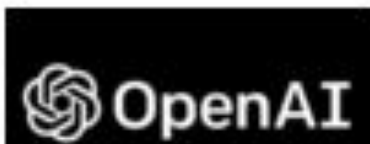


+ Yargi, Emily, Anna and Archit +

Group 6

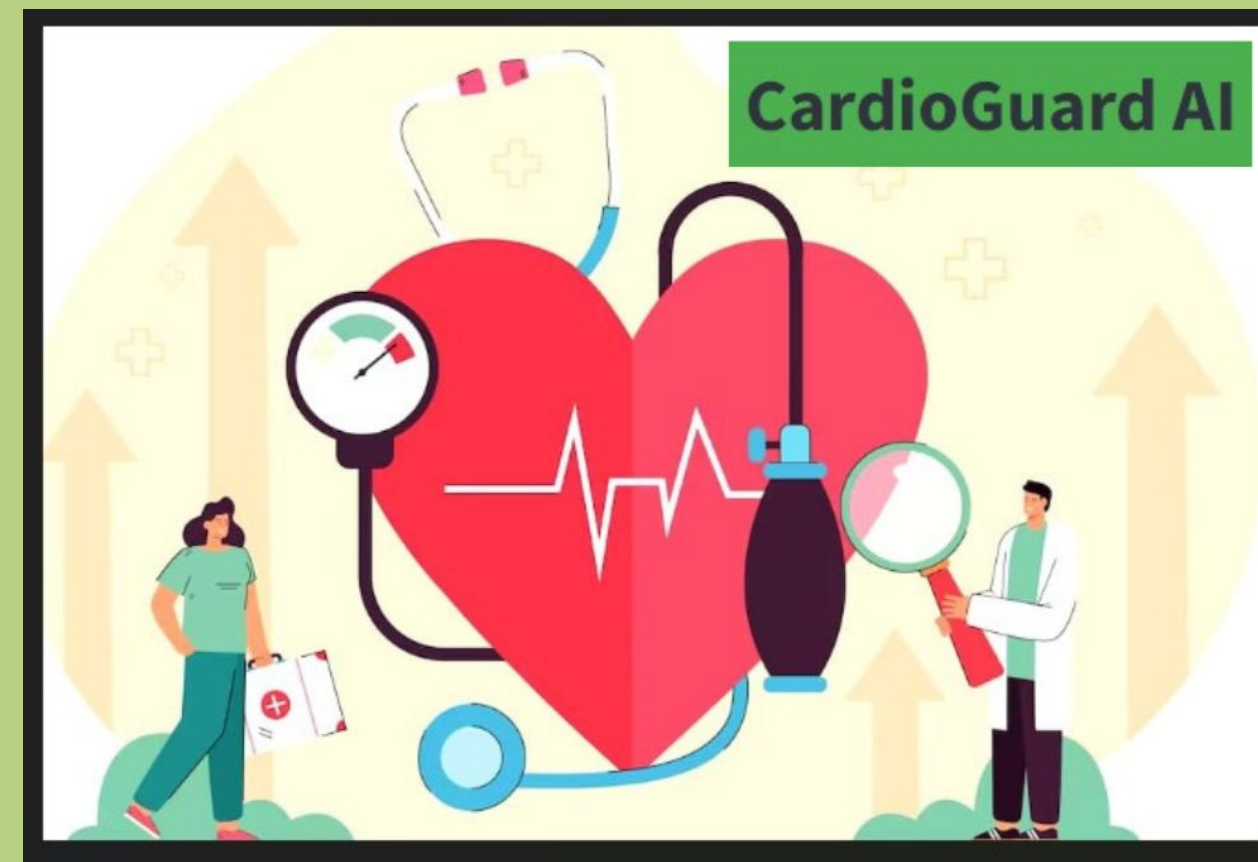
CardioGuard AI: Predicting Heart Disease Risk & Providing AI Lifestyle Advice



Life-Style
Advices
Powered by
OpenAI API



Health APPs by
ChatGPT



User Questions
Answered by
ChatBOT



Answered
by ChatBOT



Target Audience

Healthcare Professionals

- **Early Detection Tool:** Assists in identifying potential heart disease risk.
- **Informed Interventions:** Guides on when to consider further medical evaluation.

Heart health conscious people

- **Educational Insight:** Provides preliminary assessment of heart disease risk factors.
- **Guided Advice:** Offers recommendations for lifestyle adjustments and seeking professional care.





Data Sourcing



Data Source

k

- **Data Repository Name:** Kaggle
- **Dataset Name:** Heart Failure Prediction Dataset
- **URL:** [Heart Failure Prediction Dataset](#)

Key Dataset Features

k

- **Records:** 918
- **Features:** 12
- **Target:** Heart Disease



Data Cleaning, Modelling and Preprocessing



```
# heart_data = pd.read_sql('SELECT * FROM heart_data', engine)
heart_data = pd.read_sql('SELECT * FROM heart_data', engine)
# Display sample data
heart_data.head()
```

	age	sex	chestpaintype	restingbp	cholesterol	fastingbs	restingecg	maxhr	exerciseangina	oldpeak	st_slope	heartdisease
0	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
1	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
2	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
3	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
4	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0

Our dataset was very clean, limited cleaning was needed. However categorical features were mapped as a preprocessing step.

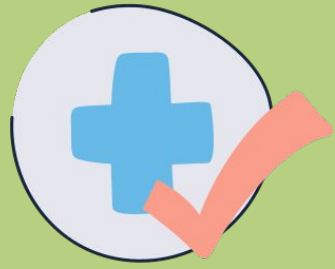
```
# Mapping for categorical features
sex_mapping = {'M': 1, 'F': 0}
cp_mapping = {'NAP': 0, 'ATA': 1, 'ASY': 2, 'TA': 3}
bs_mapping = {'Normal': 0, 'ST': 1}
ecg_mapping = {'Normal': 0, 'ST': 1}
angina_mapping = {'N': 0, 'Y': 1}
slope_mapping = {'Up': 0, 'Flat': 1, 'Down': 2}

# Convert categorical columns to numeric in the original dataset
heart_data['sex'] = heart_data['sex'].map(sex_mapping)
heart_data['chestpaintype'] = heart_data['chestpaintype'].map(cp_mapping)
heart_data['fastingbs'] = heart_data['fastingbs'].map(bs_mapping)
heart_data['restingecg'] = heart_data['restingecg'].map(ecg_mapping)
heart_data['exerciseangina'] = heart_data['exerciseangina'].map(angina_mapping)
heart_data['st_slope'] = heart_data['st_slope'].map(slope_mapping)
```

```
# Looking at heartdisease value counts for binning
application_counts = heart_data.heartdisease.value_counts()
application_counts
```

```
heartdisease
1    508
0    410
Name: count, dtype: int64
```

We have a good split between instances that indicate heart disease and instances that indicate no heart disease.



Use of Elephant SQL - Cloud based



Elephant SQL features

- Cloud SQL software
- Free plan available
- Administers tasks of PostgreSQL, including installation, upgrades and backup handling
- Allows us to have user interaction and cloud connection






Use in code

```
# ElephantSQL database URI
database_uri = 'postgresql://caqkfcsy:b41wIKaMBjOtBNDe1Y6EL_Xz-uoGqvU@salt.db.elephantsql.com/caqkfcsy'

# Create the database engine
engine = create_engine(database_uri)

# Load the dataset from PostgreSQL
heart_data = pd.read_sql('SELECT * FROM heart_data', engine)

# Rename columns based on PostgreSQL column names
```

Server	salt.db.elephantsql.com (salt-01)	A
Region	amazon-web-services::us-east-1	
Created at	2024-01-30 01:28 UTC+00:00	
User & Default database	caqkfcsy	Reset
Password	***  	 Rotate password
URL	postgres://caqkfcsy:***@salt.db.elephantsql.com/caqkfcsy  	
Current database size	120 KB	
Max database size	20 MB	

Machine learning models

1st

Random Forest Classifier

Highest accuracy score: 0.86

Reasoning for testing:

- Gathers forecasts from each tree and predicts based on majority of votes
- Meta-estimator, enhancing models accuracy and prevents over-fitting.

2nd

Gradient Boosting Classifier

Highest accuracy score: 0.83

Reasoning for testing:

- Powerful boosting algorithm
- Updates weights based on gradients
- Good for classification improvement

3rd

SVM

Highest accuracy score: 0.82

Reasoning for testing:

- Mostly utilized for classification difficulties
- Creates three dimension classification model

Machine learning model optimization

1st

Random Forest Classifier

Highest accuracy score: 0.86

Optimized over three times

Final optimization highest when not using standard scaler (0.86 vs 0.82)

Chosen model for app

```
# Create a random forest classifier
rf_model2 = RandomForestClassifier(n_estimators=400, random_state=78)
```

```
Accuracy Score : 0.8206521739130435
Classification Report
              precision    recall  f1-score   support

     0       0.74       0.88       0.80        77
     1       0.90       0.78       0.83       107

 accuracy          0.82          184
 macro avg         0.82          184
 weighted avg      0.83          184
```

```
# Create a random forest classifier
rf_model = RandomForestClassifier(n_estimators=200, random_state=42, max_depth=10, min_samples_split=2, min_samples_leaf=1)
```

```
Accuracy Score : 0.8206521739130435
Classification Report
              precision    recall  f1-score   support

     0       0.74       0.88       0.80        77
     1       0.90       0.78       0.83       107

 accuracy          0.82          184
 macro avg         0.82          184
 weighted avg      0.83          184
```

The same features were then ran again with not standard scaler for a third optimization and achieved a 0.86 accuracy score.



Front-end features

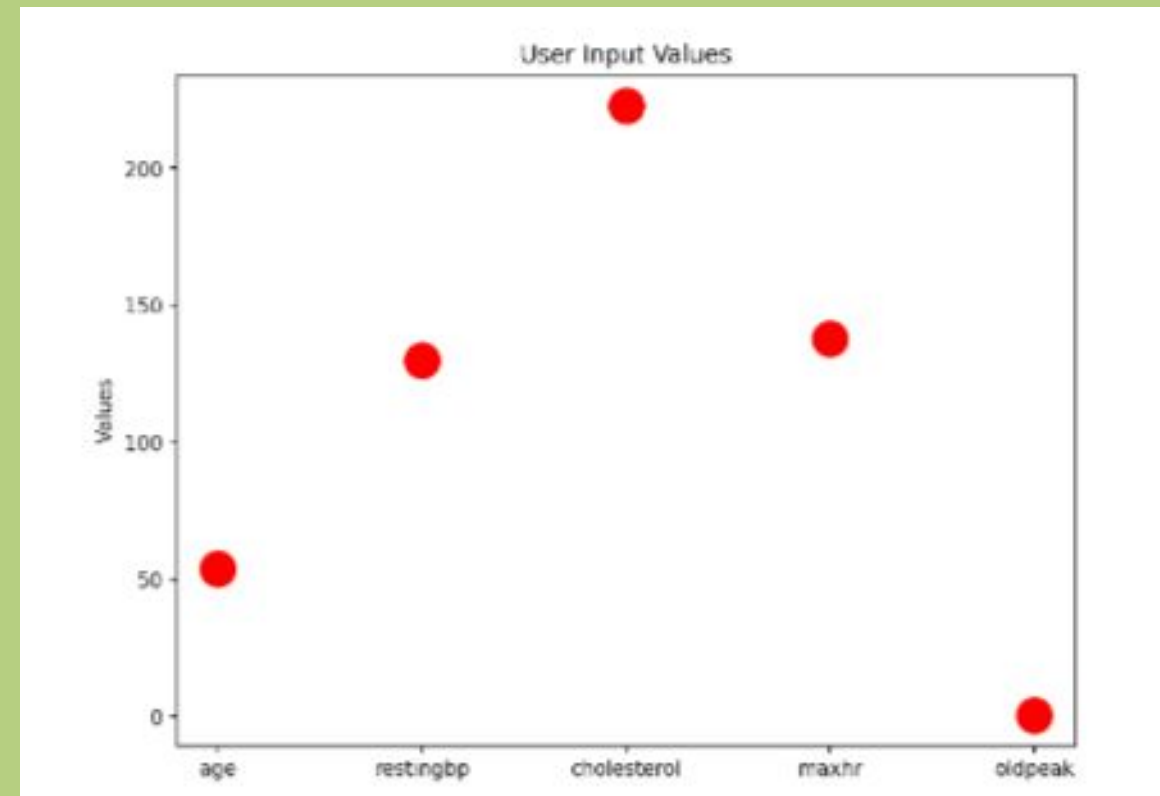
matplotlib

Streamlit

Dynamic plotting and
responsive design

Interactive sidebar for user
interaction

Streamlit



User Input Features

Age

28 54 77

Resting Blood Pressure

0 130 200

Cholesterol

0 223 603

Max Heart Rate

60 138 202

Oldpeak

-2.60 0.60 6.20

Sex

M

Chest Pain Type

NAP

Fasting Blood Sugar

Normal

Resting ECG

import streamlit as st

install streamlit by running: pip install streamlit

to run: streamlit run app.py

```
# Scatter plot for user input
ax1.scatter(user_input_labels, user_input_values, color='red', marker='o', s=3*100) # Adjust the size with s parameter
ax1.set_ylabel('Values')
ax1.set_title('User Input Values')

# Bar chart for numeric features in the dataset
numeric_features = ['age', 'restingbp', 'cholesterol', 'maxhr', 'oldpeak']
ax2.barh(numeric_features, heart_data[numeric_features].mean(), color='red')
ax2.set_xlabel('Mean Values')
ax2.set_title('Numeric Features in Dataset')

st.pyplot(fig)
```



```
# Function to generate advice using OpenAI API
def generate_advice(user_input):
    prompt = f"Given the user input: {user_input}, provide advice to reduce the risk of heart disease."

    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo-1106",
        messages=[
            {"role": "system", "content": "You are a helpful assistant."},
            {"role": "user", "content": prompt}
        ],
        max_tokens=150,
        n=1
    )

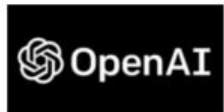
    advice = response['choices'][0]['message']['content']
    return advice
```



Lifestyle Advice:

To reduce the risk of heart disease, here are some general recommendations based on the given input:

1. Maintain a healthy weight: It's important to keep your body weight within a healthy range by eating a balanced diet and engaging in regular physical activity.
2. Monitor blood pressure: Keep track of your blood pressure levels and consult with a healthcare professional to manage and maintain healthy levels.
3. Consider cholesterol levels: Pay attention to your cholesterol levels and make adjustments to your diet and lifestyle if needed to keep them within a healthy range.
4. Manage stress: Incorporate stress-reducing activities such as meditation, deep breathing, or yoga to help manage stress levels.
5. Follow a healthy diet: Aim to consume a diet that is rich in fruits

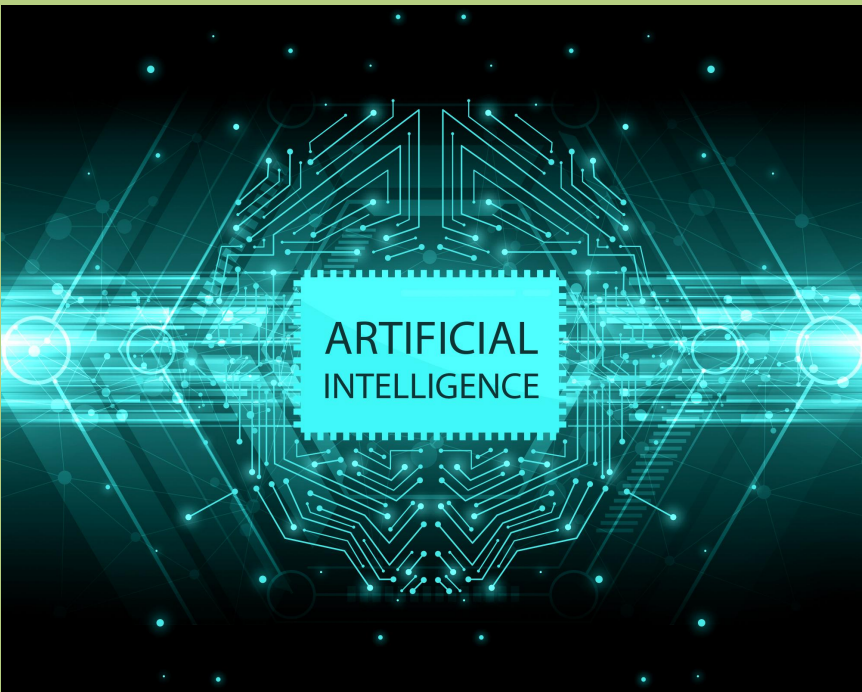


Powered by OpenAI



Life-Style
Advices
Powered by
OpenAI API

AI powered User Interaction



ChatGPT Response:

Sure! Here are some of the best apps that can help you stay healthy and count your walking distance:

1. **Step Tracker:** Step Tracker is a simple pedometer app that tracks your walking distance, steps taken, and calories burned. It's a great tool for tracking your daily physical activity.
2. **MapMyWalk:** MapMyWalk uses GPS to track and map your walking route. It also provides audio feedback on your pace, distance, and calorie burn, making it a great



Lifestyle Advice from
ChatGPT



Health APPs by
ChatGPT

```
# Function to generate ChatGPT response
def generate_chatgpt_response(prompt):
    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo-1106",
        messages=[
            {"role": "system", "content": "You are a helpful assistant."},
            {"role": "user", "content": prompt}
        ],
        max_tokens=100,
        n=1
    )

    return response['choices'][0]['message']['content']
```



User Questions
Answered by
ChatBOT

Chat with Hearty: Your AI Support Companion:

Hello! My name is Hearty, your dedicated support companion! Celebrate your journey towards transformative lifestyle changes and remarkable achievements. How may I assist you on this empowering path? I can provide personalized advice tailored just for you.

You:



Answered
by ChatBOT



Live Demonstration

CardioGuard AI

Predicting Heart Disease Risk & Providing Lifestyle Advice using AI

Welcome to CardioGuard AI: Your Personalized Health Assistant!

Empower yourself with personalized predictions for heart disease risk and actionable lifestyle advice, all powered by cutting-edge AI technology.

To get started, simply enter your information in the sidebar, and let us guide you through insightful predictions and tailored recommendations.

Don't forget to engage with our friendly chatbot, Hearty, who's here to answer your questions and provide additional support along your health journey!



Healthy Hearts
with
CardioGuard AI



Life-Style
Advices
Powered by
OpenAI API



Health APPs by
ChatGPT



User Questions
Answered by
ChatBOT



Thanks!



+ Yargi, Emily, Anna and Archit +

2024 CardioGuard AI. All rights reserved.

Many thanks to our TA`s: Tom-Jordan-Yash-Mohammed and Sureer
for their effort and great support during our data journey!

