

SRM UNIVERSITY AP

Department of Computer Science and Engineering

Machine Learning project report on

Heart Disease Prediction

By

U Abhinav – AP21110010915

MD Ameen – AP21110010948

D Sai Amruth – AP21110010962

M Karthik – AP21110010967

Under the Guidance of Dr. M Mahesh Kumar

**Introduction:**

The aim of a heart disease prediction project in machine learning is to develop a model that can accurately predict the likelihood of an individual having heart disease based on various input features such as age, gender, blood pressure, cholesterol levels, etc. Once the data is collected, it needs to be pre-processed to handle missing values, normalize features, and encode categorical variables. This step ensures that the data is suitable for training a machine learning model.

We have collected the data from Kaggle, it contains 14 features and 303 instances. We have given Target as our class label and target values are 0 & 1, 0 represents the person don’t have heart disease, 1 represents the person have heart disease.

🡪0 - Healthy Heart

🡪1 - Defective Heart

If your model has a high accuracy score, it means your model is making correct predictions most of the time. If the accuracy score is low, it means your model is not very good at predicting outcomes accurately. Accuracy score is usually represented as a percentage, where higher percentages indicate better performance.

**Abstract:**

Machine learning techniques, particularly supervised learning algorithms, have emerged as powerful tools for heart disease prediction by leveraging patient data. This abstract presents an overview of heart disease prediction using machine learning methods, focusing on logistic regression.

The process begins with the collection of comprehensive patient data, including demographics, medical history, and clinical parameters such as blood pressure, cholesterol levels, and electrocardiogram (ECG) readings. Following data preprocessing and feature selection, logistic regression models are trained to learn the relationship between input features and the presence or absence of heart disease.

Evaluation of the trained models is crucial for assessing their predictive performance. Metrics such as accuracy, and area under the receiver operating characteristic (ROC) curve are employed to quantify model effectiveness in correctly identifying patients with heart disease. Additionally, considerations such as data imbalance and model interpretability are addressed to ensure reliable predictions in real-world applications.

By providing a comprehensive overview of heart disease prediction using logistic regression and machine learning, this abstract, aims to underscore the potential of predictive modelling in improving cardiovascular health outcomes. The insights gained from such models can inform clinical decision-making, enabling timely interventions and personalized treatment strategies for individuals at risk of heart disease.

**System Requirements:**

System requirements for heart disease prediction in machine learning depend on various factors, including the size of the dataset, complexity of the machine learning model, and available computational resources. Here's a general outline of the system requirements:

**Hardware Requirements:**

CPU: A multi-core processor with sufficient processing power to handle data preprocessing, model training, and evaluation. Modern CPUs with multiple cores (e.g., Intel Core i5/i7 or AMD Ryzen series) are suitable for most tasks.

Memory (RAM): An adequate amount of RAM is essential, especially when working with large datasets. At least 8 GB of RAM is recommended, although more may be necessary for handling bigger datasets or complex models.

Storage: Sufficient disk space is needed to store datasets, intermediate results, and trained models. Solid-state drives (SSDs) are preferred for faster data access and model training.

**Software Requirements:**

Python: Most machine learning libraries and frameworks are written in Python, making it the preferred programming language for developing heart disease prediction models. Ensure Python is installed along with popular data science libraries like NumPy, pandas, scikit-learn.

Machine Learning Libraries: Utilize popular machine learning libraries and frameworks for building and training predictive models. Scikit-learn offers a wide range of algorithms.

Integrated Development Environment (IDE): Choose an IDE or text editor for coding and experimentation. Common choices include Jupyter Notebook, PyCharm, Visual Studio Code, and Spyder, each offering features for efficient development and debugging.

Internet Connectivity: Access to the internet may be necessary for downloading datasets, accessing documentation, and installing software packages and dependencies.

Operating System: Most machine learning tools and libraries are compatible with major operating systems like Windows, macOS, and Linux. Choose an operating system based on personal preference and compatibility with required software.

**Data Preprocessing:**

There are no missing values in our dataset. So, it is already clean, and the duplicate values are removed.

**Model Training:**

Splitting the Data into Training data & Test data

**Model Training:**

**Logistic Regression:**

Logistic Regression model is useful for Binary classification.

**Data Collection:** Relevant patient data is collected from various sources, including medical records, clinical databases, and research studies. This data typically includes demographic information (age, sex), medical history (family history of heart disease, smoking status), and clinical measurements (blood pressure, cholesterol levels, ECG readings).

**Data Preprocessing:** The collected data is preprocessed to handle missing values, normalize features, and encode categorical variables. This ensures that the data is in a suitable format for training the logistic regression model.

**Model Training:** The preprocessed data is split into training and testing sets. The logistic regression model is then trained on the training. During training, the model learns the relationship between the input features and the probability of a patient having heart disease.

**Model Evaluation:** Once trained, the model is evaluated using the testing set to assess its performance. Evaluation metrics such as accuracy and area under the receiver operating characteristic (ROC) curve are calculated to measure the model's ability to correctly classify patients with and without heart disease.

**Random Forest:**

Random Forest is a powerful machine learning algorithm used for classification and regression tasks. It belongs to the ensemble learning methods, where multiple individual models are combined to improve the overall performance.

In addition to using random samples of the training data, Random Forest also uses a random subset of features for each decision split.

Once all the trees are trained, classification in Random Forest is done through a voting mechanism for classification problems. Each tree "votes" for the class label, and the class with the majority of votes is predicted as the final output.

**Accuracy on Random Forest:**

**Model Evaluation:**

**Accuracy Score:**

In machine learning, the accuracy score is a common performance metric used to evaluate the performance of a classification model. It measures the proportion of correctly classified instances out of all instances evaluated.

Here's how the accuracy score is calculated:

Accuracy=Number of Correct Predictions/Total Number of Predictions×100%

In the context of heart disease prediction, accuracy score quantifies how well the predictive model correctly identifies patients with and without heart disease.

For example, let's say you have a dataset with 100 patient records, and you've trained a heart disease prediction model. After evaluating the model, you find that it correctly predicts 80 patients' heart disease status (either present or absent) out of the total 100 patients in the dataset.

Accuracy=100/80​×100%=80%

So, the accuracy score of the model is 80%. This means that the model correctly classified 80% of the patients' heart disease status in the dataset.

While accuracy score provides a straightforward measure of a model's overall performance, it may not always be sufficient, especially in cases of imbalanced datasets where one class is much more prevalent than the other. In such cases, other metrics like precision, recall, F1-score, and area under the receiver operating characteristic (ROC) curve provide a more comprehensive evaluation of the model's performance.

The above picture shows the accuracy on Training Data

The above picture shows the accuracy on Test Data

Accuracy on training data = 85%

Accuracy on test data = 82%

That was not a big difference so that our model is correct predictions, most of the time. This accuracy score is usually represented as a percentage, where higher percentages indicate better performance.

When it comes to large difference between training data and test data it comes under Overfitting model. But our model accuracy score is not having big difference. So, that does not come under Overfitting.

**Result:**

Building a Predictive system:

A screenshot of a computer program

Description automatically generated

According to our Data set the prediction is correct by this we can say our model is evaluating the predictions correctly most of the time. Whenever we have a high accuracy model gives correct predictions.

**Conclusion:**

A heart disease prediction project using logistic regression and accuracy score aims to

\*\*Logistic regression is a type of machine learning algorithm used for binary classification tasks, which is suitable for predicting the presence or absence of heart disease in this context.

\*\*The accuracy score refers to how often our model makes correct predictions. A high accuracy score means the model is good at predicting whether someone has heart disease or not, while a low accuracy score indicates the model is less reliable.

**References:**

<https://ieeexplore.ieee.org/document/9734880>

<https://www.researchgate.net/publication/351545128_Heart_Disease_Prediction_Using_Machine_Learning>

<https://github.com/g-shreekant/Heart-Disease-Prediction-using-Machine-Learning>