Cardio Care.ai

Heart Disease Prediction using Machine Learning

- Akash Duttachowdhury

Project Overview

This project aims to predict whether a patient is at risk of heart disease using various machine learning models such as Logistic Regression, K-Nearest Neighbors (KNN), Naive Bayes, Decision Tree, and Support Vector Machine (SVM). The predictions are based on a dataset containing various health-related attributes such as age, sex, cholesterol levels, and more.

The project consists of a Flask web application where users can input their health data, and the trained models will predict whether they are at risk of heart disease. The results are displayed with the model's accuracy and prediction outcome.

Table of Contents

Project Structure

The project is structured as follows:

```
├── app.py                    # Main Flask application
├── heart_disease_model.ipynb # Jupyter notebook for
training models
├── templates/
│   ├── main.html             # Web form for user input
│   ├── result.html           # Displays the prediction
results
├── static/
│   ├── style.css             # CSS styling for the web
application
├── logistic_regression.model  # Pre-trained Logistic
Regression model
├── naive_bayes.model         # Pre-trained Naive Bayes
model
├── knn.model                 # Pre-trained K-Nearest
Neighbors model
├── dt.model                  # Pre-trained Decision Tree
model
├── svm.model                 # Pre-trained SVM model
```

Installation

To get started with the project, follow the instructions below:

Clone the repository:

```
git clone https://github.com/akash-duttac/heart-disease-prediction.git
cd heart-disease-prediction
```

Install dependencies:

It is recommended to use a virtual environment to manage dependencies.

1. Create a virtual environment:

```
python3 -m venv venv
source venv/bin/activate  # On Windows use `venv\Scripts\activate`
```

2. Install required Python packages:

```
pip install -r requirements.txt
```

Run the application

Once the dependencies are installed, you can start the Flask web application by running the following command:

```
python app.py
```

The application will be hosted on http://127.0.0.1:5000/. You can open this URL in your browser to access the Heart Disease Prediction form.

Usage

1. **Enter your health details:** Fill in the form with your personal health information, such as age, cholesterol levels, blood pressure, and more.
2. **Submit the form:** After filling in the details, click on the “Submit” button.
3. **View Prediction Results:** The results page will show predictions from the five machine learning models, indicating whether you are at risk of heart disease and the accuracy of each model.

Models

The following machine learning models have been trained and integrated into the web application for predicting heart disease:

Models	Accuracy
Logistic Regression	81.32%
Naive Bayes	82.42%

Models	Accuracy
K-Nearest Neighbours	59.34%
Decision Tree	84%
Support Vector Machine	82.42%

These models were trained using a dataset containing the following features:

- Age: Age of the patient
- Sex: Sex (1=male, 0=female)
- Cp: Chest pain type (1: typical angina, 2: atypical angina, 3: non-anginal pain, 4: asymptomatic)
- Trestbps: Resting blood pressure (in mm Hg)
- Chol: Colesterol level (in mg/dl)
- Fbs: Fasting blood sugar (1 = true, 0 = false)
- Restecg: Resting ecocardiogram results
- Thalach: Maximum heart rate achieved
- Exang: Exercise-induced angina (1 = yes, 0 = no)
- Old peak: ST depression induced by exercise relative to rest
- Slope: Slope of the peak exercise ST segment
- Ca: Number of major vessels coloured by fluoroscopy (0-3)
- Thal: Thallium heart scan value (0 = no value, 1 = normal, 2 = fixed defect, 3 = reversible defect)
- Target: Target variable (1 = presence of heart disease, 0 = absence)

Dataset

The dataset used for training the models is publicly available in the UCI Machine Learning Repository. It contains health-related data points used to predict heart disease.

Web Application

The web application is built using Flask and consists of the following components:

1. main.html: Contains the form where users can input their details such as age, sex, blood pressure, cholesterol level, etc.
2. result.html: Displays the prediction results from the different models, including whether the user is at risk of heart disease and the accuracy of each model.
3. style.css: Provides the styling for the web application using responsive design.

Future Enhancements

- **Model Improvements:** Experiment with other machine learning algorithms and fine-tune the models to improve accuracy.
- **Model Explainability:** Add features to explain the model's predictions, such as SHAP values or LIME.
- **Deployment:** Deploy the application on a cloud platform like Heroku or AWS for easy access.
- **User Authentication:** Add user authentication to allow multiple users to access and track their heart health predictions over time.

Example Screenshots