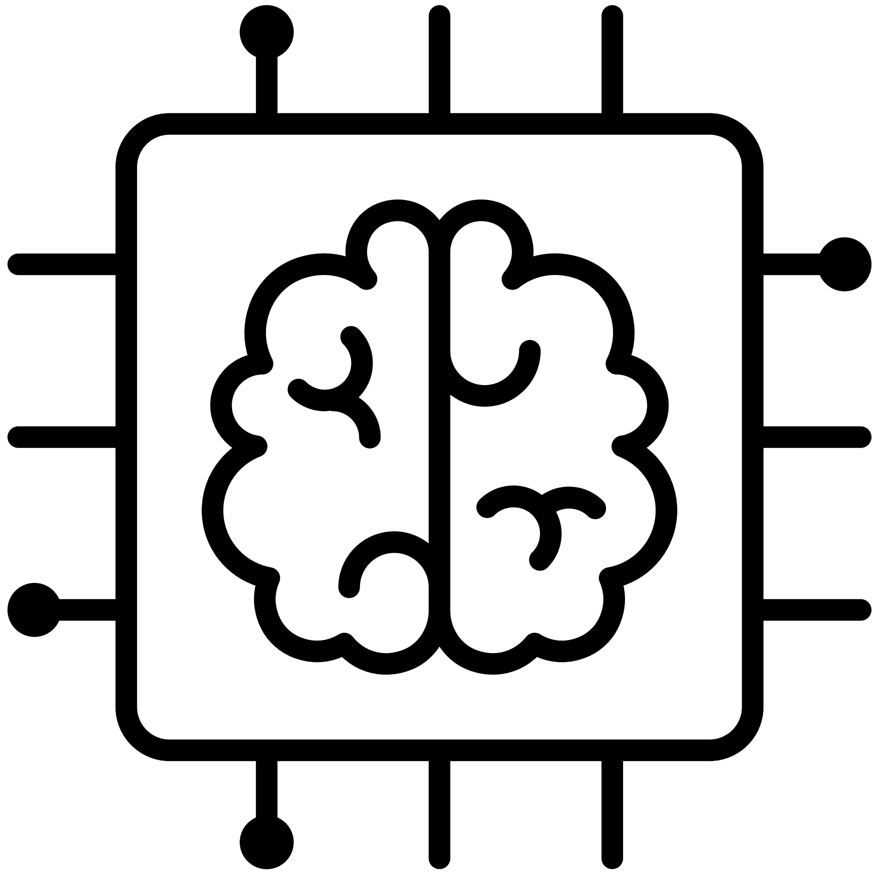
Utilizing The Power of Machine Learning in Healthcare

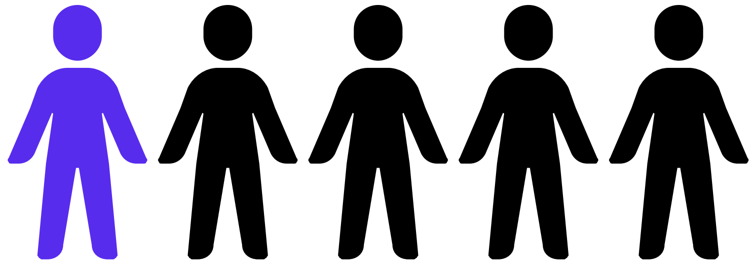


By: Yogesh Seenichamy

Topics

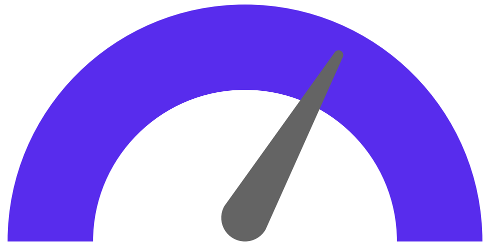
1. Heart Failure Background
2. Neural Networks
3. Application of Neural Networks
4. Sample Deployment to Website

Heart Failure Background



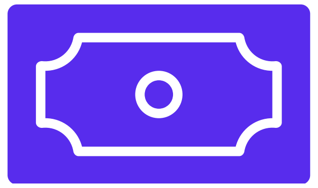
1 out 5

Deaths in the United States are Caused by Heart Disease



Limited Accuracy

for Traditional Methods of Heart Failure Prediction



$30000

Average Cost in the United States for Treating Heart Failure

Current Limitations of Diagnosis

LEADING CAUSE OF DEATH

Heart disease is the leading cause of death in the United States, accounting for 655,000 deaths each year

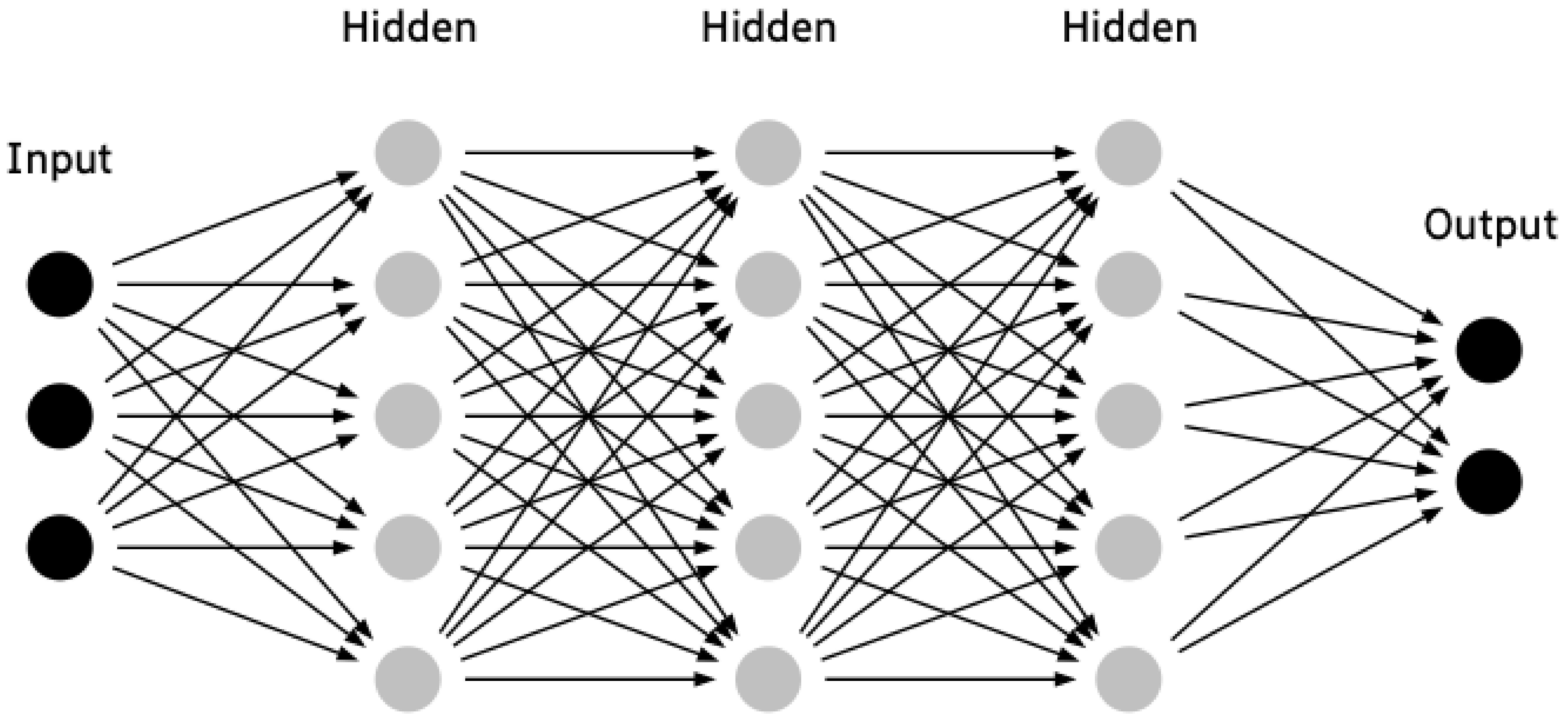
TRADITIONAL SHORTCOMINGS

Traditional methods often rely on subjective clinical assessment, which can lead to inconsistencies and errors in diagnosis

COSTLY AND TIME-CONSUMING

In addition, these methods can be costly and time-consuming, which can create delays in diagnosis and treatment

Neural Networks



kaggle

Create

Home

Competitions

Datasets

Models

Code

Discussions

Learn

More

Search

Sign In

Regist

FEDESORIANO · UPDATED 2 YEARS AGO

2138

New Notebook

Download (9 kB)

Heart Failure Prediction Dataset

11 clinical features for predicting heart disease events.

Data Card Code (750) Discussion (20)

About Dataset

Similar Datasets

- Hepatitis C Dataset: LINK
- Body Fat Prediction Dataset: LINK
- Cirrhosis Prediction Dataset: LINK
- Stroke Prediction Dataset: LINK
- Stellar Classification Dataset - SDSS17: LINK
- Wind Speed Prediction Dataset: LINK
- Spanish Wine Quality Dataset: LINK

Context

Cardiovascular diseases (CVDs) are the number 1 cause of death globally, taking an estimated 17.9 million lives each year, which accounts for 31% of all deaths worldwide. Four out of 5CVD deaths are due to heart attacks and strokes, and one-third of these deaths occur prematurely in people under 70 years of age. Heart failure is a common event caused by CVDs and this dataset contains 11 features that can be used to predict a possible heart disease.

People with cardiovascular disease or who are at high cardiovascular risk (due to the presence of one or more risk factors such as hypertension, diabetes, hyperlipidaemia or already established disease) need early detection and management wherein a machine learning model can be of great help.

View more

Usability

10.00

License

Database: Open Database, Con...

Expected update frequency

Never

cookies on Kaggle to deliver our services, analyze web traffic, and improve your experience on the site. By using Kaggle, you agree to our use of cookies. Got it Learn m

https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction

Dataset Information

1. Age: age of the patient [years]
2. Sex: sex of the patient [M: Male, F: Female]
3. ChestPainType: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]
4. RestingBP: resting blood pressure [mm Hg]
5. Cholesterol: serum cholesterol [mm/dl]
6. FastingBS: fasting blood sugar [1: if FastingBS > 120 mg/dl, 0: otherwise]
7. RestingECG: resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]
8. MaxHR: maximum heart rate achieved [Numeric value between 60 and 202]
9. ExerciseAngina: exercise-induced angina [Y: Yes, N: No]
10. Oldpeak: oldpeak = ST [Numeric value measured in depression]
11. ST_Slope: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]
12. HeartDisease: output class [1: heart disease, 0: Normal]

TensorFlow

colab

```python
#Load in Dataset
import pandas as pd

dataset = pd.read_csv("heart.csv")
dataset.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

Data Preprocessing

Data Preprocessing

```
[20] #Convert the categorical columns into numerical columns
     restingECG = pd.get_dummies(dataset['RestingECG'], drop_first=True)
     chestPainType = pd.get_dummies(dataset['ChestPainType'], drop_first=True)
     exerciseAngina = pd.get_dummies(dataset['ExerciseAngina'], drop_first=True)
     st_slope = pd.get_dummies(dataset['ST_Slope'], drop_first=True)
     sex = pd.get_dummies(dataset['Sex'], drop_first=True)
```

```
[21] #Add newly created numerical columns and remove categorical columns
     dataset = dataset.drop(["Sex","ST_Slope","ExerciseAngina","RestingECG","ChestPainType"], axis=1)
     dataset = pd.concat([dataset, restingECG, chestPainType, exerciseAngina, st_slope, sex], axis=1)
     dataset.head()
```

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease	Normal	ST	ATA	NAP	TA	Y	Flat	Up	M
0	40	140	289	0	172	0.0	0	1	0	1	0	0	0	0	1	1
1	49	160	180	0	156	1.0	1	1	0	0	1	0	0	1	0	0
2	37	130	283	0	98	0.0	0	0	1	1	0	0	0	0	1	1
3	48	138	214	0	108	1.5	1	1	0	0	0	0	1	1	0	0
4	54	150	195	0	122	0.0	0	1	0	0	1	0	0	0	1	1

```
[22] #Get Input and Output Arrays
     X = dataset.drop("HeartDisease", axis=1)
     Y = dataset["HeartDisease"]
```

```
[24] #Split input and output into training and testing data
     from sklearn.model_selection import train_test_split

     X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = .2, random_state = 0) #Using 20% of dataset for testing accuracy
```

```
[25] #Feature Scaling
     from sklearn.preprocessing import StandardScaler

     scaler = StandardScaler()
     X_train = scaler.fit_transform(X_train)
     X_test = scaler.transform(X_test)
```

Model Creation

Model Creation

```
[28] import keras
     from keras.models import Sequential
     from keras.layers import Dense

     model = Sequential()

     model.add(Dense(units = 16, activation="relu", input_dim=X_train.shape[1]))
     model.add(Dense(units = 16, activation="relu"))
     model.add(Dense(units = 1, activation="sigmoid"))

     model.compile(optimizer = "adam", loss = "binary_crossentropy", metrics = ["accuracy"])
```

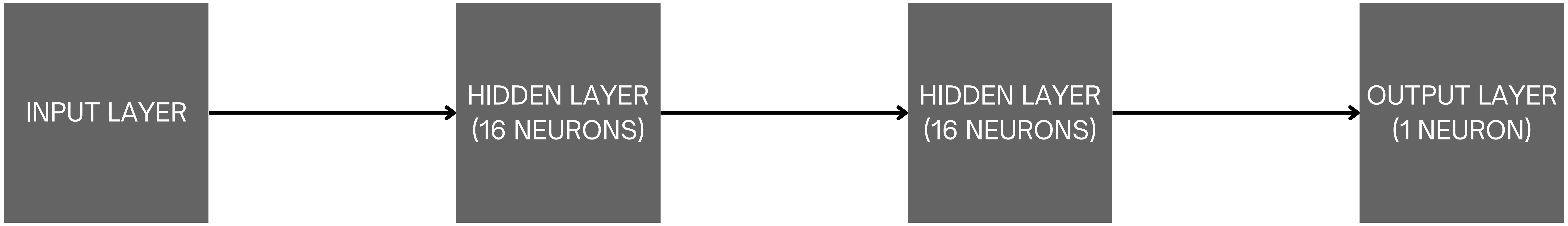
```
[29] history = model.fit(X_train, Y_train, validation_split = 0.33, batch_size = 10, epochs = 100)
```

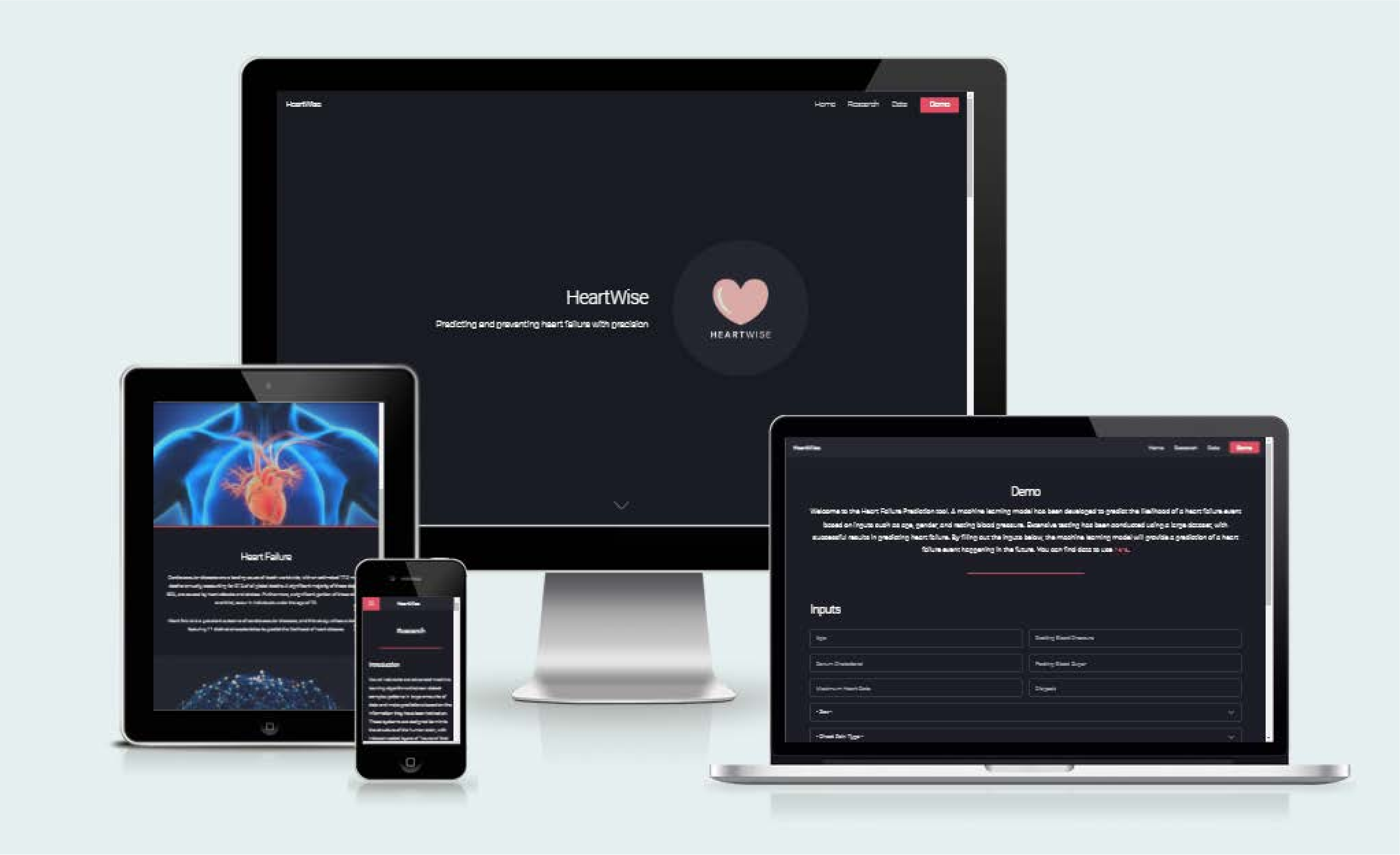
```
[30] predictions = model.predict(X_test)
     predictions = (predictions > 0.5)

     from sklearn.metrics import accuracy_score
     score = accuracy_score(predictions, Y_test)

     score
```

```
6/6 [==============================] - 0s 7ms/step
0.7880434782608695
```





Home
Research
Data
Demo
HeartWise
Predicting and preventing heart failure with precision
HEARTWISE
Heart Failure
Research
Demo
Inputs

Thank You!

Code Link:
https://colab.research.google.com/drive/1Jq9pszNpVk8qW9CrwGwapHB2j4xipgcH?usp=sharing