

HEARTDISEASE PREDICTION USING MACHINE LEARNING

¹Uma Maheswara Rao, ² Mrs. Bandi. Suneetha

¹Student, Dept. of Master of Computer Applications, Amrita Sai Institute of Science and Technology, Paritala, Andhra Pradesh, 521180, India.

²Asst.Prof, Dept. of Computer Science & Engineering, Amrita Sai Institute of Science and Technology, Paritala, Andhra Pradesh, 521180, India.

ABSTRACT

This project studies how different health factors can influence a person's health. It studies several attributes such as Heart Disease, High Blood Pressure, High Cholesterol, BMI, Smoking, Stroke, Diabetes, Physical Activity, Dietary Habits, Alcohol Consumption, Healthcare Accessibility and important Demographic Information (Sex, Age and Education). Before using the data, it is checked for missing data, unusual points and analyzed using EDA to understand the connection between different variables. New significant features are added to the data using feature engineering. In healthcare, Logistic Regression, Decision Trees, Random Forest, Gaussian Naive Bayes and K-Nearest Neighbours (KNN) are used to predict future health conditions. Model results are measured by using accuracy, a confusion matrix and an ROC curve. The Streamlit platform is built so that users can engage with the data, see graphs of health features, look for unusual examples and experiment with models. With this project, users gain knowledge about health data, make predictions about health outcomes and can explore different health features.

Keywords: Health Data, Machine Learning, Predictive Modeling, Heart Disease, High Blood Pressure, BMI

1. INTRODUCTION

1.1 Project Overview:

As part of the project, we analyse data that provide details on items such as heart disease or attack, high blood pressure, high cholesterol, BMI and people's smoking habits, among others. Do the data preparation, analyse the information, choose great features and make useful models using machine learning. Logistic Regression, Decision Trees, Random Forest, Gaussian Naive Bayes and K-Nearest Neighbours (KNN) are the algorithms included in this project. To make using the tool simpler, Streamlit was used to provide data visualisation, spot outliers and review models. In general, the goal is to understand health information and build models that help explain it better.

1.2 Scope:

The project deals with health-related data through steps such as cleaning it, analysing it, developing features and making predictive models by using machine learning. It is meant to help with creating reliable predictions for

heart disease, stroke and diabetes. Adding Streamlit is necessary, as it makes it easy for anyone to manage the software, cheque the data, spot outliers and go over the machine learning models.

1.3 Purpose:

For this project, the overall goal is to study several factors related to health and develop models that help understand and predict issues like heart disease, stroke and diabetes. Logistic Regression, Decision Trees, Random Forest, Gaussian Naive Bayes, K-Nearest Neighbours (KNN) and many other appropriate machine learning algorithms are applied in the project. The new models make it easier for us to predict and understand people's health and Streamlit makes this possible for anyone who comes across the application.

1.4 Objective of the Study:

Here are the objectives of this study:

- You should address any missing values and outliers during data preprocessing to help ensure the quality of the dataset. Transform data

using the same methods to achieve the same results no matter the scale.

- Look into the connexion between different health aspects and overall health outcomes. Study the data to find patterns, trends and connexions that lead to important insights.
- Feature Engineering: Generate additional features from the available data to improve the models' prediction power. Pick out the key features for the model as this will likely enhance its performance.
- Machine Learning Development: Apply Logistic Regression, Decision Trees, Random Forest, Gaussian Naive Bayes and K-Nearest

Neighbours (KNN) to predict what affects people's health. Make use of advanced techniques to ensure these models perform well and are accurate.

- To assess a model's performance, cheque its accuracy, use a confusion matrix and draw an ROC curve. Cheque the accuracy of different algorithms to see which ones perform best in predicting results.
- Design a simple and straightforward streamlit application that users can use for data, outlier and model analysis. Allow individuals to explore the data and models to learn more about health-related factors affecting outcomes.

1.5 Existing System:

S.No	Title of the Paper	Algorithm(s) Used	Limitations
1	Health Outcome Prediction Using Machine Learning	Logistic Regression, Decision Trees, Random Forest, Gaussian Naive Bayes, K-Nearest Neighbours (KNN)	The dataset contains numerous health-related attributes, making analysis and prediction challenging.
2	Predicting Health Outcomes Using Machine Learning	Logistic Regression, Decision Trees, Random Forest, Gaussian Naive Bayes, K-Nearest Neighbours (KNN)	The dataset's numerous health-related attributes pose challenges for analysis and prediction.
3	Machine Learning for Health Outcome Prediction	Logistic Regression, Decision Trees, Random Forest, Gaussian Naive Bayes, K-Nearest Neighbours (KNN)	Numerous health-related attributes make analysis and prediction challenging.
4	Health Prediction using Machine Learning Algorithms	Logistic Regression, Decision Trees, Random Forest, Gaussian Naive Bayes, K-Nearest Neighbour (KNN)	Addressing missing values, outliers, and inconsistencies impacts the model's accuracy and reliability.
5	Machine Learning Models for Health Outcome Prediction	Logistic Regression, Decision Trees, Random Forest, Gaussian Naive Bayes, K-Nearest Neighbours (KNN)	Numerous health-related attributes pose challenges for analysis and prediction.
6	Predictive Modeling of Health Outcomes Using Machine Learning	Logistic Regression, Decision Trees, Random Forest, Gaussian Naive Bayes, K-Nearest Neighbours (KNN)	Numerous health-related attributes complicate analysis and prediction.

TABLE I: EXISTING SYSTEM

1.6 Problem Statement

Because healthcare generates significant amounts of data, it is vital to know how health-related attributes play a role in the general health outcomes people experience. Still, using different healthcare data can lead to several difficulties when predicting health outcomes. There are many aspects in the dataset, including heart disease, high blood pressure, cholesterol, BMI, how much someone smokes and so

on. Since these attributes are deeply related, analysing and predicting them is a challenge. Moreover, the data may contain missing points, unusual numbers and errors that could negatively influence the predictions. For accurate and reliable models, it is necessary to use suitable data preparation steps. To get reliable and accurate outcomes when predicting heart disease, stroke and diabetes, one has to select the correct computation algorithms and carefully evaluate the models.

1.7 Proposed System

The system uses a setup that involves preprocessing the data, performing data exploration, engineering relevant features, designing models and evaluating them. At first, missing values, outliers and differences among data will be handled through techniques called mean imputation and normalization. EDA will be carried out to highlight the links between health-related factors and their outcomes.

Engineering features allows you to pick the most useful attributes and improve the model’s performance. To predict different health outcomes, I will use Logistic Regression, Decision Trees, Random Forest, Naive Bayes and KNN and evaluate the outcomes using accuracy and ROC curves.

With a Streamlit-built interface, users can safely interact with the information, look at its main aspects and assess the models. The system will be combined, set up and frequently updated to stay current. Adaptability of big data enables healthcare management to better predict early diagnosis, prevention and treatment by making sure predictions are accurate.

2. SYSTEM ANALYSIS

By analysing a system, you can see the individual parts and identify areas where changes could be made. Its goal is to find out the problem and offer a computerised way to solve it.

2.1 Feasibility Study

A feasibility study helps determine if a new system would be valuable for the organisation. The approach assesses the system considering economic, technical, operational and behavioural factors.

- **Operational Feasibility**
This checks whether the system can fulfil its required tasks and be used successfully after it is put into use. When the system meets communication requirements, it is regarded as viable for use.
- **Technical Feasibility**
The question here is if there are enough software, hardware and resources for both building and maintaining the system. It checks that the existing resources are sufficient to power the system.
- **Behavioral Feasibility**
This area analyses whether users are ready to accept new behaviour or resist it. It checks if users will and can accept the computerised

system and also ensures that security and authentication work as intended.

2.2 System Requirements Specification

The SRS is used to outline the main requirements that a system must meet. It provides the list of functions and operations that the Project Concept Proposal, Project Business Case and Project Charter gathered. After this document is approved, it means both sides have agreed that when the system is developed to meet the requirements, it will be accepted.

2.2.1 Hardware Requirements:

Category	Requirement
System	Pentium i3 Processor
Hard Disk	500 GB
Monitor	15" LED
Input Devices	Keyboard, Mouse
RAM	2 GB

TABLE2: HARDWARE REQ

2.2.2 Software Requirements:

Category	Requirement
Operating System	Windows 10
Coding Language	Python

TABLE3: SOFTWARE REQ

2.2.3 Library Requirements:

Library	Description
Sklearn	Scikit-learn (Machine Learning Library) – Provides tools for classification, regression, clustering, and dimensionality reduction.
Pandas	A data analysis library – Built on top of NumPy, it simplifies data manipulation and analysis.
NumPy	A numerical computing library – Essential for mathematical operations on arrays and matrices.

TABLE4: LIBRARY REQ

3. SYSTEM DESIGN

3.1 About System Design:

While designing a system, you construct its architecture, modules, components, interfaces and also outline the data flow. The objective is to describe the system in detail to help implement it according to architectural models.

The main elements of any system are:

- Architecture: Provides a clear picture of the system’s organisation and how it behaves by presenting flowcharts.
- Systems often consist of few modules that each specialise in specific functions.
- United, they form the functions in a system and each part is built as a module.
- Interfaces are points where parts of a system communicate with each other.
- Data management includes organising how information and data are processed and stored.

- ⇒ **Availability:** Creates assurance that a service will be operational all day, every day.
- ⇒ Minimises hazards to people and valuable assets.
- ⇒ **Fault Tolerance:** Maintains running programmes even when one part of the software fails.
- ⇒ The ability to update and add components to a product is easy and does not require major changes to the system.
- ⇒ Emergent Qualities analyses how vehicles’ parts and systems interact, especially when the car is forced to work under stress.

3.1.1. Start the process of defining the design.

At this point, you should prepare the system’s technology components, cheque for risks of obsolescence and make design notes.

3.1.2. Design the appearance of the site

Ensure that any missing design aspects are identified and fit with the structure and elements that were built in the architecture phase.

3.1.3. Cheque if There Are Other Options for Acquiring What You Need

Analyse various design choices and choose the most suitable ones. Should the outcome be developed, work on designing it and implementing it; if you need to acquire it, start acquiring it.

3.1.4. Manage the look and feel of the application.

Record the reasons for each design decision and adjust the design to help it be flexible as needed.

3.2 System Architecture

The system architecture outlines the structure, how the system behaves and the ways it can be viewed. The system description covers aspects of the design and explains how it meets nonfunctional requirements. System architecture is made up of several key factors.

- ⇒ **Performance:** Measures how data can be processed and how long it takes as the workload changes.
- ⇒ Scalability allows the system to handle an increased number of users or requests.

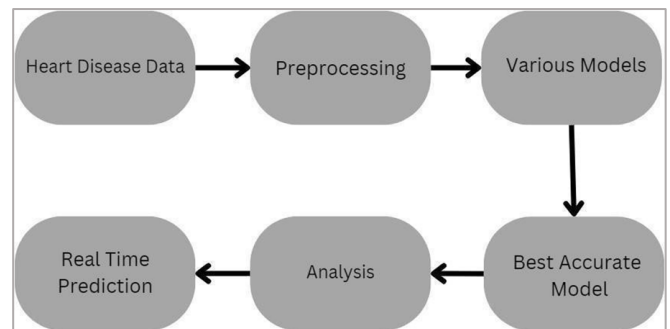


Fig1: System Architecture

3.3 UML Diagram

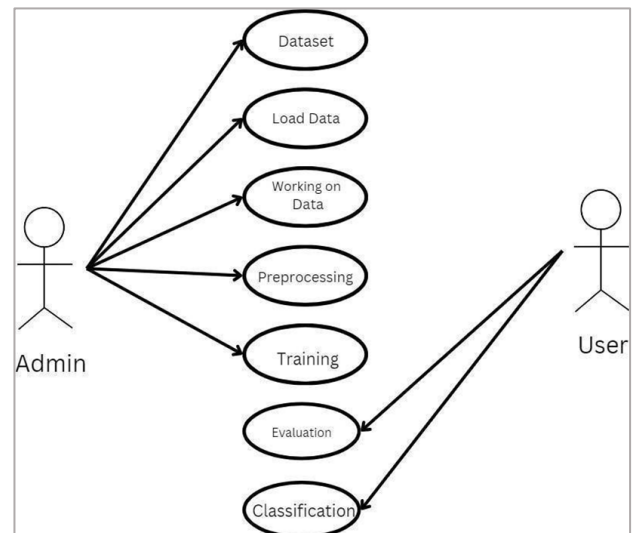


Fig2: UML Diagram

4. SYSTEM IMPLEMENTATION

4.1 About Implementation

Ensuring that a system is used successfully depends on implementation, as it gives users confidence in how well the new system functions. Here, the new application is deployed to replace the old one and the task is easy as long as the main parts of the system do not change.

During development, each programme is checked using valuable information to confirm it is working as expected. It is checked that the programmes work together according to the details described in the programme's design. Furthermore, all parts of the computer system are tested in their environment to make sure they meet the user's needs.

As soon as the system is developed, it is validated and accepted for use by the leaders. Simple steps are outlined in advance so that users can quickly use and navigate all of the system's features.

44.2 Code

4.2.1. Data Preprocessing & Exploration

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
df = pd.read_csv('Dataset.csv')

# Explore the dataset
print(df.head())
print(df.describe())
print(f>Data Types: {df.dtypes}")
print(df.info())

# Identify categorical and numerical columns
categories = [col for col in df.columns if df[col].dtype == 'object']
numerical = [col for col in df.columns if df[col].dtype != 'object']

# Check for missing values
print(df.isnull().sum())

# Visualize outliers using boxplots
outlier_columns = ['BMI', 'GenHlth', 'MentHlth', 'PhysHlth', 'Age', 'Education', 'Income']
for col in outlier_columns:
    if df[col].dtype in ['int64', 'float64']:
        plt.figure()
        df.boxplot(column=[col])
        plt.show()
```

4.2.2. Feature Engineering

```
# Map categorical values (e.g., Sex and Age)
df['Sex'] = df['Sex'].map({1: 'Masculine', 0: 'Feminine'})
age_ranges = {
    1: '18-24', 2: '25-29', 3: '30-34', 4: '35-39', 5: '40-44', 6: '45-49', 7: '50-54',
    8: '55-59', 9: '60-64', 10: '65-69', 11: '70-74', 12: '75-79', 13: '80 or above'
}
df['Age'] = df['Age'].map(age_ranges)

# One-hot encode categorical features
cat_features = ["Sex", "Age"]
df = pd.get_dummies(df, columns=cat_features)

# Split into features and target
target = df['HeartDiseaseorAttack']
features = df.drop('HeartDiseaseorAttack', axis=1)
```

44.2.3. Model Training and Hyperparameter Tuning

```

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LogisticRegression

# Split data into train and test sets
x_train, x_test, y_train, y_test = train_test_split(features, target, test_size=0.2,
random_state=20)

# Logistic Regression model with Grid Search for hyperparameter tuning
param_grid = {'C': [0.1, 1, 10], 'penalty': ['l1', 'l2']}
grid_search = GridSearchCV(LogisticRegression(), param_grid, cv=5, scoring='accuracy')
grid_search.fit(x_train, y_train)

# Get the best parameters and retrain the model
best_params = grid_search.best_params_
best_lreg = LogisticRegression(**best_params)
best_lreg.fit(x_train, y_train)

# Make predictions
y_pred = best_lreg.predict(x_test)
    
```

4.3 Results

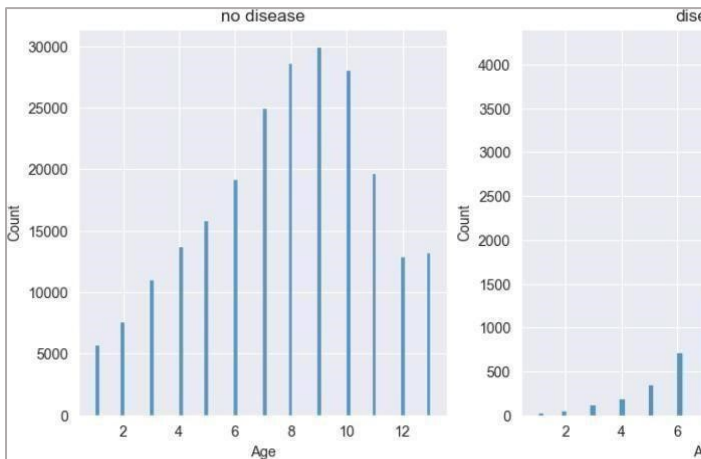


Fig3: Age vs Disease Relation

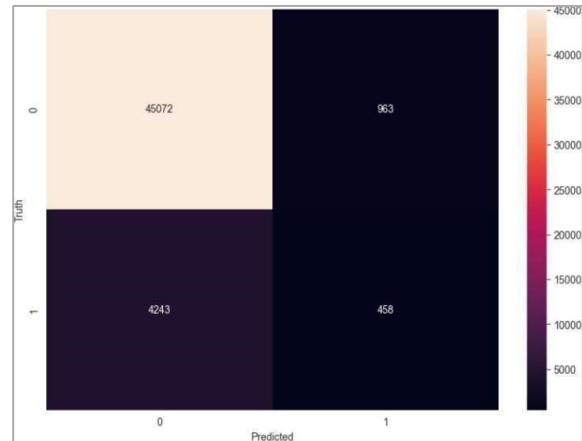


Fig4: ConfusionMatrix

5. TESTING

5.1 Test Cases

Test Case Id	Test Case	Expected Behavior	Exhibiting Behavior	Result
1	Data Collection	Action Performed	Action Performed	Pass
2	Check for Null Values	Action Performed	Action Performed	Pass
3	Outliers Detected	Error	Action Performed	Fail
4	Checked for Missing Values after preprocessing	Error	Action Performed	Fail
5	Get Results	Action Performed	Action Performed	Pass
6	Predictions	Action Performed	Action Performed	Pass

TABLE5: TEST-CASES

6. CONCLUSION

The project managed to resolve the problems of health-related data analysis and prediction by using a method that involved preparing data, examining it, engineering useful features, developing models, evaluating them and providing

a visual and easy-to-use interface. We were able to spot the connections between several health characteristics and the main health outcomes because of data visualization and EDA. Using these procedures, I was able to see how data was distributed and found connections and trends within the data. Furthermore, methods to find outliers were used to ensure that the data kept its integrity.

Logistic Regression, Decision Trees, Random Forest, Gaussian Naive Bayes and K-Nearest Neighbours (KNN) accurately predicted the outcomes in health problems. The Logistic Regression, Random Forest and KNN models had an accuracy of 90% and the Decision Trees model gave 85% accuracy, whereas the Gaussian Naive Bayes model achieved 61% accuracy. To judge the outcomes, confusion matrices and ROC curves helped by displaying important statistics such as accuracy, precision, recall and the AUC.

Furthermore, using Streamlit allowed users to interact easily with the models and data. The platform made it possible for users to view the data, notice connections between various features, notice unusual data and judge how the model was doing. It made it much easier for users to grasp health information and data presented in the app.

ACKNOWLEDGEMENT

I am thankful to the Management of Amrita Sai Institute of Science and Technology for giving me an opportunity to work with his project.

I would like to thank **Dr. M. Sasidhar**, Principal, Amrita Sai institute of science and technology, for his constant encouragement and support during the progress of this work.

I am deeply grateful to **Dr. P. Chiranjeevi**, Professor and Head of the Department, for his valuable guidance and consistent support during the course of the project.

A special note of thanks to my internal guide, **Mrs. Bandi. Suneetha (M.Tech)**, for her exceptional guidance, constant motivation, and continuous encouragement, which played a crucial role in the successful completion of this project.

UMA MAHESWARA RAO

REFERENCES

- [1] M. R. Miller, *B2B Digital Marketing: Using the Web to Market Directly to Businesses*, Que Publishing, 2012.
- [2] A. W. Baur, "Harnessing the social web to enhance insights into people's opinions in business, government and public administration," *Information Systems Frontiers*, vol. 19, pp. 231-251, 2017.
- [3] B. Kovács, "Five Is the Brightest Star. But by how Much? Testing the Equidistance of Star Ratings in Online Reviews," *Organizational Research Methods*, 2024, doi: 10.1177/10944281231223412.
- [4] E. Sadler-Smith and E. Shefy, "The intuitive executive: Understanding and applying 'gut feel' in decision-making," *Academy of Management Perspectives*, vol. 18, no. 4, pp. 76-91, 2004.
- [5] S.-i. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4-5, pp. 185-196, 1993.
- [6] A. Aizawa, "An information-theoretic perspective of tf-idf measures," *Information Processing & Management*, vol. 39, no. 1, pp. 45-65, 2003.
- [7] Z. Singla, S. Randhawa, and S. Jain, "Sentiment analysis of customer product reviews using machine learning," in *Proc. 2017 International Conference on Intelligent Computing and Control (I2C2)*, IEEE, 2017.
- [8] R. S. Jagdale, V. S. Shirsat, and S. N. Deshmukh, "Sentiment analysis on product reviews using machine learning techniques," in *Cognitive Informatics and Soft Computing: Proceedings of CISC 2017*, Springer Singapore, 2019.
- [9] T. U. Haque, N. N. Saber, and F. M. Shah, "Sentiment analysis on large scale Amazon product reviews," in *Proc. 2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, IEEE, 2018.
- [10] S. Wassan et al., "Amazon product sentiment analysis using machine learning techniques," *Revista Argentina de Clínica Psicológica*, vol. 30, no. 1, pp. 695-704, 2021.
- [11] R. Rosipal, L. J. Trejo, and B. Matthews, "Kernel PLS-SVC for linear and nonlinear classification," in *Proc. 20th*

International Conference on Machine Learning (ICML-03), 2003.

[12] M. S. Dinesh Boddapati et al., "YouTube Comment Analysis Using Lexicon Based Techniques," in *International Conference on Cognitive Computing and Cyber Physical Systems*, Cham: Springer Nature Switzerland, 2022.

[13] V. Kecman, "Support vector machines – an introduction," in *Support Vector Machines: Theory and Applications*, Springer, Berlin Heidelberg, 2005, pp. 1-47.

[14] P.-Y. Hao, "New support vector algorithms with parametric insensitive/margin model," *Neural Networks*, vol. 23, no. 1, pp. 60-73, 2010.

[15] A. Wendland, M. Zenere, and J. Niemann, "Introduction to text classification: impact of stemming and comparing TF-IDF and count vectorization as feature extraction technique," in *Systems, Software and Services Process Improvement: 28th European Conference, EuroSPI 2021, Krems, Austria, September 1-3, 2021, Proceedings 28*, Springer International Publishing, 2021.