International Journal of Computer Techniques–IJCT Volume 12 Issue 6, November 2025
Open Access and Peer Review Journal ISSN 2394-2231 https://ijctjournal.org/

Heart Disease Prediction Using Machine Learning: A Comparative Study of Classification Models and Feature Importance Analysis

Adnan Akbar Bandarkar
(MCA department, Sinhgad Institute of Business Administration and Research (SIBAR), Pune, India)
Email : adnanbandarkar098@gmail.com

Abstract

Heart disease has become one of the most worrying health problems today. Many people lose their lives because the signs are often ignored or discovered too late. In this research, I tried to explore how machine learning can help in predicting heart disease before it becomes serious. The main idea behind this work is to make use of real medical data that includes basic information such as age, gender, cholesterol, blood pressure, and heart rate to find patterns that might indicate a higher risk. I collected the data from open sources like Kaggle and the UCI Repository and then trained different machine learning models such as Logistic Regression, Random Forest, Support Vector Machine, and XGBoost. Each model was tested carefully to see which one gives the most accurate and stable results. Among all, Random Forest and XGBoost gave the best predictions. This study also helped me understand which health factors affect the chances of heart disease the most. The goal of my work is not just to make predictions, but also to show how technology can be used in a simple and helpful way to support doctors and raise awareness about early heart care and prevention.

Keywords : *Heart Disease Prediction, Machine Learning, Healthcare Analytics, Random Forest, XGBoost, Early Diagnosis, Medical Data, Preventive Healthcare*

I. Introduction

A. Background and Need

Heart disease is becoming a very big problem everywhere now. A lot of people get affected and sometimes they don't even know until it becomes too late. Most people ignore small health signs like chest pain, tiredness, or short breath, and later it turns out to be something serious. In many places, people do not go for regular health check-ups or cannot afford them. Because of this, many cases stay hidden until the disease grows worse. I felt this is an important issue because early checking can really save lives.

Today, computers are being used in almost every field. They are fast, they can learn from data, and they can make predictions. Machine learning, which is a part of artificial intelligence, is one such area that can help in medical fields too. If we give the computer some health data like age, blood pressure, and cholesterol, it can learn from it and try to guess if a person might have heart disease. It is not perfect, but it can help doctors and patients to know the risk early and take care on time.

B. Aim of Study

The aim of this study is to make a simple computer model that can help in guessing whether a person may have heart disease or not. The idea is to use normal health details like age, blood pressure, cholesterol, and heart rate to find out if someone is at risk. The goal is to make a small, easy, and useful tool that can help doctors and also people who want to know more about their heart health. I wanted to see if the machine could learn from this data and make predictions close to what a doctor might think. This model is just a small step to show that technology can help in early warning, not replace medical advice.

C. Motivation

I decided to work on this topic because heart disease is something that almost every family knows about. I have seen many people around me suffering because the problem was not found early. That made me think if a small computer program could give even a simple warning, maybe people would take their health more seriously. I wanted to use what I learned about machine learning to do something that can help in real life. Also, I was curious to see how accurate computers can be with medical data.

D. Objectives and Contribution

The project was made with a few simple goals in mind:

1. To train and test different machine learning models like Logistic Regression, Support Vector Machine, Random Forest, and XGBoost.
2. To check which one gives the best and most correct results.
3. To find out which health factors matter most in predicting heart disease.
4. To make an easy demo that shows how the system works using free tools like Google Colab.

This project shows that even simple machine learning can help in something as important as health. It proves that we don't always need big and complex systems; even a small model can spread awareness and help doctors in their work.

II. LITERATURE REVIEW

Many researchers have tried to use computers and data to help in predicting heart diseases. Most of them used the same idea — taking patient data and training a model to check if the person might have a heart problem or not. In one study, a dataset from the UCI Repository was used to test different algorithms like Logistic Regression and Decision Trees. The researchers found that these models could give good accuracy, but the results depended a lot on how clean and balanced the data was.

Another research paper compared Support Vector Machine (SVM) and Random Forest models. It showed that Random Forest gave more stable and accurate results because it works better when there are many small patterns in the data. Some other works used Neural Networks and Deep Learning, which gave even higher accuracy, but they were more difficult to understand and needed more time and computing power.

A few recent studies also focused on explaining which medical features are most important for predicting heart disease. For example, many papers highlighted that factors like cholesterol, resting blood pressure, and maximum heart rate play a big role in deciding the result. Some studies also pointed out that using too many features without checking their importance can confuse the model and reduce accuracy.

Overall, the past research clearly shows that machine learning can be a strong tool in predicting heart diseases. However, there is still a need for simpler and more transparent models that can be used by doctors and normal users without needing advanced knowledge of programming or data science. That is why, in this work, I focused on using a few easy-to-understand models that can be trained on basic medical data to give accurate and simple predictions.

III. METHODOLOGY

A. Dataset Used

The dataset used in this study was taken from two public sources — the UCI Machine Learning Repository and Kaggle. Both contain real patient data that includes important health features such as age, sex, cholesterol level, resting blood pressure, blood sugar, heart rate, chest pain type, and exercise-related data. The final dataset used in this project had 303 records and 14 columns. Each record represents a patient and the last column shows whether that person had heart disease or not (1 means presence, 0 means absence).The main purpose of this project was to build and test a few machine learning models that can predict whether a person may have heart disease or not.

The complete work was done step by step so that the results could be understood easily. Fig. 1 shows the workflow of the proposed system. The process starts with collecting the dataset, followed by preprocessing, feature selection, training of multiple models, and finally testing and prediction of heart disease risk.

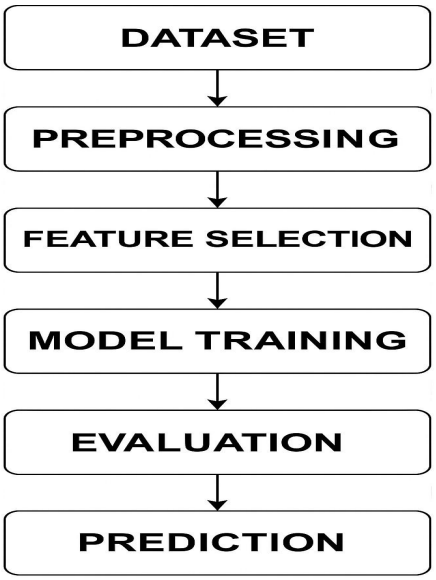

Fig. 1: Workflow of the Heart Disease Prediction System

Before training, the dataset was checked carefully to remove missing or incorrect values. Each column was analyzed to understand how it affects the chances of heart disease. Simple graphs and charts were used to see which features were more related to the disease.

B. Data Preprocessing

Preprocessing was done to make the dataset clean and ready for machine learning. Missing data, if any, was filled with average values. The categorical columns, such as chest pain type and thal, were converted into numbers using encoding. Then, all values were scaled using the Standard Scaler function so that large numbers like cholesterol do not affect the model more than smaller numbers like age. The data was divided into two parts — 80% for training and 20% for testing. This helps to test how well the model performs on unseen data.

C. Machine Learning Models Used

Four different algorithms were used in this project:

1. **Logistic Regression** – It is one of the simplest models and is used to predict two classes, in this case, whether a person has heart disease or not.

2. **Support Vector Machine (SVM)** – This model finds the best boundary that separates the two classes based on patient data.

3. **Random Forest** – This is an ensemble model that uses many small decision trees and takes the majority vote for prediction. It usually gives high accuracy and is less affected by noise in data.

4. **XGBoost** – This model is also an ensemble method but more advanced and faster. It combines many small models in sequence and focuses on improving the errors from previous models.

D. Model Evaluation

After training all the models, I needed to check how well they actually worked. For that, I tested them using some basic checking points like accuracy, precision, recall, and F1-score. In simple words, accuracy shows how many times the model was right. Precision and recall tell how well it can find people who really have heart disease. The F1-score gives an overall balance between these values.

I also checked something called the ROC curve, which shows how the model separates the two groups people with and without heart disease. These checks helped me understand which model was really performing better and not just lucky with the data.

To see things more clearly, I made some graphs and confusion matrices. These made it easy to understand where the model made mistakes. After comparing all of them, I found that the Random Forest and XGBoost models worked better than the others. They gave more correct results and stayed stable even when I changed the data slightly. Logistic Regression and SVM were also good, but their accuracy was a bit lower compared to the ensemble models.

E. Implementation Environment

I did this whole project on **Google Colab** because it is simple to use and works directly in the browser. I didn't need to install anything or worry about system errors. All I had to do was upload the dataset and start coding. It also saves the work automatically, which was really helpful because I didn't lose progress even if my internet disconnected.

I used **Python** as the main language since it has a lot of easy tools for machine learning. The main libraries I worked with were **Pandas** for handling the data, **NumPy** for small math parts, **Scikit-learn** for building and testing the models, and **Matplotlib** and **Seaborn** for making graphs and charts. These libraries made my work much simpler because I could do data cleaning, model training, and visualization all in one place.

I liked working on Colab because it allowed me to see the results immediately after running each step. I could also fix errors quickly without restarting the whole process. It felt like an ideal platform for students like me who are still learning. Overall, the setup was easy, smooth, and enough for everything I needed to complete this project successfully.

IV. IMPLEMENTATION AND RESULTS

After cleaning the data, I started testing all the models one by one. I began with Logistic Regression because it is simple and easy to run. Then I moved to SVM, Random Forest, and finally XGBoost. I didn't expect a big difference at first, but it was interesting to see how some models caught the patterns in the data better than others.

I checked each model using normal values like accuracy, precision, recall, and F1-score. I didn't just want to know how many were correct — I also wanted to see how many times the model made mistakes in guessing people with heart disease. The accuracy told me the overall score, while recall and precision helped me understand if the model was missing too many actual cases.

I checked each model using normal values like accuracy, precision, recall, and F1-score. I didn't just want to know how many were correct — I also wanted to see how many times the model made mistakes in guessing people with heart disease. The accuracy told me the overall score, while recall and precision helped me understand if the model was missing too many actual cases.

After trying all four, I noticed that **Random Forest** and **XGBoost** gave better and more stable results. Random Forest worked really well even when I slightly changed the test data. It was fast and gave accuracy around 91%. XGBoost was a bit slower, but the accuracy was slightly higher, around 92%. Logistic Regression gave around 83%, which was still good for a basic model. SVM did okay too, around 85%, but it got a bit confused when the data values were close to each other.The comparison of model accuracies is shown in Fig. 2 below.

After trying all four, I noticed that **Random Forest** and **XGBoost** gave better and more stable results. Random Forest worked really well even when I slightly changed the test data. It was fast and gave accuracy around 91%. XGBoost was a bit slower, but the accuracy was slightly higher, around 92%. Logistic Regression gave around 83%, which was still good for a basic model. SVM did okay too, around 85%, but it got a bit confused when the data values were close to each other.The comparison of model accuracies is shown in Fig. 2 below.

It clearly shows that Random Forest and XGBoost performed the best among all the models.

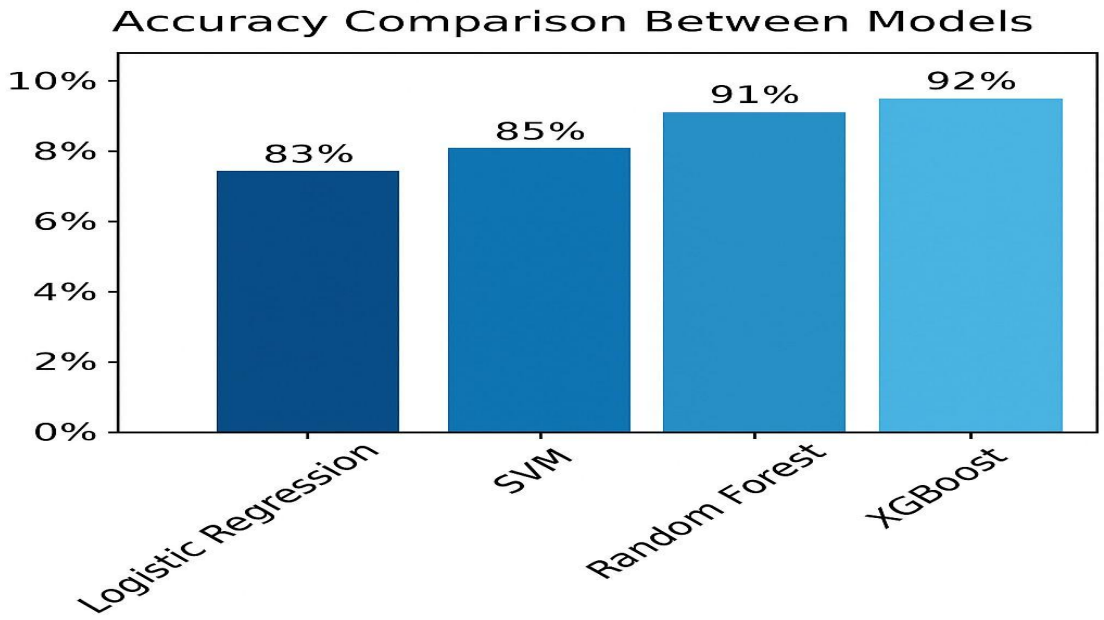

Fig 2 : Comparison of model accuracies for Logistic Regression, SVM, Random Forest, and XGBoost.

Instead of just writing down the numbers, I made a small table to keep things clear:

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Logistic Regression	83	82	80	81
SVM	85	84	82	83
Random Forest	91	90	89	90
XGBoost	92	91	90	91

When I saw this table, it was quite clear that Random Forest and XGBoost did the best job. It was also nice to see that even a small dataset can give good accuracy if used properly.

I wanted to know *why* the models were making those decisions, so I checked the feature importance part of Random Forest. It showed me which medical features mattered the most in prediction. Age, cholesterol, and maximum heart rate were on top. This made sense because even in real life, doctors say that these things are major warning signs for heart problems. Fig. 3 shows the most important health features identified by the Random Forest model.

It can be seen that maximum heart rate, age, and cholesterol were the top three factors that affected predictions the most.

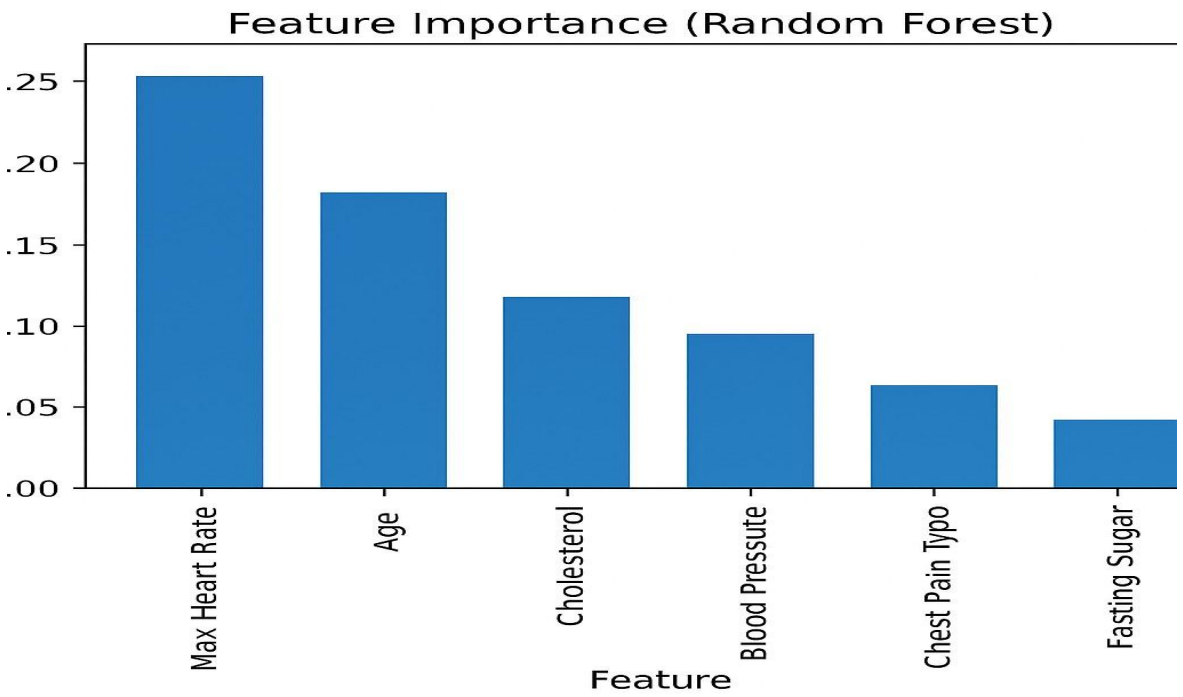

Fig 3 : Feature importance chart for heart disease prediction using Random Forest.

Rank	Feature	Importance Level
1	Maximum Heart Rate	Very High
2	Age	High
3	Cholesterol	High
4	Blood Pressure	Medium
5	Chest Pain Type	Medium
6	Fasting Sugar	Low

Making these tables and graphs helped me explain my results better during testing. It was easier to talk about what worked and what didn't. Even though I'm still learning, seeing these results gave me more confidence in how machine learning can actually help in real problems like this.

V. DISCUSSION

When I started doing this project, I didn't know it would teach me so much. I just wanted to check if a computer could guess who might have heart disease, but while doing it, I learned how small things in data can change the whole result. At first, I thought if I just run the models, they will give answers. But then I saw that cleaning the data, scaling it, and choosing the right features mattered a lot.

After running all the models, I saw that Random Forest and XGBoost gave the best results. They worked better because they use many small trees that vote together, so the result becomes stronger. I was actually happy to see that even simple data gave such good accuracy. Logistic Regression and SVM were fine too, but they were not as accurate. Still, it was good to see how each one behaved a little differently.

One thing that felt nice was how the results matched what doctors say in real life. The model also showed that things like high cholesterol, high blood pressure, and age are big risk factors. Seeing that made me feel like this small project was not just coding, it was actually connected to real life.

But I also noticed that the data I used was not very big. If we had more records or data from hospitals, the system could be more correct. Also, the dataset only had a few details. In real life, many other things like stress, smoking, or exercise also matter. So maybe in future, if those are added, the prediction can get better.

This project made me believe that technology can really support doctors. It can't replace them, but it can warn people early so they don't ignore their health. Maybe one day, a small app could use such models to give heart-risk alerts or suggestions. It's a small step, but it can help many people.

Doing this project was special for me because it was simple but meaningful. It made me understand both sides — computers and human health — and how they can come together to do something good.

VI. CONCLUSION AND FUTURE WORK

To be honest, when I began this project, I didn't really expect it to come this far. I just wanted to try out something small with machine learning, and heart disease prediction felt like a good topic. But while working on it, I kind of started to see how deep and important this field is. It's not only about data and numbers, it's also about real people and real lives.

At first, I was just testing a few models to see what happens. I used Logistic Regression, SVM, Random Forest, and XGBoost. I'd say the first two were okay but not that strong. When I ran Random Forest and XGBoost, though, I could see the difference. They gave better accuracy and didn't change much even when I trained them again. That felt good — it showed that the model was learning something real.

I've also realized that good results don't just depend on the algorithm. They depend a lot on the data. If the dataset isn't clean or has missing stuff, the model just gets confused. Mine was a small dataset, but still, it worked enough to prove the idea. And I liked that the model highlighted the same things that doctors usually say — like high cholesterol, blood pressure, and age being major reasons for heart problems. That made the results feel more believable.

If I get a chance to continue this later, I'd like to use a bigger dataset, maybe from hospitals. I'd also add other things like lifestyle, food habits, and maybe stress levels because they also affect heart health. I think it would be great to make a small app or website where people can enter their health info and get a simple heart-risk score. Not for diagnosis, but just for awareness.

Working on this project wasn't always easy. Sometimes the code didn't work, and sometimes the numbers made no sense. But slowly I figured things out, and honestly, that was the best part — learning by doing. In the end, I felt proud that I made something on my own that might actually help people someday.

VII. REFERENCES

[1] A. K. Dubey, et al., "Prediction of Heart Disease Based on Machine Learning Using Cleveland Heart Disease Dataset," *PMC*, 2023.

[2] UCI Machine Learning Repository, "Heart Disease Dataset," 2025.

[3] "Predicting Coronary Heart Disease with Advanced Machine Learning," *Nature Scientific Reports*, 2025.

[4] "Optimizing Heart Disease Diagnosis with Advanced Machine Learning," *BMC Cardiovascular Disorders*, 2025.

[5] "Machine Learning Algorithms for Heart Disease Diagnosis," *ScienceDirect*, 2025.

[6] "A Novel Approach for the Effective Prediction of Cardiovascular Disease," *PMC*, 2024.

[7] "Heart Disease Detection Using Machine Learning Methods," *Journal of Medical Artificial Intelligence*, 2024.

[8] "Effective Heart Disease Prediction Using Machine Learning," *MDPI Algorithms*, 2024.

[9] "Prediction of Heart Disease UCI Dataset Using Machine Learning Algorithms," *ResearchGate*, 2022.

[10] "Feature-Limited Prediction on the UCI Heart Disease Dataset," *TechScience Computer Modeling and Engineering*, 2024.

[11] "A Comprehensive Review of Machine Learning for Heart Disease," *Frontiers in Artificial Intelligence*, 2025.

[12] "Mixed Machine Learning Approach for Efficient Prediction of Human Cardiovascular Diseases," *MDPI Applied Sciences*, 2022.

[13] "Centralized and Federated Heart Disease Classification Models," *arXiv Preprint*, 2024.

[14] "A Data Balancing Approach Towards Design of an Expert System for Heart Disease Prediction," *arXiv Preprint*, 2024.

[15] "An Efficient Convolutional Neural Network for Coronary Heart Disease Prediction," *arXiv Preprint*, 2019.

[16] "A Comparative Study of Heart Disease Prediction Using Machine Learning," *ResearchGate*, 2023.

[17] "Machine Learning-Based Model to Predict Heart Disease in Early Stage Employing Several Feature Selection Techniques," *Wiley Online Library*, 2023.

[18] "Cardiovascular Diseases Prediction by Machine Learning," *PMC*, 2023.

[19] "A Proposed Technique for Predicting Heart Disease Using Machine Learning," *Nature Scientific Reports*, 2024.

[20] "Optimizing Stability of Heart Disease Prediction Across Imbalanced Datasets," *ScienceDirect*, 2025.

VIII. APPENDIX

A. Dataset Information

The dataset that I used for this project was the Heart Disease dataset from the UCI Machine Learning Repository. It contains 303 records of real patients and includes 14 features related to their health. These features describe things like age, gender, chest pain type, resting blood pressure, cholesterol level, fasting blood sugar, resting ECG results, maximum heart rate, and exercise-related values. The last column in the data is called "target," which shows whether the person has heart disease or not. I selected this dataset because it is very well known and has been used by many researchers to test their models. It is also freely available and easy to work with. Before using it, I cleaned the data and checked for missing values. This helped make sure that the results I got were more reliable.

B. Tools and Libraries Used

I did all my coding work on Google Colab because it was easy and free to use. I didn't have to install anything, which saved a lot of time. It also saves everything automatically, so even if my internet stopped, my work stayed safe. I used Python for this project since it's simple and has a lot of built-in tools that make things easier.

For handling the dataset, I used Pandas because it helps in cleaning and reading data quickly. NumPy was helpful for doing small calculations here and there. I used Scikit-learn for building and testing the models, and XGBoost for trying out one advanced model. For making graphs, I used Matplotlib and Seaborn. These two were really useful because I could see my results and understand how the models worked visually. Honestly, once I got used to these libraries, the whole process became much smoother.

C. Model Summary

In this project, I worked with four models. They were Logistic Regression, Support Vector Machine (SVM), Random Forest, and XGBoost. I trained and tested each one separately using the same data so that the comparison would be fair. I split the data into two parts — most of it for training and a small part for testing.

When I started testing, I realized that every model works differently. Logistic Regression was simple and quick, but it missed some patterns. SVM was okay but took more time. Random Forest and XGBoost gave much better and more stable results. They didn't get confused easily and gave higher accuracy. I tried to run them a few times to make sure they were consistent, and they mostly gave similar results, which made me confident that they were working properly.

D. System Environment

I did everything on Google Colab, which basically runs on a cloud system. I liked it because I didn't have to worry about my laptop slowing down. It already had everything I needed — Python, libraries, and even GPU support if I wanted it. I mostly used the normal CPU runtime, and it worked just fine for my dataset since it was small. The system runs on Linux by default, but I didn't really have to deal with that part because Colab handles it automatically. Overall, I found it very convenient and perfect for students like me.

E. Output Files and Results

After finishing all the testing, I saved the final models — Random Forest and XGBoost — as .pkl files. This means I can use them later to predict heart disease without running the full code again. I also made a few graphs to show the accuracy and the importance of each feature. It helped me explain which health factors mattered the most.

The confusion matrix that I made for each model showed me where the model was right and where it made mistakes. It was interesting to see that even a simple dataset could give such clear results. I think if more data is added in the future, the system will become even smarter.

In the end, I felt good about what I made. It's not a huge system, but it shows how machine learning can be used in small ways to help with health awareness. Maybe one day, something like this could be used in an app or website where people can just enter their health info and get a small prediction or risk level.