

Flask Cardio Risk Building a Predictive Model for Cardiovascular Health

Divya Bharathi M¹, Shai Subramanian GJ², Parthiban G³, Raghu K⁴ and Vinodh D⁵

¹Assistant Professor -Department of Information Technology & Kings Engineering College-India.

^{2,3,4,5} Department of Information Technology & Kings Engineering College-India.

Abstract - Flask, a Python web framework, to assess an individual's risk of cardiovascular diseases. Employing machine learning algorithms, this project analyzes diverse health parameters to generate personalized risk assessments. The primary objective is to establish a user-friendly web interface where users can conveniently input their health data for receiving tailored cardiovascular risk evaluations. By offering early insights into potential health risks, this project strives to empower individuals with proactive measures for preventing and managing cardiovascular diseases. Through accessible and personalized risk assessments, "FlaskCardioRisk" aims to contribute to improved healthcare strategies for better cardiovascular health outcomes.

1.INTRODUCTION

Flask, a Python web framework, to assess an individual's risk of cardiovascular diseases. Employing machine learning algorithms, this project analyzes diverse health parameters to generate personalized risk assessments. The primary objective is to establish a user-friendly web interface where users can conveniently input their health data for receiving tailored cardiovascular risk evaluations. By offering early insights into potential health risks, this project strives to empower individuals with proactive measures for preventing and managing cardiovascular diseases. Through accessible and personalized risk assessments, "FlaskCardioRisk" aims to contribute to improved healthcare strategies for better cardiovascular health outcomes.

1.2Objective:

Flask Cardio Risk is a project using Flask, a Python web framework, to create a predictive model for assessing an individual's risk of cardiovascular diseases. This initiative aims to leverage machine learning algorithms to analyze health data, offering a user-friendly interface for users to input information and receive personalized risk assessments. Ultimately, it seeks to enable proactive healthcare by providing early insights into cardiovascular health risks for better prevention and management.

1.3Problem Statement:

The "Flask Cardio Risk" project addresses the need for an accurate and scalable predictive model for assessing cardiovascular disease risk. Current methods often lack accuracy and scalability, hindering effective prevention and management. Leveraging Flask and machine learning algorithms, the project aims to develop a user-friendly web interface for personalized risk assessments, empowering individuals with early insights for proactive healthcare measures.

1.3Scope of The Project:

The scope of the "Flask Cardio Risk" project encompasses the development of a user-friendly web interface using Flask, a Python web framework, for assessing an individual's risk of cardiovascular diseases. The project will involve the integration of machine learning algorithms to analyze diverse health parameters and provide personalized risk assessments. Key aspects of the project include data collection, preprocessing, feature extraction, model training, evaluation, and the creation of a responsive and intuitive web interface for users to input their health data and receive tailored risk evaluations. Additionally, the project will focus on scalability, efficiency, and accuracy to ensure reliable predictions and facilitate proactive healthcare measures for disease prevention and management.

1.4Existing System:

The existing systems for cardiovascular disease risk assessment face several challenges. These include limitations in accuracy, scalability, and efficiency, particularly when dealing with large datasets. Current approaches may also struggle with overfitting or underfitting issues, and some are not adaptable for real-time data collection and analysis. Moreover, certain methods may only be applicable to small datasets, limiting their utility in broader healthcare contexts. Overall, there is a need for improved predictive models that can address these shortcomings and provide more accurate and scalable assessments of cardiovascular disease risk.

1.4.1Disadvantages:

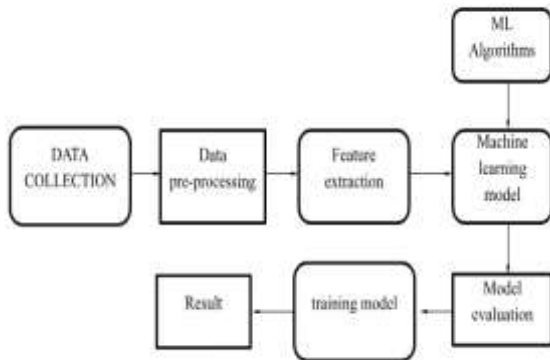
- Limited Applicability
- Overfitting or Underfitting
- Scalability Issues
- Accuracy Limitations
- Real-time Adaptability

1.5Proposed System:

Incorporating boosting algorithms into our proposed system for heart disease prediction offers several advantages. Boosting algorithms, like AdaBoost or Gradient Boosting, improve prediction accuracy by combining weaker learners iteratively. This enhances the precision of risk assessments for individuals while ensuring scalability to handle large healthcare datasets efficiently. Moreover, these algorithms converge faster, reducing time consumption in model training and prediction. Their ability to maintain high performance in continuously updating the prediction model ensures accuracy and relevance over time, making them valuable additions to our system for proactive healthcare measures.

1.5.1 Advantages:

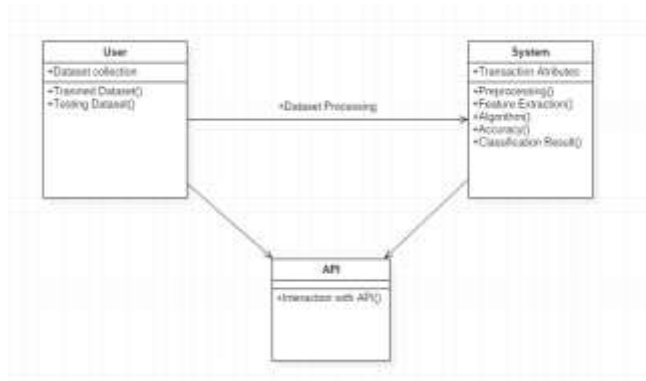
- Improved Prediction Accuracy.
- Scalability.
- Adaptability to Changing Data.
- Robustness to Overfitting



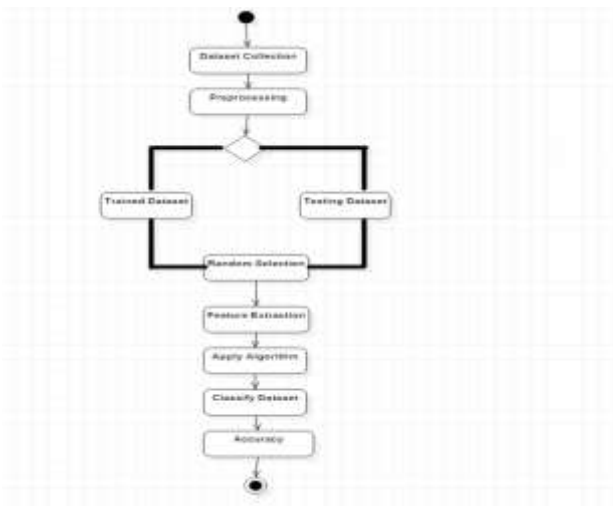
2.SYSTEM DESIGN:

2.1.DATA FLOW DIAGRAM

CLASS DIAGRAM



ACTIVITY DIAGRAM



2.2 UML diagram:

Unified Modelling Language (UML) is used to specify, visualize, modify, build, and document the artefacts of object-oriented software-intensive systems under development. UML provides a standard way to visualize a system's architectural blueprint, including elements such as:

Actor

- Business process
- (logical) components
- Activities
- programming language statements
- Database schema and
- Reusable software components. UML combines the best practices of data modelling (entity-relationship diagrams), business modelling (workflows), object modelling, and component modelling. It can be used in any process, across different implementation technologies, throughout the software development lifecycle. UML synthesized Booch's method, the Object Modelling Technique (OMT), and the Object Oriented Software Techniques (OOSE) notation by unifying them into one popular and widely used modelling language. UML aims to be a standardized modelling language that can model concurrent and distributed systems.

Sequence Diagram:-

Sequence diagram Represents the objects involved in an interaction horizontally and vertically in time. A use case is a type of behaviour classifier that represents a declaration of provided behaviour. Each use case specifies specific behaviour. This may include variants that the subject can perform cooperatively with one or more of her actors. A use case defines the behaviour provided by a subject without reference to the internal structure of the subject. These actions, including interactions between actors and subjects, can lead to changes in the subject's state and communication with its environment. A use case can have many variations on the basic behaviour, such as anomalous behaviour and error handling.

Activity Diagram:-

Activity diagrams are graphical representations of workflows containing step-by-step activities and actions that support selection, repetition, and concurrency. Unified Modelling Language allows you to use activity diagrams to step-by-step describe the business and operational workflows of the components in your system. Activity diagrams show the overall control flow of this project.

Use case diagram:-

UML is a standard language for specifying, visualizing, building, and documenting the artefacts of software systems. • UML was developed by the Object Management Group (OMG) and the UML 1.0 draft specification was submitted to OMG in January 1997. • OMG is continuously committed to creating true industry standards. • UML stands for Unified Modeling Language. • UML is a visual language used to create software designs. Class chart:-

Class diagrams are a key building block of object-oriented modeling. It is used for general conceptual modeling of the application's system and detailed modeling by translating the model into programming code. Class diagrams can also be used for data modeling. [1] Classes in class diagrams represent both major elements, interactions within the application, and classes to be programmed. In the diagram, a class is represented by a box containing three compartments.

1. The top compartment contains the name of the class. Make it bold, centered, and capitalize the first letter.
2. The centre compartment consists of the magnificence attributes. They are left aligned and the primary letter is lowercase.
3. The backside compartment consists of operations that the magnificence can perform. They also are left aligned and the primary letter is lowercase.

Domain Specification

3.MACHINE LEARNING

Machine Learning is a system that can learn from example through self-improvement and without being explicitly coded by programmer. The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., example) to produce accurate results.

Machine learning combines data with statistical tools to predict an output. This output is then used by corporate to makes actionable insights. Machine learning is closely related to data mining and Bayesian predictive modeling. The machine receives data as input, use an algorithm to formulate answers.

A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user's historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation.

Machine learning is also used for a variety of task like fraud detection, predictive maintenance, portfolio optimization, automatize task and so on.

Machine Learning vs. Traditional Programming

Traditional programming differs significantly from machine learning. In traditional programming, a programmer code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.

Machine Learning

How does Machine learning work?

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if its feed a previously unseen example, the machine has difficulties to predict.

The core objective of machine learning is the learning and inference. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the data. One crucial part of the data scientist is to choose carefully which data to provide to the machine. The list of attributes used to solve a problem is called a feature vector. You can think of a feature vector as a subset of data that is used to tackle a problem.

The machine uses some fancy algorithms to simplify the reality and transform this discovery into a model. Therefore, the learning stage is used to describe the data and summarize it into a model.

For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant: This is the model

Inferring

When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.

4.SOFTWARE AND HARDWARE REQUIREMENTS:

Hardware:

- OS – Windows 7, 8 and 10 (32 and 64 bit)
- RAM – 4GB
- Database

Software:

- Python / Anaconda Navigator
- Packages: numpy, Pandas, matplotlib, Sklearn
- Flask Framework

Hardware Explanation:

Operating System (OS):

The OS is the software that manages computer hardware and provides common services for computer programs. Windows 7, 8, and 10 are different versions of the Windows operating system developed by Microsoft.

RAM (Random Access Memory):

RAM is a type of computer memory that is used to store data and machine code currently being used or processed by the CPU.

ACKNOWLEDGEMENT

We thank God for his blessings and also for giving us good knowledge and strength in enabling us to finish our project. Our deep gratitude goes to our founder late **Dr. D. SELVARAJ, M.A., M.Phil.**, for his patronage in the completion of our project. We like to take this opportunity to thank our honourable chairperson **Dr.S. NALINI SELVARAJ, M.COM., MPhil., Ph.D.** and honourable director, **MR.S. AMIRTHARAJ, M.Tech., M.B.A** for their support given to us to finish our project successfully. We wish to express our sincere thanks to our beloved principal, **Dr. C. Ramesh Babu Durai M.E., Ph.D** for his kind encouragement and his interest towards us. We are extremely grateful and thanks to our professor **Dr. D. C. Jullie Josephine M.Tech., PhD**, Head of Information Technology, Kings Engineering College, for his valuable suggestion, guidance and encouragement. We wish to express our sense of gratitude to our project supervisor **Mrs. Divya Bharathi M.E.**, Assistant Professor of Information Technology Department, Kings Engineering College whose idea and direction made our project a grand success. We express our sincere thanks to our parents, friends and staff members who have helped and encouraged us during the entire course of completing this project work successfully.

CONCLUSION:

FlaskCardioRisk successfully integrates machine learning with a user-friendly Flask-based web interface to assess cardiovascular disease risk. By analyzing health parameters, it provides personalized risk evaluations, empowering individuals with early insights into their heart health. This tool

encourages proactive healthcare decisions, enabling users to adopt preventive measures and lifestyle changes to mitigate risks. With a simple yet effective design, it bridges the gap between medical expertise and accessibility, fostering greater awareness about cardiovascular diseases. Overall, FlaskCardioRisk serves as a valuable digital health assistant, contributing to improved public health outcomes by promoting early detection and prevention strategies.

FUTURE WORK:

Future enhancements for FlaskCardioRisk could include expanding the dataset to improve predictive accuracy and integrating real-time health monitoring via wearable devices. Implementing advanced deep learning models can enhance risk evaluation precision. Additionally, multilingual support and voice-assisted inputs could improve accessibility for a wider audience. A mobile application version would enhance usability, allowing users to track their cardiovascular health on the go. Collaborations with healthcare professionals to validate predictions and provide medical recommendations could add credibility. Finally, incorporating lifestyle coaching features based on risk assessments can encourage long-term health improvements and disease prevention strategies.

REFERENCES:

1. S. Abdullah and R. R. Rajalaxmi, "A data mining model for predicting the coronary heart disease using random forest classifier", Proc. Int. Conf. Recent Trends Comput. Methods Commun. Controls, pp. 22-25, Apr. 2012.
2. A. H. Alkeshuosh, M. Z. Moghadam, I. Al Mansoori and M. Abdar, "Using PSO algorithm for producing best rules in diagnosis of heart disease", Proc. Int. Conf. Comput. Appl. (ICCA), pp. 306-311, Sep. 2017.
3. N. Al-milli, "Backpropagation neural network for prediction of heart disease", J. Theor. Appl. Inf. Technol., vol. 56, pp. 131-135, 2013.
4. C. A. Devi, S. P. Rajamhoana, K. Umamaheswari, R. Kiruba, K. Karunya and R. Deepika, "Analysis of neural networks based heart disease prediction system", Proc. 11th Int. Conf. Hum. Syst. Interact. (HSI), pp. 233-239, Jul. 2018.
5. P. K. Anooj, "Clinical decision support system: Risk level prediction of heart disease using weighted fuzzy rules", J. King Saud Univ.-Comput. Inf. Sci., vol. 24, no. 1, pp. 27-40, Jan. 2012.