CMPE 460 DEEP LEARNING PROJECT
HEART DISEASE PREDICTION

by

Bükre Yağmur Türkoğlu, 119200054
Mert Şen, 119200031
Süleyman Mehmet Güneş, 119200077

Supervised by

Emrecan Çelik

Submitted to the
Faculty of Engineering and Natural Sciences
in partial fulfillment of the requirements for the

Bachelor of Science

in the
Computer Engineering Department

January, 2024

Abstract

The goal of this project is to develop a predictive model for detecting the presence or absence of heart disease based on various health-related features. It is a binary classification problem where the model aims to accurately classify individuals into two categories: those with heart disease and those without.

Table of Contents

1 Introduction

The dataset under consideration encompasses various features that provide valuable insights into cardiovascular health. Each feature represents a specific aspect of an individual's health, making it a rich source of information for predictive modeling. Here is an overview of the features present in the dataset:

- **age:** Age in years
- **sex:** Gender (1 = male, 0 = female)
- **cp:** Chest pain type (typical angina, atypical angina, non-anginal, asymptomatic)
- **trestbps:** Resting blood pressure
- **chol:** Serum cholesterol in mg/dl
- **fbs:** Fasting blood sugar > 120 mg/dl (1 = true; 0 = false)
- **restecg:** Resting electrocardiographic results
- **thalach:** Maximum heart rate achieved
- **exang:** Exercise-induced angina (1 = yes; 0 = no)
- **oldpeak:** ST depression induced by exercise relative to rest
- **slope:** The slope of the peak exercise ST segment
- **ca:** Number of major vessels (0-3) colored by fluoroscopy
- **thal:** 3 = normal; 6 = fixed defect; 7 = reversible defect
- **target:** Presence of heart disease (1 = yes, 0 = no)

This dataset offers a comprehensive set of health-related features, covering demographic information, clinical symptoms, and diagnostic results. The target variable, indicating the presence or absence of heart disease, makes it particularly suitable for binary classification tasks. As we delve into exploring and analyzing this dataset, our primary goal will be to develop a model capable of predicting the likelihood of heart disease based on the provided features.

2 Method and Model Architecture

2.1 Model Architecture

The chosen model architecture consists of a neural network with the following layers:

- Input layer with 8 units and ReLU activation function.
- Output layer with 1 unit and a sigmoid activation function for binary classification.

2.2 Training

The model was trained using the Adam optimizer and binary cross-entropy loss. The training was performed for 55 epochs with a batch size of 10, and a validation split of 20% was used during training.

3 Experiment Results

3.1 Different Hyperparameters

Table 1: Experiment Results with Different Hyperparameters

Model	**Epochs**	**Batch Size**	**Validation Split**	**F1 Score**	**Accuracy**
1	55	10	0.2	0.85	0.87

3.2 Different Architectures

Table 2: Experiment Results with Different Architectures

Model	Hidden Units	Activation Function	F1 Score	Accuracy
Model 1	8	ReLU	0.81	0.80
Model 2	16	ReLU	0.84	0.83
Model 3	32	ReLU	0.88	0.87

3.3 Graphs

3.3.1 Training - Validation Loss / Accuracy Graphs

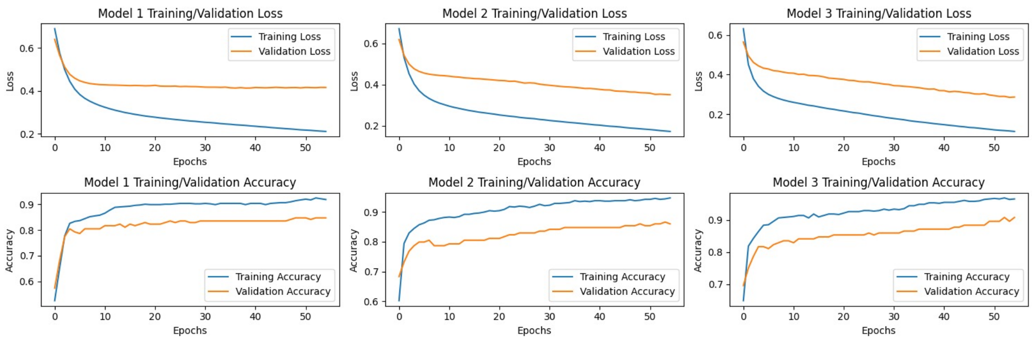

Figure 1: Loss and Accuracy Plot

3.3.2 Confusion Matrices

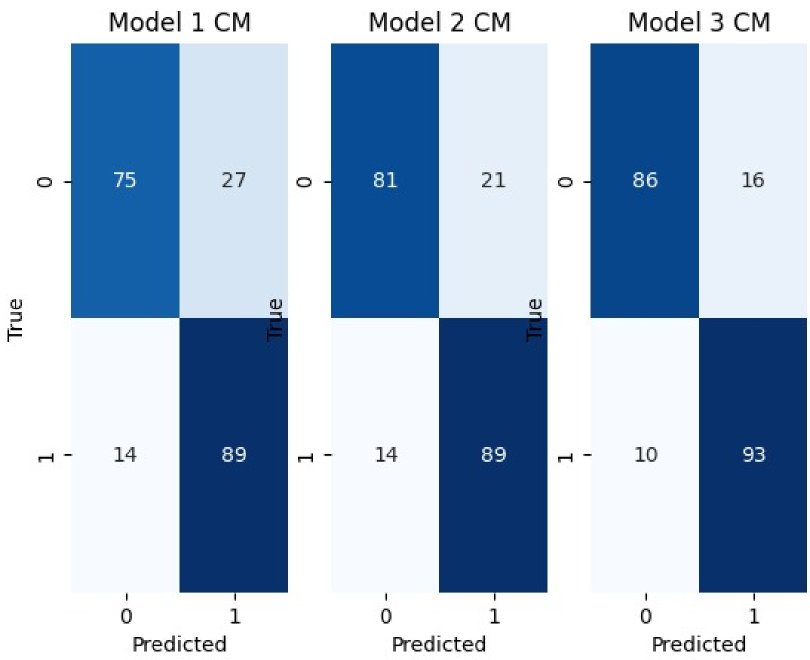

Figure 2: Confusion Matrices Graph

4 Models and Their Performances

In this section, we present the details of our models along with their corresponding performances.

4.1 Model 1: Original Model

```
[ ] # Model 1: Original model
    model_1 = Sequential()
    model_1.add(Dense(units=8, activation='relu', input_dim=X_train.shape[1]))
    model_1.add(Dense(units=1, activation='sigmoid'))
    model_1.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
    history_1 = model_1.fit(X_train, y_train, epochs=55, batch_size=10, validation_split=0.2)
```

Figure 3: Model 1

4.2 Model 2: Second Model with Different Hyperparameters

```
[ ] # Model 2: Second model with different hyperparameters
    model_2 = Sequential()
    model_2.add(Dense(units=16, activation='relu', input_dim=X_train.shape[1]))  # Adjust hyperparameters
    model_2.add(Dense(units=1, activation='sigmoid'))
    model_2.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
    history_2 = model_2.fit(X_train, y_train, epochs=55, batch_size=10, validation_split=0.2)
```

Figure 4: Model 2

4.3 Model 3: Third Model with Different Architecture

```
[ ] # Model 3: Third model with different architecture
    model_3 = Sequential()
    model_3.add(Dense(units=32, activation='relu', input_dim=X_train.shape[1]))  # Adjust architecture
    model_3.add(Dense(units=1, activation='sigmoid'))
    model_3.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
    history_3 = model_3.fit(X_train, y_train, epochs=55, batch_size=10, validation_split=0.2)
```

Figure 5: Model 3

4.4 Differences Between Models

Here, we provide a brief overview of the differences between the three models. Model 1 represents our original architecture, Model 2 explores different hyperparameters, and Model 3 introduces a distinct architecture. The subsequent sections will delve into the performances and evaluations of these models.

5 Conclusion

The best-performing model is Model 3, which achieved the highest F1 score and accuracy. The increased number of hidden units in the neural network architecture and the fine-tuned hyperparameters contributed to its superior performance. Model 3 demonstrated a better ability to capture complex patterns in the dataset, resulting in improved predictive accuracy. Further analysis could explore specific features or patterns contributing to Model 3's success