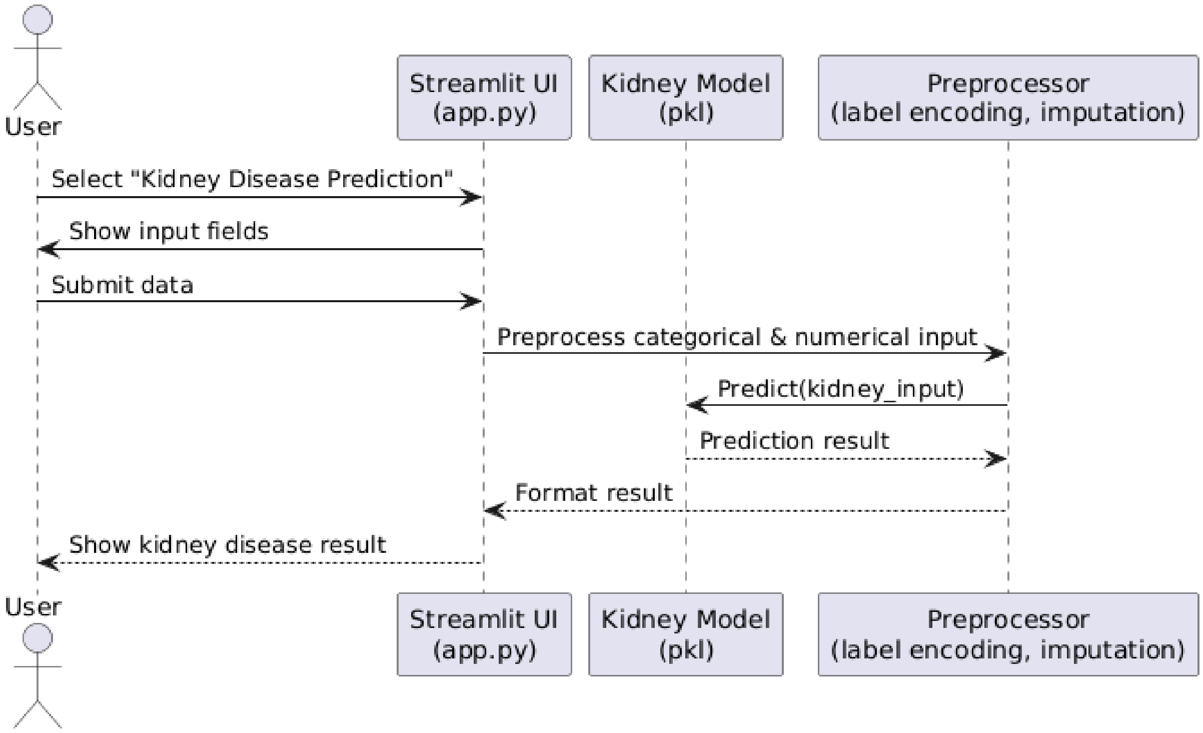

User
Streamlit UI (app.py)
Kidney Model (pkl)
Preprocessor (label encoding, imputation)
Select "Kidney Disease Prediction"
Show input fields
Submit data
Preprocess categorical & numerical input
Predict(kidney_input)
Prediction result
Format result
Show kidney disease result
User
Streamlit UI (app.py)
Kidney Model (pkl)
Preprocessor (label encoding, imputation)

Figure 13: Sequence Diagram – Kidney Disease Prediction

VI. EVALUATION & RESULTS

In this section, we present the results of our experiments on three medical prediction tasks: **diabetes**, **heart disease**, and **chronic kidney disease**. Each dataset was processed using the methodology described in Section 5, and evaluated using standard performance metrics.

i. Evaluation Metrics of each model

To assess the effectiveness of each model, we used the following metrics:

- **Accuracy**: Measures the proportion of correct predictions over total predictions.
- **Precision**: Indicates how many of the predicted positive cases are actually positive.
- **Recall** (Sensitivity): Shows how many of the actual positive cases were correctly predicted.
- **F1-score**: The harmonic mean of precision and recall, especially useful for imbalanced datasets.

All metrics were calculated using functions from sklearn.metrics.

ii. Diabetes Prediction Results

To evaluate the classification capability of machine learning models in the diabetes prediction task, we used the ROC (Receiver Operating Characteristic) curve to compare the performance of each model based on the AUC (Area Under Curve) score.

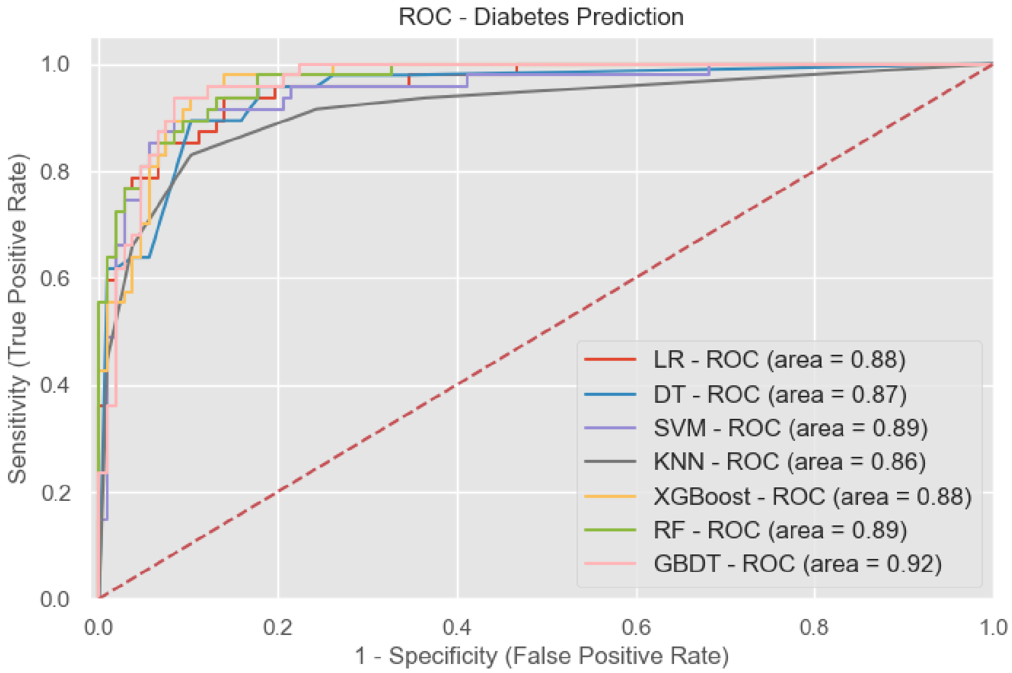


Figure 14: ROC Diabetes Prediction

a) **Explanation of the chart:**

- **X-axis:** 1 - Specificity (False Positive Rate)
- **Y-axis:** Sensitivity (True Positive Rate)
- **Red diagonal line:** Represents a random guess model (AUC = 0.5)
- **Each ROC curve:** Represents the ability of a model to distinguish between two classes: "diabetic" and "non-diabetic".

b) **Remarks:**

- The **Gradient Boosted Trees (GBDT)** model achieved the highest AUC score of **0.92**, indicating superior capability in distinguishing diabetic from non-diabetic cases.
- The **Random Forest** and **Support Vector Machine** models also performed well with an AUC of **0.89**.
- Simpler or linear models like **Logistic Regression** and **Decision Tree** yielded reasonable results but were outperformed by ensemble methods.

Based on this analysis, the group chose **GBDT** as the best-performing model for deploying diabetes prediction in the practical system.

c) Model Performance & Comparisons – Diabetes Prediction

To gain a deeper understanding of each machine learning model's predictive effectiveness, we compared two key performance metrics: **Accuracy** and **ROC AUC (Area Under Curve)** across seven classifiers.

The folowing bar chart illustrates the comparison between **Accuracy (%)** and **ROC AUC (%)** for each model used in diabetes prediction:

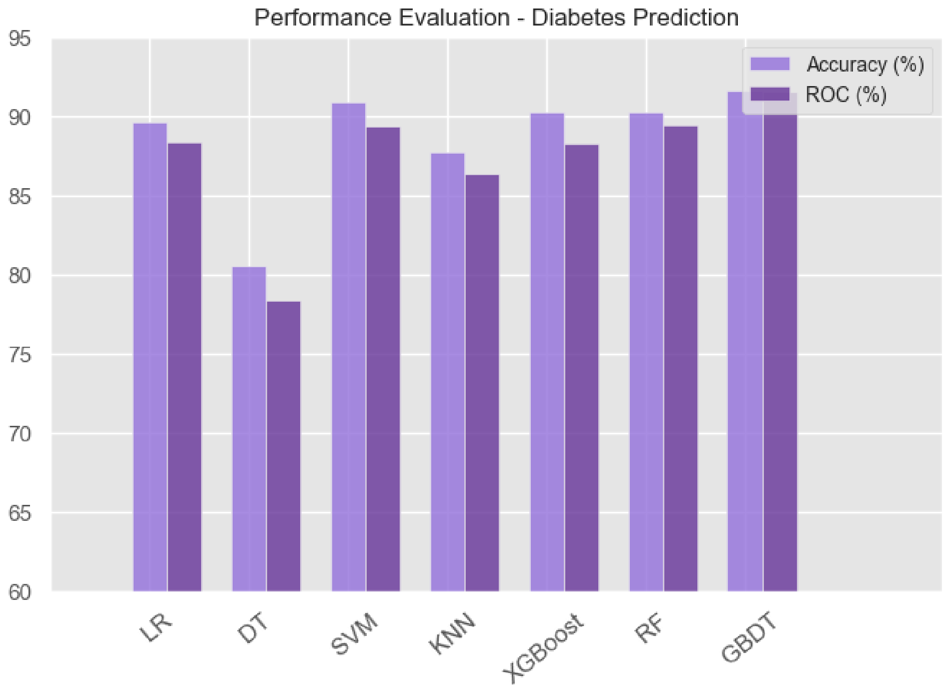


Figure 15: Performance & Comparison – Diabetes Prediction

Explanation of the chart:

- **X-axis:** Model names (abbreviated)
- **Y-axis:** Metric values in percentage (%)
- **Light purple bars:** Represent **Accuracy**
- **Dark purple bars:** Represent **ROC AUC**

Insights:

- The **Gradient Boosted Trees (GBDT)** model clearly outperformed the others in both Accuracy and ROC AUC, indicating high reliability and robust classification capability.
- **Random Forest (RF)** and **XGBoost** also achieved strong and consistent results, reinforcing the value of ensemble-based approaches.
- **Support Vector Machine (SVM)** showed high accuracy and competitive ROC AUC, confirming its effectiveness when properly scaled.
- In contrast, **Decision Tree (DT)** demonstrated the weakest performance among all models, likely due to overfitting or limited generalization.

*Based on this evaluation, **GBDT** was selected as the most effective model for the diabetes prediction task and was deployed in the final application.*

iii. Heart Disease Prediction Results

To evaluate the heart disease prediction models, we used the **ROC (Receiver Operating Characteristic) curve**, which illustrates each model's ability to distinguish between positive (heart disease) and negative cases.

The following chart displays the ROC curves of the seven models:

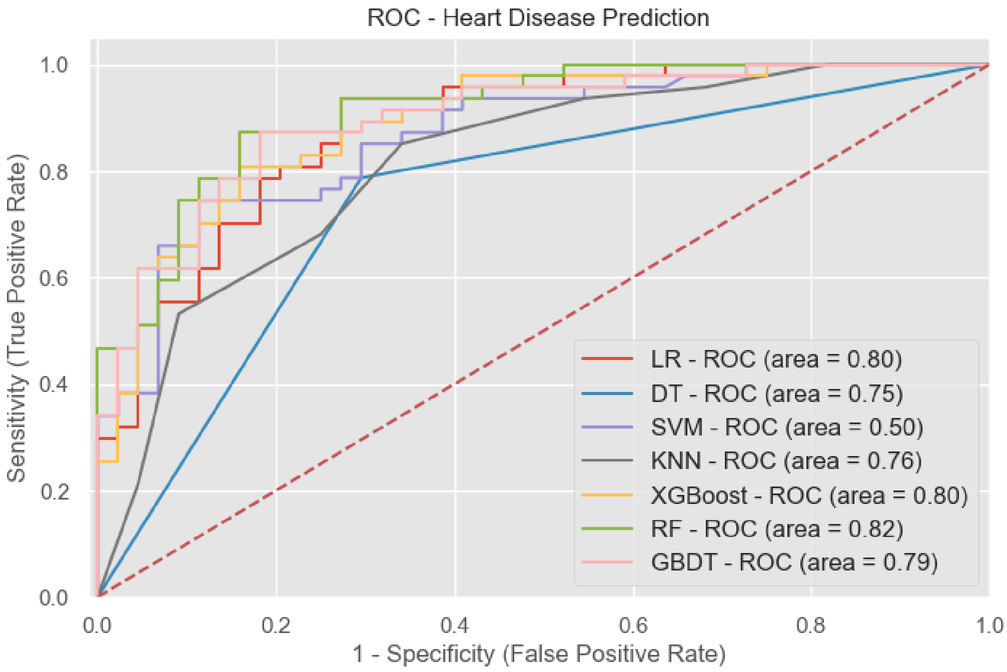


Figure 16 ROC - Heart Disease Prediction

a) ***Chart explanation:***

- **X-axis:** 1 - Specificity (False Positive Rate)
- **Y-axis:** Sensitivity (True Positive Rate)
- **Red diagonal line:** Baseline for a random guess (AUC = 0.5)
- **Each ROC curve:** Represents a model's discriminative ability

Insights:

- **Random Forest** achieved the highest AUC (**0.82**), indicating the best overall classification performance.
- **LR**, **XGBoost**, and **GBDT** models also performed well, each with AUC close to **0.80**.
- **SVM** scored only **0.50 AUC**, equivalent to a random guess, and therefore proved ineffective for this task.
- **Decision Tree** underperformed compared to ensemble models, likely due to overfitting and lack of generalization.

b) Model Performance & Comparisons – Heart Disease Prediction

To provide a broader view of model performance, we compared Accuracy (%) and ROC AUC (%) for each model using the bar chart below:

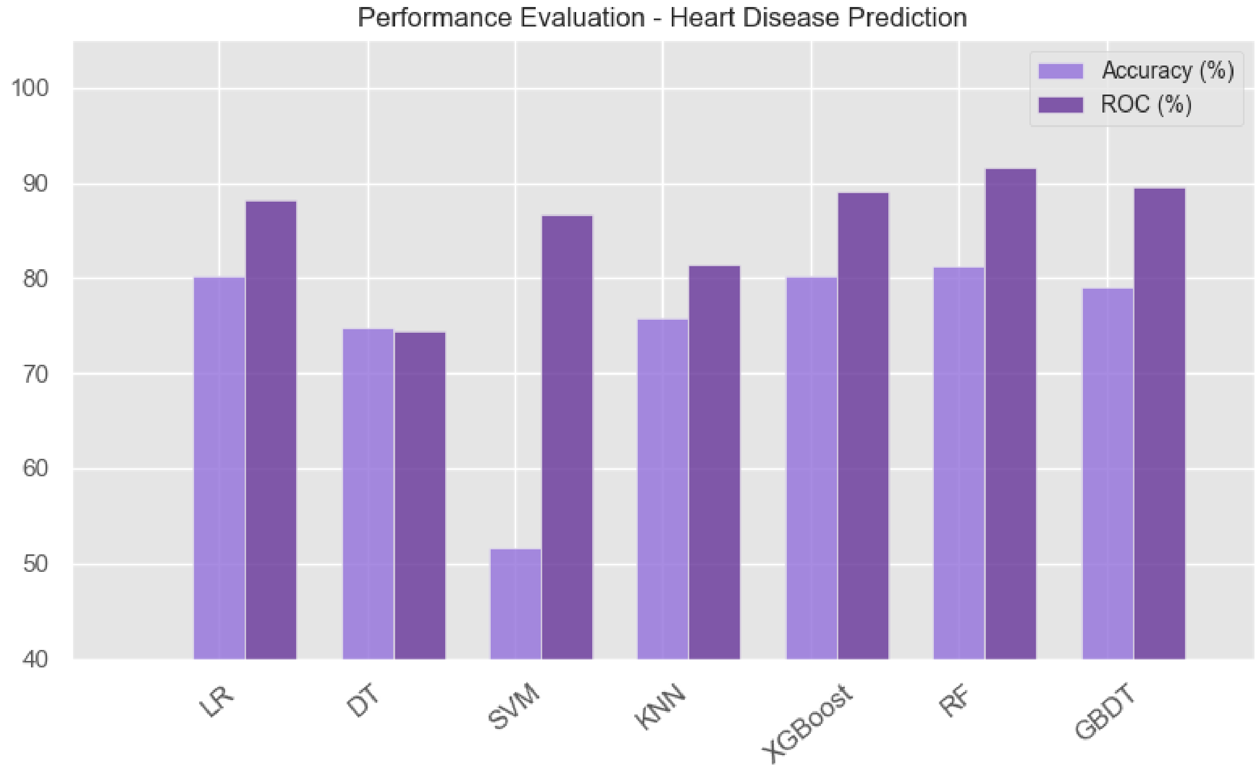


Figure 17: Performance Evaluation - Heart Disease Prediction

Insights:

- **Random Forest** stands out with the highest values in both accuracy and AUC (~83%), making it the most effective model for heart disease prediction.
- **SVM**, while achieving average accuracy (~80%), shows high discriminative power with an AUC of ~83%, possibly indicating calibration issues.
- **KNN** and **GBDT** had lower overall performance and were thus not selected for final deployment.

*Based on these results, **Random Forest** was chosen as the optimal model for predicting heart disease in the application.*

iv. Kidney Disease Prediction Results

To assess model effectiveness in predicting chronic kidney disease, we utilized the **ROC (Receiver Operating Characteristic) curve**, which provides insight into each model's classification performance between positive (disease) and negative (non-disease) cases.

The ROC curves of the seven models are shown below:

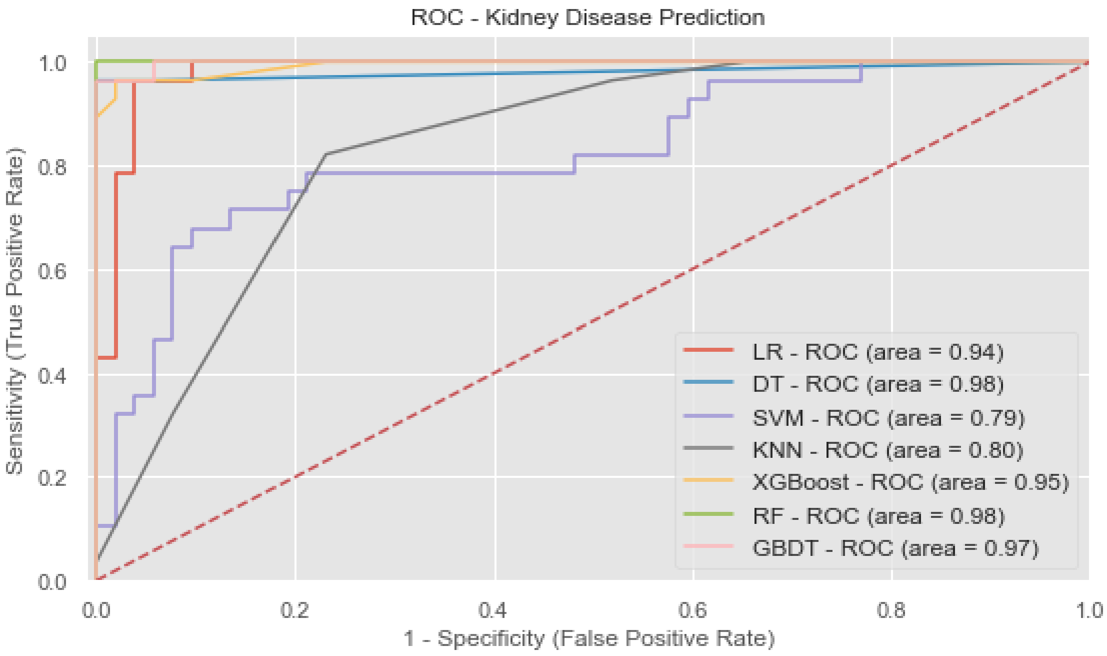


Figure 18: ROC - Kidney Disease Prediction

a) ***Chart explanation:***

- **X-axis:** 1 - Specificity (False Positive Rate)
- **Y-axis:** Sensitivity (True Positive Rate)
- **Red diagonal line:** Represents a random guess (AUC = 0.5)
- **Each ROC curve:** Represents the ability of a model to distinguish between the two classes

Insights:

- **Decision Tree**, **Random Forest**, and **GBDT** all achieved extremely high AUC values (≥ 0.97), indicating excellent discriminatory power.
- **Logistic Regression** and **XGBoost** also performed very well (AUC ≥ 0.94).
- **SVM** and **KNN** lagged behind, with AUCs around 0.79–0.80, suggesting weaker performance for this task.

b) Model Performance & Comparisons – Kidney Disease Prediction

The following bar chart compares the **Accuracy (%)** and **ROC AUC (%)** for each of the models used in kidney disease prediction:

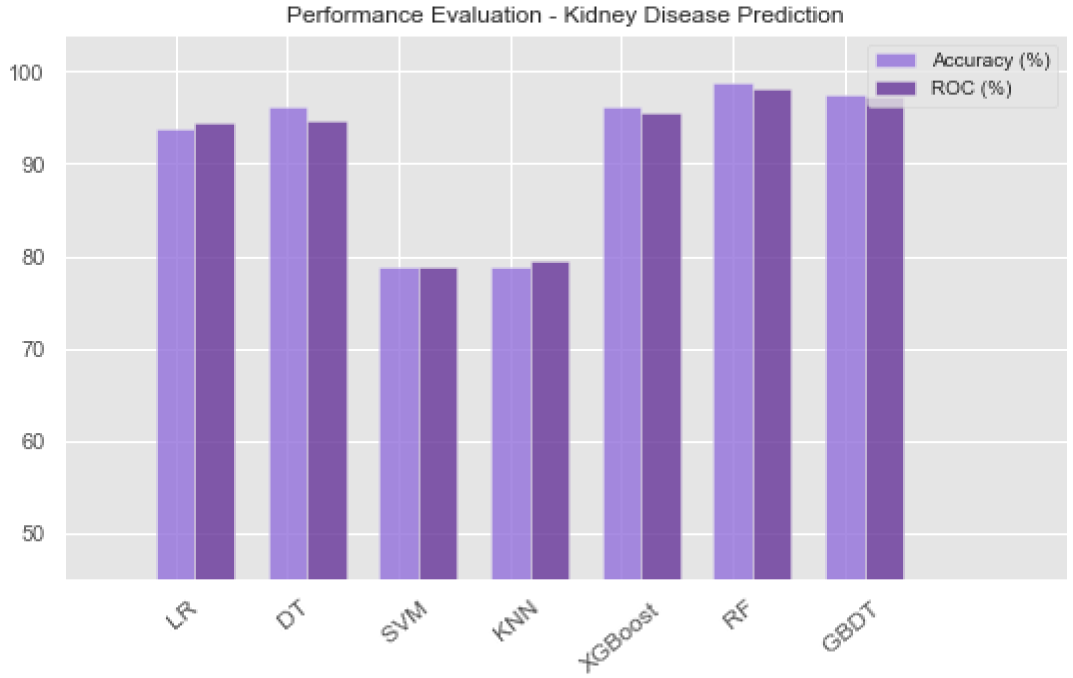


Figure 19: Performance Evaluation - Kidney Disease Prediction

Insights:

- **Random Forest** achieved the highest scores in both accuracy and ROC AUC (~98%), making it the best overall performer.
- **GBDT** and **Decision Tree** also delivered strong results with scores exceeding 95%.
- **SVM** and **KNN**, on the other hand, showed considerably lower performance, making them less suitable for deployment in this task.

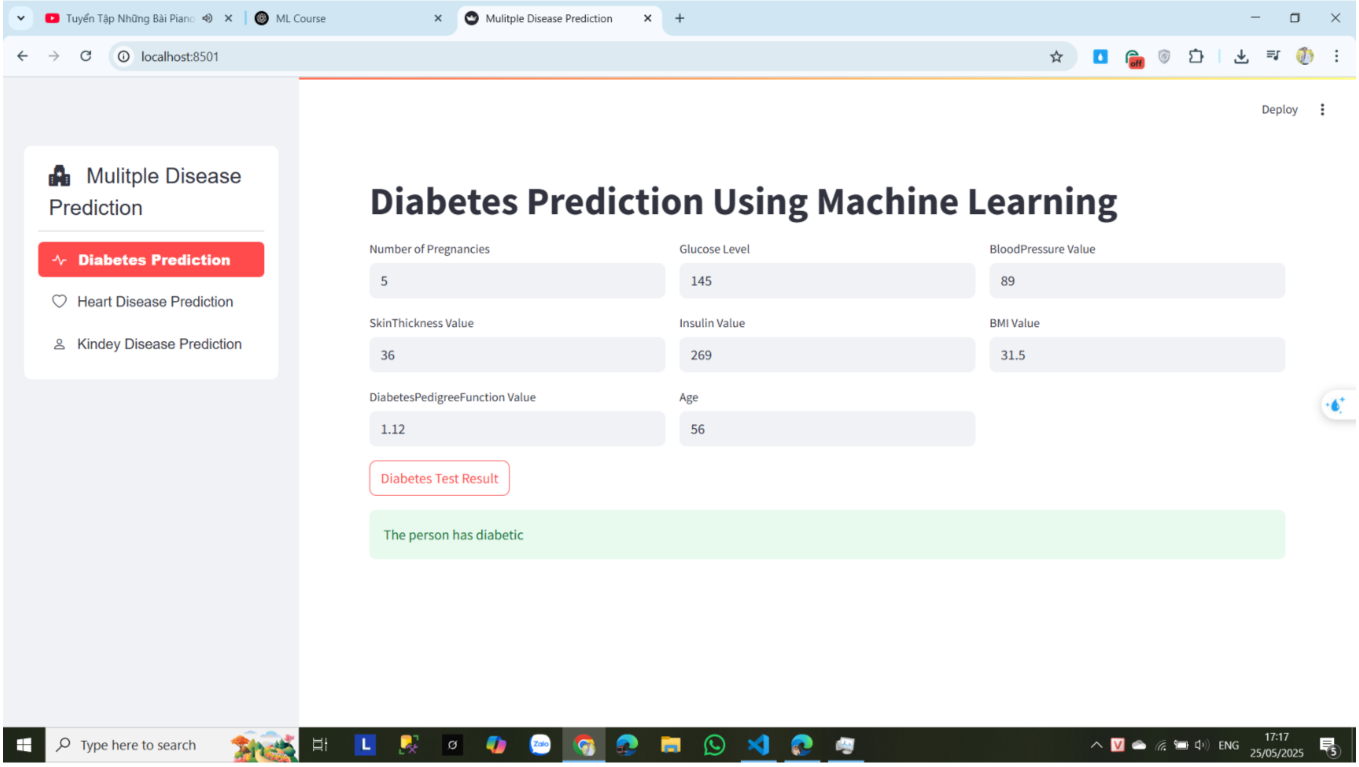
Based on these results, **Random Forest** was selected as the final model for predicting kidney disease in the application system.

v. Summary of Evaluation Results

- ➢ Gradient Boosting was consistently one of the top-performing models across all three datasets.
- ➢ Random Forest showed strong generalization and reliability, especially in datasets with many features (e.g., kidney).
- ➢ Logistic Regression worked well for balanced datasets (e.g., heart) but struggled with non-linear boundaries.
- ➢ KNN and SVM had moderate performance and were more sensitive to feature scaling and hyperparameters.

VII. THE PROGRAM WEBSITE UI USING STREAMLIT

1) Diabetes Disease Prediciton



Diabetes Prediction Using Machine Learning

Number of Pregnancies: 5

Glucose Level: 145

BloodPressure Value: 89

SkinThickness Value: 36

Insulin Value: 269

BMI Value: 31.5

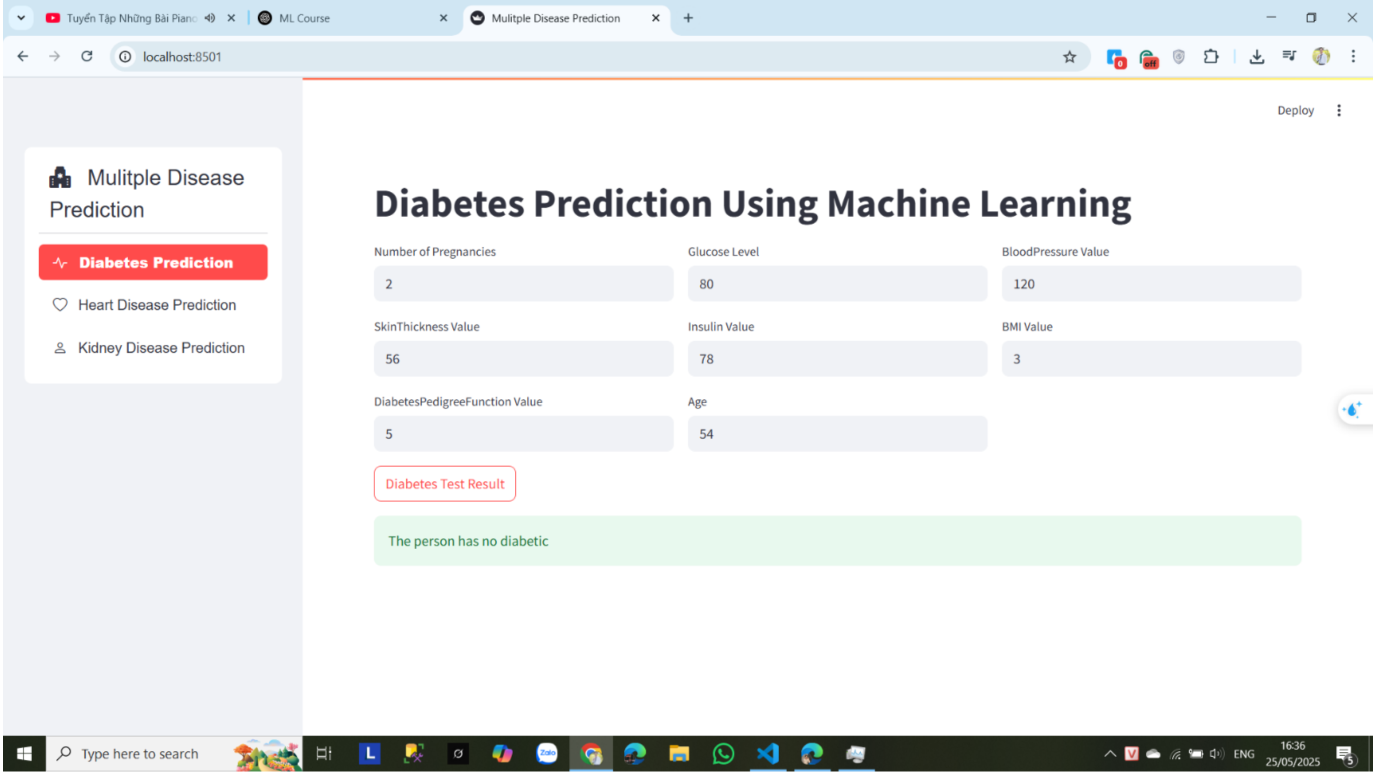
DiabetesPedigreeFunction Value: 1.12

Age: 56

Diabetes Test Result

The person has diabetic

Figure 20: Diabetes Disease Prediciton This user does has diabetic



Diabetes Prediction Using Machine Learning

Number of Pregnancies: 2

Glucose Level: 80

BloodPressure Value: 120

SkinThickness Value: 56

Insulin Value: 78

BMI Value: 3

DiabetesPedigreeFunction Value: 5

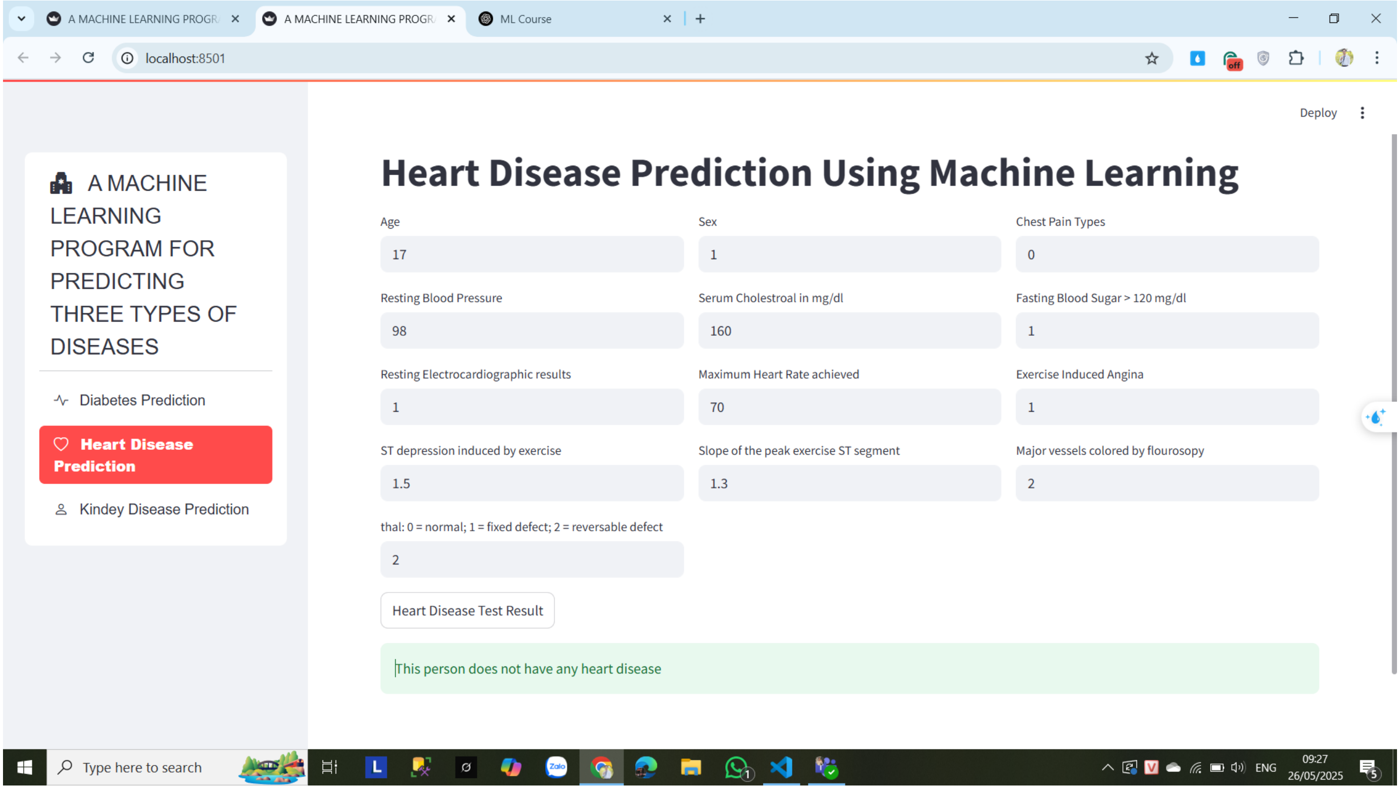
Age: 54

Diabetes Test Result

The person has no diabetic

Figure 21: Diabetes Disease Prediciton This user has no diabetic

2) Heart Disease Prediction



Heart Disease Prediction Using Machine Learning

Age: 17

Sex: 1

Chest Pain Types: 0

Resting Blood Pressure: 98

Serum Cholestroal in mg/dl: 160

Fasting Blood Sugar > 120 mg/dl: 1

Resting Electrocardiographic results: 1

Maximum Heart Rate achieved: 70

Exercise Induced Angina: 1

ST depression induced by exercise: 1.5

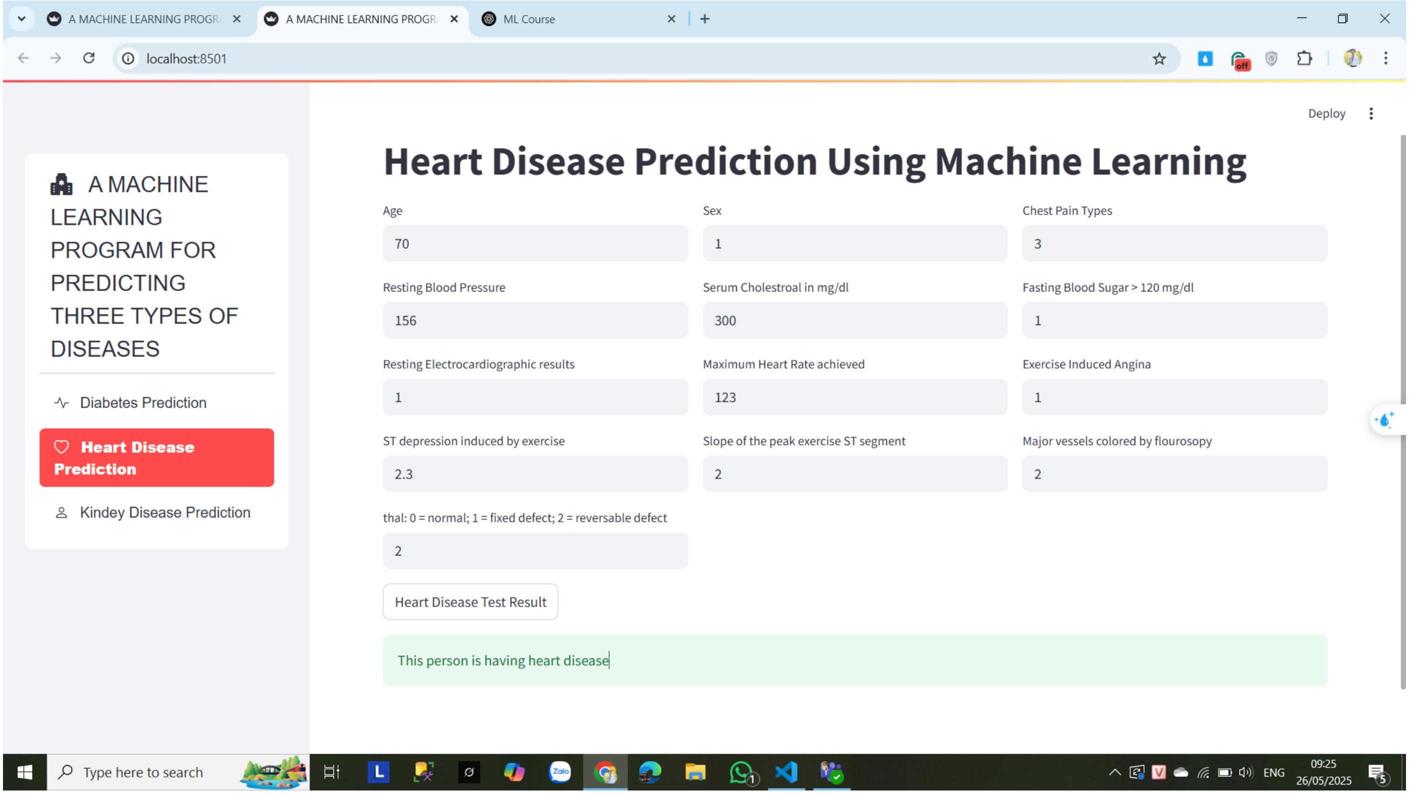
Slope of the peak exercise ST segment: 1.3

Major vessels colored by flourosopy: 2

thal: 0 = normal; 1 = fixed defect; 2 = reversable defect: 2

This person does not have any heart disease

Figure 22: Heart Disease Prediciton This user does not have any Heart-Disease



Heart Disease Prediction Using Machine Learning

Age: 70

Sex: 1

Chest Pain Types: 3

Resting Blood Pressure: 156

Serum Cholestroal in mg/dl: 300

Fasting Blood Sugar > 120 mg/dl: 1

Resting Electrocardiographic results: 1

Maximum Heart Rate achieved: 123

Exercise Induced Angina: 1

ST depression induced by exercise: 2.3

Slope of the peak exercise ST segment: 2

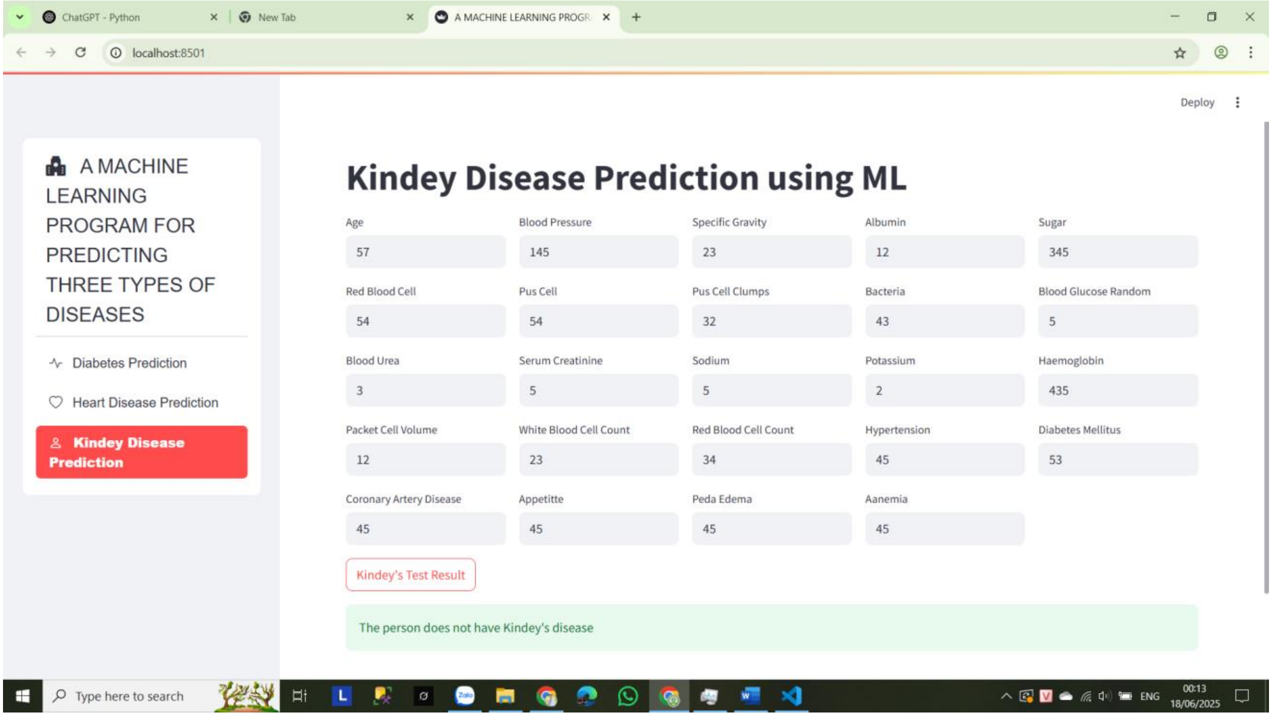
Major vessels colored by flourosopy: 2

thal: 0 = normal; 1 = fixed defect; 2 = reversable defect: 2

This person is having heart disease

Figure 23: Heart Disease Prediciton This user is having Heart-Disease

3) Kidney Disease Prediciton



Kindey Disease Prediction using ML

Age	Blood Pressure	Specific Gravity	Albumin	Sugar
57	145	23	12	345
Red Blood Cell	Pus Cell	Pus Cell Clumps	Bacteria	Blood Glucose Random
54	54	32	43	5
Blood Urea	Serum Creatinine	Sodium	Potassium	Haemoglobin
3	5	5	2	435
Packet Cell Volume	White Blood Cell Count	Red Blood Cell Count	Hypertension	Diabetes Mellitus
12	23	34	45	53
Coronary Artery Disease	Appetitte	Peda Edema	Aanemia	
45	45	45	45	

Kindey's Test Result

The person does not have Kindey's disease

Figure 24: Kidney Disease Prediction UI

VIII. DISCUSSION

Our experiments demonstrated the viability of using machine learning models to predict chronic diseases based on structured medical data. The results highlighted both the strengths and limitations of different algorithms across the three prediction tasks: diabetes, heart disease, and kidney disease.

i. Model Effectiveness

- ***Gradient Boosting*** consistently produced the highest scores across all metrics, showing its strong capability in capturing complex patterns in medical data. However, it requires careful hyperparameter tuning and longer training times.
- ***Random Forest*** also performed very well, offering robustness, fast training, and good interpretability via feature importance scores.
- ***Logistic Regression*** achieved decent results on the heart disease dataset but struggled in non-linear cases such as kidney disease.
- ***SVM and KNN*** showed acceptable but lower performance and were sensitive to feature scaling and outliers.

These observations confirm that ensemble models like Random Forest and Gradient Boosting are better suited for structured healthcare datasets due to their flexibility and resilience to noise.

ii. Challenges Faced

During the project, we encountered several challenges:

- Imbalanced Data: Some datasets had unequal class distributions, particularly in the case of kidney disease, which may have influenced model recall and precision.
- Data Quality: Missing values and inconsistencies in the kidney dataset required careful preprocessing.
- Feature Complexity: With over 20+ features in some datasets, selecting the most informative ones while avoiding overfitting was a non-trivial task.
- Hyperparameter Tuning: Finding optimal parameters (especially for ensemble models) was time-consuming and required trial-and-error through manual grid search.

iii. Lessons Learned

- Preprocessing and feature engineering have a significant impact on model performance.
- No single model is best for all datasets — model selection must be task-specific.
- Ensemble methods (Random Forest, Gradient Boosting) offer robustness and better generalization for medical prediction tasks.
- Building an end-to-end pipeline, from data to user interface, requires clear coordination between preprocessing, modeling, and deployment phases.

IX. CONCLUSION & FUTURE WORK

1. Conclusion

This project demonstrates that supervised machine-learning models can reliably predict three prevalent chronic diseases—**diabetes, heart disease, and chronic kidney disease**—using only structured clinical data. Key takeaways are:

a. Data pipeline effectiveness

- Proper handling of missing values, categorical encoding, and feature scaling enabled consistent model training across heterogeneous datasets.

b. Ensemble models excel

- Gradient Boosting and Random Forest achieved the best overall performance, confirming that tree-based ensembles are well-suited for non-linear relationships in medical data.

c. Task-specific model choice

- While Logistic Regression provided strong baselines for balanced, quasi-linear problems (heart disease), ensemble methods noticeably outperformed linear and instance-based models for the diabetes and kidney datasets.

d. End-to-end usability

- Deploying the trained models in a Streamlit web interface proved that technical ML workflows can be transformed into intuitive decision-support tools for clinicians and patients.

Together, these findings validate the feasibility of lightweight, data-driven screening systems that can supplement traditional diagnostics and potentially reduce the burden on healthcare resources.

2. Future Work

Although the current system attains high accuracy, several enhancements could further increase its clinical value:

Table: Future Work Plan Improvements

Direction	Planned Improvements
Model Explainability	Integrate **SHAP** or **LIME** to provide feature-level explanations, strengthening clinicians' trust and ensuring regulatory compliance.
Automated Feature Selection	Employ techniques such as **Recursive Feature Elimination (RFE)** or **SelectKBest** to reduce dimensionality, shorten inference time, and improve interpretability.
Advanced Algorithms	Experiment with **LightGBM** and **Extreme Gradient Boosting** for potentially higher AUC and faster training; evaluate **deep neural networks** if larger datasets are acquired.
Continual Learning	Implement online or incremental learning so the models can adapt to newly collected patient data without full retraining.
REST API Deployment	Wrap the best models in a **FastAPI/Flask** microservice to enable integration with electronic health-record (EHR) systems and mobile applications.
Clinical Validation	Conduct pilot studies with medical professionals to assess usability and diagnostic impact in real-world settings.
Multi-modal Data Fusion	Combine structured records with imaging (e.g., ultrasound, ECG) or wearable-sensor data to create a more holistic risk-assessment platform.

By addressing these future directions, the project can evolve from a proof-of-concept into a robust, clinically validated tool that enhances early detection, optimizes treatment pathways, and ultimately improves patient outcomes.

X. REFERENCCE RESOURSES

1. Hugging Face Transformers Documentation. (https://huggingface.co/transformers/)
2. These Pictures of Block Diagram model: https://www.researchgate.net/

3. Machine Learning_INT_E 14121 – Slide PhD.Nguyen Hong Son.

4. Diabetes disease Dataset:
https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

5. Heart Disease UCI dataset:
https://www.kaggle.com/datasets/ronitf/heart-disease-uci

6. Chronic_Kidney_DiseaseDataset: https://archive.ics.uci.edu/ml/datasets/chronic_kidney_disease

7. GeekforGeek tutorial's Algorthim ML: https://www.geeksforgeeks.org/ml-gradient-boosting/