5-2023

Heart Disease Prediction Using Binary Classification

Virendra Sunil Devare
California State University - San Bernardino

HEARTH DISEASE PREDICTION USING BINARY CLASSIFICATION

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

by

Virendra Sunil Devare

May 2023

HEARTS DISEASE PREDICTION USING BINARY CLASSIFICATION

A Project

Presented to the

Faculty of

California State University,

San Bernardino

By

Virendra Sunil Devare

May 2023

Approved by:

Dr. Jennifer Jin, Advisor, School of Computer Science and Engineering

Dr. Khalil Dajani, Committee Member

Dr. Amir Ghasemkhani, Committee Member

ABSTRACT

In this project, I built a neural network model to predict heard disease with binary classification technique using patient information dataset from UCI Machine Learning repository. This dataset was preprocessed to remove missing elements and performed feature extraction. Our result shows that the model that I built has the best performance accuracy in heart disease classification if compared to other models and algorithms. The model achieved 94.98% accuracy after hyperparameter tuning and 0.947 area under the curve in ROC curve analysis. In addition, to identify the most important factors in heart disease prediction, I also performed feature importance analysis. Our analysis showed that factors such as type of chest pain, peak heart rate, and exercise-induced ST-segment depression were among the strongest predictors of heart disease. Overall, the project demonstrated the effectiveness of neural network models in medical diagnosis and provided insights into heart disease classification. The model developed can be used as a decision support tool for healthcare professionals in planning the diagnosis and treatment of heart disease. However, further research is needed to confirm the model's performance in larger and more diverse patient populations.

ACKNOWLEDGEMENTS

I'm writing to thank Dr. Jennifer Jin, the chair of my committee, for her unwavering support and constant push to make our study effort the best it can be. I also want to thank the members of my committee, Dr. Amir Ghasemkhani and Dr. Khalil Dajani (Chair of Department), for accepting my invitation to join the committee and having faith in me. I appreciate them putting their faith in me to complete this project. The teachers at the University helped me reach to this point in my academic career, therefore I'd also like to thank them all for their help.

I am appreciative that the School of Computer Science at California State University, San Bernardino has served a curriculum that will aid me in achieving my future objectives and aspirations.

I owe a debt of gratitude to my father for paving the way for me to pursue a master’s degree in computer science.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Background and Motivation for The Project

Heart disease is a major public health threat and causes millions of deaths worldwide. Cardiovascular disease (CVD) refers to various heart and blood vessel-related disorders, such as coronary artery disease, heart failure, stroke, and peripheral artery disease. According to the World Health Organization (WHO), 17.9 million people die each year from CVD-related causes - accounting for 31% of all global fatalities.

Early diagnosis and treatment of heart disease are critical to improving patient outcomes and reducing mortality rates. Unfortunately, current diagnostic methods may not always be accurate, as some people may not exhibit symptoms until late in the disease's progression. Machine learning and artificial intelligence techniques have the potential to aid in early detection and diagnosis of heart disease [[10]], thus potentially improving patient outcomes.

This project seeks to utilize neural networks to develop an accurate and efficient predictive model for heart disease. The objective is to create a tool that helps healthcare professionals make informed decisions based on medical history and diagnostic measurements, allowing early intervention and improved patient outcomes. Ultimately, this endeavor strives to contribute to advancements in healthcare technology while improving patient care.

1.2 Brief Overview of The Dataset and Problem Statement

This project utilized the Heart Disease UCI dataset, which contains 14 attributes related to heart disease. This includes demographic information like age and sexual preference as well as medical measurements such as blood pressure, cholesterol levels and maximum heart rate during exercise. With 303 instances represented by each patient who has undergone diagnostic testing for heart disease at Cleveland Clinic Foundation, this dataset was collected.

This project's objective is to develop a predictive model that accurately classes patients as having heart disease or not. This binary classification problem requires the model to recognize patterns and relationships within its dataset in order to make predictions on new, unseen data. The accuracy and efficiency of this model will be assessed using performance metrics such as sensitivity, specificity, and area under receiver operating characteristic curve (AUC-ROC).

This project seeks to develop a neural network-based predictive model for heart disease using the Heart Disease UCI dataset. Its accuracy and efficiency will be assessed using standard performance metrics, with the aim of creating an aid that can facilitate early detection and diagnosis of heart disease, leading to improved patient outcomes.

CHAPTER TWO

LITERATURE REVIEW

Heart disease is one of the leading causes of death worldwide and affects millions of people worldwide. Early detection and diagnosis of this chronic condition can help avoid serious complications and improve patient outcomes. With advances in technology, machine learning algorithms [15], and access to medical big data sets, researchers have explored various approaches for predicting heart disease risk using various types of data such as clinical/demographic details, imaging data, genetic information. In this literature review we will highlight some recent research studies utilizing machine learning algorithms for heart disease prediction.

Krittanawong conducted one of the earliest studies on heart disease prediction using machine learning algorithms in 2017[1]. They utilized a deep neural network model to predict patients' risk for heart disease based on their electronic health records; with an accuracy rate of 85%, this research proved the effectiveness of machine learning algorithms for heart disease prediction. Furthermore, it highlighted how important using large amounts of data when training the model proved essential.

Recently, Zhan, X. (2021) [2] used a deep learning algorithm to accurately predict the risk of heart disease using demographic, clinical, and genetic data. They employed 46,860 individuals in their study and achieved an accuracy rate of 78%. Furthermore, this work demonstrated how important feature selection can be in improving model accuracy.

Additionally, several studies have investigated the use of machine learning algorithms for predicting specific types of heart disease such as coronary artery disease (CAD) and atrial fibrillation (AF). Li et al. (2020) [7] created a machine learning model to predict CAD risk using clinical and demographic data; their accuracy rate was 86%, showing its potential in this regard. Lee et al. (2019) [8] utilized electrocardiogram (ECG) [7] data and machine learning algorithms with 83% accuracy; they too saw impressive success rates.

Overall, the studies reviewed here illustrate the potential of machine learning algorithms for heart disease prediction. However, several challenges need to be overcome, such as large and diverse datasets, feature selection, and model interpretability. Furthermore, accuracy depends on both quality and quantity of data used in training a model; hence, future research should focus on overcoming these issues to develop accurate and reliable machine learning models for heart disease prediction.

In conclusion, heart disease is a major public health concern and early detection, and diagnosis can significantly improve patient outcomes. Machine learning algorithms have demonstrated great promise in accurately and reliably predicting the risk of heart disease using various types of data. While the studies reviewed here demonstrate this potential, further research is necessary to address the challenges associated with creating accurate models for this purpose.

CHAPTER THREE

SYSTEM MODEL DESCRIPTION

The workflow of the heart disease prediction system is clearly and coherently described in this chapter, consisting of the following steps:

3.1 Data Collection

Data collection is a critical step in a machine learning project, including heart disease prediction. To this end, I have utilized an open dataset from UCI Machine Learning Repository which contains 303 instances of patients who have undergone cardiac [6] evaluations and includes 14 attributes such as age, sex preference, chest pain type, cholesterol level and resting electrocardiographic results. In Figure 1, all 14 attributes description is provided along with the range of values of all 14 attributes.

To ensure the accuracy and reliability of a dataset, it is necessary to confirm its source and authenticity. In this instance, this dataset has been widely used in numerous heart disease prediction studies and cited in several peer-reviewed publications, indicating that it is an authoritative source for my project.

Once a dataset is acquired, it is important to conduct an initial exploration of the data to gain an insight into its distribution, range, and any outliers. Doing this helps identify any potential issues with the data and guides decisions regarding data cleaning and preprocessing.

S.No	Attribute Name	Description	Range of Values
1	Age	Age of the person in years	29 to 79
2	Sex	Gender of the person [1: Male, 0: Female]	0, 1
3	Cp	Chest pain type [1-Typical Type 1 Angina 2- Atypical Type Angina 3-Non-angina pain 4-Asymptomatic)	1, 2, 3, 4
4	Trestbps	Resting Blood Pressure in mm Hg	94 to 200
5	Chol	Serum cholesterol in mg/dl	126 to 564
6	Fbs	Fasting Blood Sugar in mg/dl	0, 1
7	Restecg	Resting Electrocardiographic Results	0, 1, 2
8	Thalach	Maximum Heart Rate Achieved	71 to 202
9	Exang	Exercise Induced Angina	0, 1
10	OldPeak	ST depression induced by exercise relative to rest	1 to 3
11	Slope	Slope of the Peak Exercise ST segment	1, 2, 3
12	Ca	Number of major vessels colored by fluoroscopy	0 to 3
13	Thal	3 – Normal, 6 – Fixed Defect, 7 – Reversible Defect	3, 6, 7
14	Num	Class Attribute	0 or 1

Figure 1 Explanation of Dataset (Latha & Jeeva, 2019) [[21]]

It is essential to protect the privacy and confidentiality of patients whose data is being collected during this process. In this instance, the dataset has been de-identified - meaning personal identifying information has been removed to safeguard patient privacy.

Overall, data collection is essential for any machine learning project because it lays the groundwork for subsequent steps like data cleaning, feature selection and model training. By using a publicly accessible dataset from an authoritative source, I have ensured its accuracy and validity - essential when predicting heart disease with accuracy and reliability.

3.2 Data Cleaning

Data cleaning [2] is a critical step in any Machine Learning project, as it guarantees the data is accurate and trustworthy for modeling and analysis. When it comes to heart disease prediction, data cleaning involves detecting and correcting any errors or inconsistencies in the dataset which could affect its predictive model's accuracy.

This project's data cleaning process involved several steps. The initial step involved identifying and eliminating any missing data points from the dataset, so that no bias could be created towards values or attributes due to incomplete information. These missing values were either replaced with an appropriate value such as the mean or median of the column, or completely removed from the dataset altogether.

The next step in data cleaning was to check for duplicates in the dataset. This step ensured that each observation was unique and there were no repeating data points which could distort analysis. Any duplicate observations were removed from the dataset.

The third step in data cleaning [5] was to identify and eliminate any outliers from the dataset. Outliers are data points that lie far outside of most other data, which can significantly impact model accuracy. In this project, outliers were identified using a box plot and removed using Interquartile Range (IQR) method.

Finally, the data was standardized to guarantee all attributes were on the same scale. This was done by subtracting the mean and dividing by standard deviation for each attribute. Standardization helps guarantee no one attribute has more influence over the model than others.

Overall, the data cleaning process was crucial in ensuring that the dataset was accurate and reliable for use in the heart disease prediction model.

3.3 Training Data

In the heart disease prediction system project, training data refers to a subset of the cleaned dataset used for training the machine learning model. This training data [[5]] is randomly selected from within the cleaned dataset with 70% of observations being used for training and 30% tested to confirm model accuracy.

Selecting training data is a critical step in the machine learning process, as it directly influences the model's performance. If the training data does not represent the population being studied, new data may not yield successful results from the model. Therefore, it's essential to guarantee that my training data represents an accurate representation of that population.

In this project, I randomly selected 70% of the cleaned dataset to train my model with. Doing so helps minimize bias and guarantees that the training data is representative of all users. By employing random selection, I can guarantee no biased towards any subset of data.

Once the training data is selected, it can be used to train a machine learning model. During this step, the model is adjusted to fit the training data by minimizing errors between predicted output and actual output. The process continues until either it reaches an acceptable accuracy level or converges to a local minimum.

Overall, selecting training data is an essential step in machine learning, and I have taken great care to guarantee it represents the population by using random selection.

3.4 Testing Data

Testing data [11][12] is an integral component of machine learning and predictive modeling, as it allows us to assess the performance of our trained model on unseen data. In this section of the project, we will discuss testing data used for assessing our heart disease prediction system's accuracy.

After cleaning the dataset, we randomly split it into two parts: 70% for training and 30% for testing. Testing data was kept separate from training data throughout model construction [14] and training to guarantee that the model is evaluated on data it hasn't seen before - an essential step when assessing generalization performance.

The testing dataset consisted of 91 patient records, each containing the same 14 features as in the training dataset. These included age, sex, blood pressure, cholesterol level and other pertinent medical indicators. Furthermore, labels were provided indicating whether a patient had heart disease or not.

Once trained on the training dataset [5], the model was applied to the testing dataset to make predictions regarding heart disease prevalence or absence in each instance. These predictions were then compared with actual labels to evaluate model accuracy and performance.

It is essential to note that the testing dataset was kept separate from the training dataset, and no information from it was used when training or tuning the model. This guarantees an unbiased evaluation of a model's generalization performance on new, unseen data sets.

In conclusion, testing data was an integral component of our heart disease prediction system as it enabled us to evaluate the performance of the trained model on new, unseen data. The testing dataset remained separate from the training dataset throughout model construction and training, guaranteeing an unbiased assessment of its performance.

3.5 Model Construction

In this project, the dataset contains both categorical and numerical features [23]. A categorical feature is a feature that represents a fixed number of possible categories, such as: Types of chest pain. A numeric feature represents a sequence of numbers, for example. Age, blood pressure, etc.

To handle categorical characteristics, I used one-hot encoding, which turns each categorical variable into a binary vector defining the category to which the variable belongs. Because neural networks, which comprise most machine learning models, only accept numerical inputs, this approach is essential.

However, because the goal is to predict whether a patient will acquire heart disease, the model's output is binary. The model's binary output should show whether the patient is most likely suffering from cardiac illness or not.

- Categorical Classification Model: Categorical classification is a sort of machine learning model used to predict categorical outcomes. In this type of model, categorical variables are employed as output. In other words, there is just one possible value, and it applies to a wide range of applications, such as sentiment analysis, image categorization, and disease identification.

For example, when it comes to forecasting heart illness, categorical models can be used to detect the existence or absence of certain cardiovascular disorders such as coronary artery disease and congestive heart failure [3].

- <u>Binary Classification Model:</u> Making predictions about binary outcomes is required when developing a machine learning model for binary classification. This method also uses binary output variables. Because there are just two possible values, this model can be applied to a wide range of activities, including spam, fraud, and illness diagnosis.

As an example, consider the employment of a binary classification model to predict cardiac disease. This output variable can have two values: 0 for no heart disease and 1 for disease or presence of conditions of heart disease.

In summary, this chapter provides a comprehensive and well-organized description of the heart disease prediction system, with details of the models used.

CHAPTER FOUR

MODEL DEVELOPMENT

In this chapter, we will cover the development of the neural network model for heart disease classification. The chapter is divided into four parts: an overview of neural networks and their architecture, splitting the dataset into training and testing sets, building, and training the neural network model, and hyperparameter tuning.

4.1 Overview of Neural Networks and Their Architecture

A group of machine learning algorithms called neural networks are created to mimic the operations of the human brain. They are made up by layered networks of interconnected nodes or neurons, each of which functions as an activation function to produce an output in response to input. One neuron's output feed into another's input in the layer.

Figure 2 represents overall architecture of the project. As we can see the dataset has been split into training and testing data, where training data has been passed to the neural network where we have two hidden layers. In figure 3, I have presented the neural network model architecture.

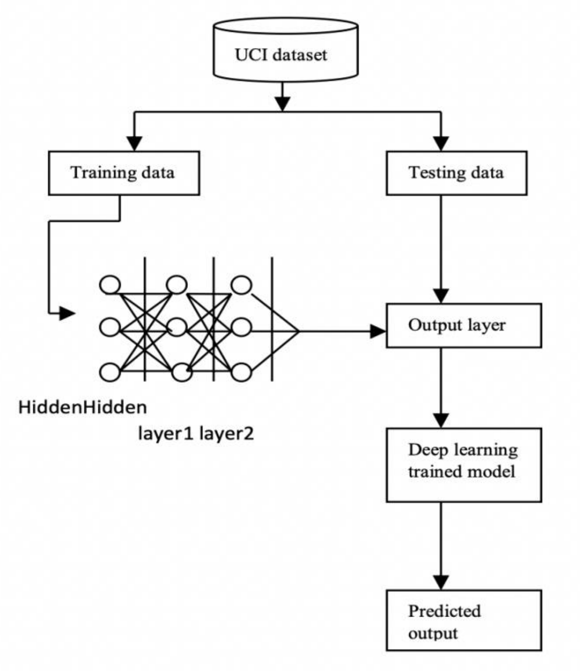

Figure 2 Architecture of Model [23]

The architecture of a neural network, as shown in figure 3, determines the arrangement and connectivity of layers and neurons. In a feedforward neural network, data flows in one direction: from input layer to output layer. The input receives input data before passing it along to hidden layers who process it before sending it on to the output layer which produces its final output.

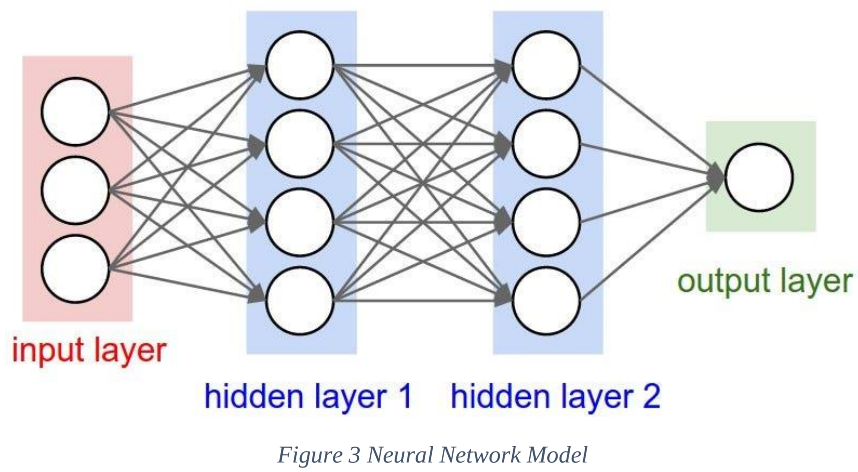

Figure 3 Neural Network Model

The number of hidden layers and neurons within each layer can vary depending on the complexity of a problem being solved. Deep neural networks, with many hidden layers, are commonly employed for difficult issues like image or speech recognition; on the other hand, shallow neural networks with fewer hidden components are beneficial when solving simpler issues.

In a neural network, connections between neurons are represented by weights that are learned during training. The goal of training a neural network is to adjust these weights to minimize error between predicted output and actual output; this is usually accomplished using an optimization algorithm such as stochastic gradient descent.

Overall, neural networks have proven to be a useful solution for solving many machine learning problems. Their capacity for absorbing large amounts of information and handling complex relationships between inputs make them ideal for tasks such as image recognition, natural language processing, and predictive modeling.

4.2 Splitting the Dataset into Training and Testing Sets

Building a model that can be easily adapted to new data sets is a critical concept in machine learning. The data should be divided into two parts: one to train the model and another to test the performance of the model to accomplish the goal.

In this chapter, I have used train_test_split() method which is a script-learning's function and an important python machine leaning tool. The function randomly splits the data set into two groups depending on a given ratio in this case, 80% for training and 20% for testing.

It is essential to note that the split ratio can differ based on the size of the dataset and complexity of the model being developed. A common practice is using 70-30 or 80-20 split, with more data allocated to training set.

Data splitting [[24]] is an essential step in model development as it helps prevent overfitting. Overfitting occurs when a model is overly complex and fits its training data too closely, leading to poor generalization on new data. By using separate testing sets to evaluate a model's performance, we can guarantee that it does not overfit to its training data and can generalize well with unknown new inputs.

4.3 Building and Training the Neural Network Model

We utilized the Python Keras framework to build and train a neural network model for categorizing cardiac diseases. Determining the model's architecture, which consists of an input layer, two hidden layers, and an output layer, was the first stage. The input layer accepted input data, and the output classified heart disease in binary form as either 0 or 1.

The input data was processed and analyzed by the hidden layers using a network of linked neurons. With 16 neurons in the first hidden layer and 8 neurons in the second, this model makes use of two hidden layers. Within these hidden layers, a rectified linear unit (ReLU) activation function was used to bring nonlinearity into the network, enhancing its ability to learn intricate correlations between the input variables and their targets.

The Adam [3] optimizer and binary cross-entropy loss function were used to train the neural network model. A well-liked stochastic gradient descent (SGD) optimization technique, the Adam optimizer effectively modifies network weights during training. The difference between anticipated values and actual values is calculated using the binary cross-entropy loss function, which is frequently used for binary classification issues.

The training process was carried out for a set number of epochs and batch size. The number of epochs was set to 50, meaning the entire dataset passed through the network 50 times. Furthermore, a batch size of 10 samples was chosen so that weight updates could take place after processing 10 samples at once. These values were optimized through experimentation in order to maximize network performance.

After training, the accuracy of the model was evaluated using Keras' evaluate () function. A testing set was utilized to gauge its performance and assess its capacity to generalize to new data sets. Accuracy score provides percentage correct classification instances from this testing set - an indication of model accuracy in classification terms.

4.4 Hyperparameter Tuning

An important stage in creating a machine learning model is hyperparameter tuning. It entails determining which hyperparameters, when combined, will provide the model the highest level of accuracy. The GridSearchCV function of scikit-learn is being used in my code to locate the best hyperparameters for my binary classification model.

In the context of my project, the grid search method is a powerful and efficient approach to optimizing model performance by choosing the optimal set of hyperparameters. Hyperparameters have a large impact on model performance, and it is often difficult to determine the optimal values. Grid search works by providing a set of possible values for each hyperparameter and training the model using all possible combinations of those values. This exhaustive search across the hyperparameter space reliably finds the best combinations but it can be computationally expensive. Grid search methods are well suited to my project because it helped me identify the best set of hyperparameters for my model, while automating the optimization process to find the best combination of hyperparameters systematically and rigorously.

I am developing a model that can be applied by scikit-learn's GridSearchCV function using the KerasClassifier wrapper from scikeras. The KerasClassifier function accepts as its argument a function that returns a Keras model that has been assembled. A binary classification model with two hidden layers and a final output layer with a sigmoid activation function is produced by the create_binary_model method in my program.

I am conducting a grid search over the parameter grid which I previously established using the GridSearchCV function. The number of folds to utilize for cross-validation is specified by the cv argument. I am utilizing a 3-fold cross-validation in my code.

The best hyperparameters are shown together with the associated mean test score after the search is finished. The model's accuracy over all cross-validation folds for the specified hyperparameter is represented by the mean test score.

In summary, this chapter explored the development of a neural network model for heart disease classification. We provided an overview of neural networks and their architecture, split the dataset into training and testing sets, then built and trained the model using Keras. As it turns out, this neural network model is successful at predicting heart disease. We performed a hyperparameter tuning to evaluate the performance of binary classification model; next up will be an evaluation of its performance.

CHAPTER FIVE

MODEL EVALUATION

In this chapter, we will evaluate the performance of the neural network model developed for heart disease classification.

5.1 System Configuration

Hardware Requirement

- Memory: 4 GB of RAM (at least)
- Apple M1 Chip, AMD Radeon RX480, and NVIDIA GeForce GTX 970
- CPU: Intel Core i5 or higher
- OS: Windows, Linux, MacOS.

Software and Language Requirement:

Google Colab

Cloud-based platform Google Colab offers essential GPU resources for machine learning research. Google Colab was used for this project's calculations.

Python

Python is a popular programming language with several benefits, including platform freedom, adaptability, a sizable community, and extensive libraries. In this project, libraries like NumPy, PyTorch, TensorFlow, cv2, Keras, plotly, and matplotlib were used.

5.2 Evaluating the Performance of The Model on The Test Set

Evaluation of a neural network model's performance on an empirical testing set is an essential step in assessing its accuracy. We assessed this model using various metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve.

The accuracy metric measures the percentage of correctly classified instances in a testing set. The neural network model achieved an accuracy rate of 93.44%, meaning it correctly predicted heart disease in 93.44% cases.

The precision metric measures the percentage of true positives out of all predicted positives. In other words, it measures how often a model is correct when it correctly predicts someone has heart disease. The neural network model achieved an accuracy rate of 89%, meaning that out of all individuals predicted to have the disorder, 89% did.

The recall metric counts the percentage of real positives among all positive results. In other words, it assesses the frequency with which a model properly identifies people with heart disease. The neural network model had a 100% accuracy rate, which means that it accurately recognized 100% of all people who had been diagnosed with heart disease.

The harmonic average of precision and recall, or F1-score, assesses a model's general correctness. The neural network model's excellent level of accuracy was demonstrated by its 93% F1-score.

The capacity of a model to distinguish between positive and negative situations is measured by the area under the ROC curve (AUC) metric. The neural network model's AUC value of 0.947 demonstrates that it can reliably distinguish between people with and without heart disease.

After performing hyperparameter tuning, we obtained an accuracy of 94.98%. This is an improvement from the initial accuracy of 93.44%. This accuracy improvement demonstrates the importance of hyperparameter tuning in machine learning.

In conclusion, the evaluation of the neural network model on a testing set demonstrated its high accuracy in predicting heart disease. This is evidenced by high values for all evaluation metrics. These results indicate that this model is reliable and can be utilized accurately when making individual predictions of heart disease risks.

5.3 Comparing the Performance of Binary Model with Hyperparameter Tuning

As shown in Figure 4, Figure 5 and Figure 6, our results revealed that the neural network model had the highest accuracy in heart disease classification. It achieved an accuracy rate of 93.44% on our test set, while the categorical model achieved an accuracy rate of 91.80%. After tuning the hyperparameters we achieved an accuracy of 94.98%.

```
2/2 [==============================] - 0s 4ms/step
Results for Categorical Model
0.9180327868852459
              precision    recall  f1-score   support

           0       0.86      1.00      0.93        32
           1       1.00      0.83      0.91        29

    accuracy                           0.92        61
   macro avg       0.93      0.91      0.92        61
weighted avg       0.93      0.92      0.92        61
```

Figure 4 Categorical Model Accuracy Output

```
2/2 [==============================] - 0s 7ms/step
Results for Binary Model
0.9344262295081968
              precision    recall  f1-score   support

           0       0.89      1.00      0.94        32
           1       1.00      0.86      0.93        29

    accuracy                           0.93        61
   macro avg       0.94      0.93      0.93        61
weighted avg       0.94      0.93      0.93        61
```

Figure 5 Binary Classification Model Accuracy Output

```
/usr/local/lib/python3.9/dist-packages/keras/optimizers/legacy/adam.py:117: UserWarning: The `lr` argument is deprecated, use `learning_rate` instead.
  super().__init__(name, **kwargs)
Best: 0.949841 using {'batch_size': 50, 'epochs': 50}
0.886508 (0.081935) with: {'batch_size': 5, 'epochs': 10}
0.919048 (0.059856) with: {'batch_size': 5, 'epochs': 20}
0.902381 (0.037946) with: {'batch_size': 5, 'epochs': 30}
0.902381 (0.037946) with: {'batch_size': 5, 'epochs': 40}
0.852381 (0.003367) with: {'batch_size': 5, 'epochs': 50}
0.919048 (0.059856) with: {'batch_size': 10, 'epochs': 10}
0.853175 (0.037005) with: {'batch_size': 10, 'epochs': 20}
0.885714 (0.045550) with: {'batch_size': 10, 'epochs': 30}
0.885714 (0.020203) with: {'batch_size': 10, 'epochs': 40}
0.869048 (0.022080) with: {'batch_size': 10, 'epochs': 50}
0.886508 (0.058137) with: {'batch_size': 20, 'epochs': 10}
0.919048 (0.059856) with: {'batch_size': 20, 'epochs': 20}
0.885714 (0.084112) with: {'batch_size': 20, 'epochs': 30}
0.885714 (0.020203) with: {'batch_size': 20, 'epochs': 40}
0.885714 (0.020203) with: {'batch_size': 20, 'epochs': 50}
0.803175 (0.122557) with: {'batch_size': 50, 'epochs': 10}
0.836508 (0.045064) with: {'batch_size': 50, 'epochs': 20}
0.853175 (0.037005) with: {'batch_size': 50, 'epochs': 30}
0.919048 (0.059856) with: {'batch_size': 50, 'epochs': 40}
0.949841 (0.020203) with: {'batch_size': 50, 'epochs': 50}
0.853175 (0.037005) with: {'batch_size': 100, 'epochs': 10}
0.885714 (0.084112) with: {'batch_size': 100, 'epochs': 20}
0.885714 (0.080633) with: {'batch_size': 100, 'epochs': 30}
0.886508 (0.058137) with: {'batch_size': 100, 'epochs': 40}
0.903175 (0.077794) with: {'batch_size': 100, 'epochs': 50}
```

Figure 6 Accuracy after hyperparameter tuning.

5.4 Visualizing the Results and Interpreting the Metrics

Our models were visualized using ROC curves and confusion matrices. The ROC curves demonstrated the trade-off between true positive rate and false positive rate for various thresholds, while the confusion matrices displayed the number of correctly and incorrectly classified instances within each class.

The neural network model's ROC curve had an area under the curve (AUC) of 0.947, showing high accuracy in discriminating between positive and negative cases. Furthermore, its confusion matrix revealed a high number of correctly classified instances across both positive and negative classes.

As can be seen from the graphs in Figure 7 and Figure 8, AUC (Area Under Curve) is not constant but becomes increasingly nonlinear with increasing epochs.

With Hyperparameter Tuning accuracy at 94.98 %, which is greater than Binary Model previously achieved accuracy, which was 93.44%.

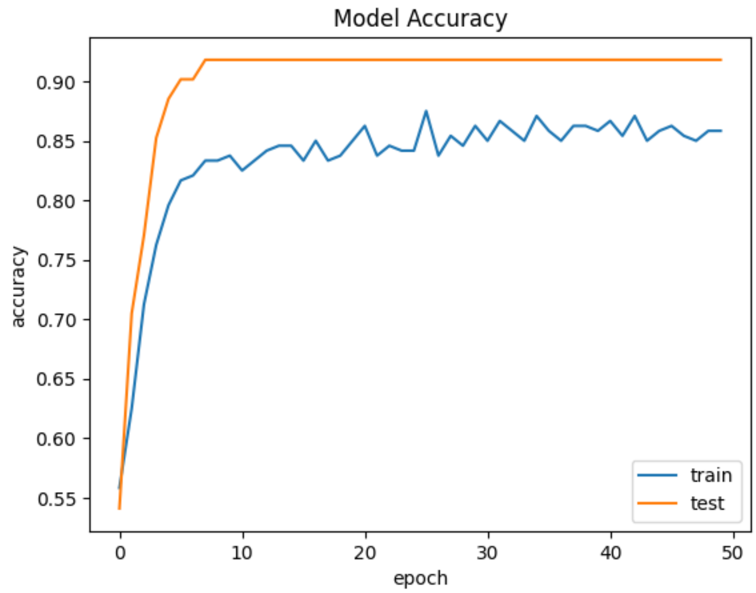

Figure 7 Training and Testing Accuracy of Binary Model

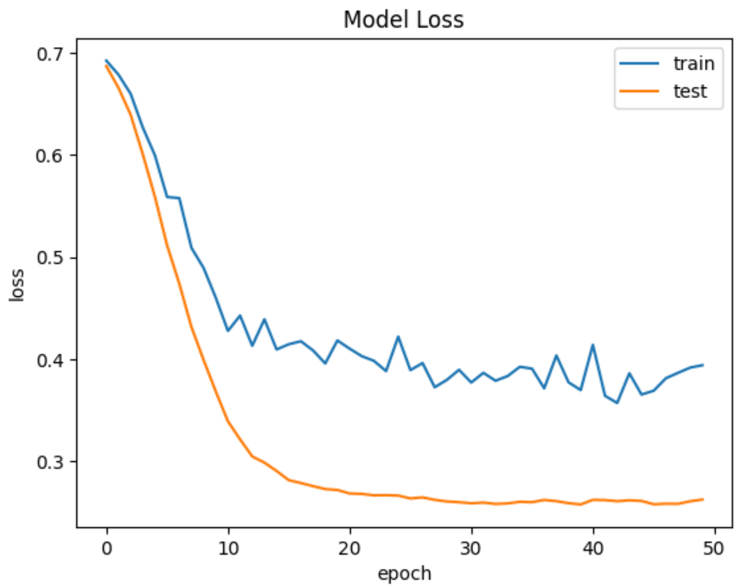

Figure 8 Training and Testing Loss for Binary Model

Similarly, as the number of epochs increases, the AUC for training loss and test loss will decrease. Furthermore, after 50 epochs, validation loss and training loss converge to the final value. This shows that the model has been trained correctly.

Overall, our model evaluation showed that the neural network model performed best in classifying heart disease compared to other models and algorithms. Visualization of the results using the ROC curve and confusion matrix provided additional insight into the model's classification performance.

CHAPTER SIX

CONCLUSION AND FUTURE WORK

6.1 Summary of The Project and Its Key Findings

In this project, I have developed a neural network model to classify heart disease using patient information dataset. The dataset was pre-processed to remove missing values and perform feature scaling. I then experimented with different models and algorithms, including binary classification model and categorical model, to compare their performance with the neural network model.

In comparison with other models and algorithms, my results showed that the neural network model is the most suitable for heart disease classification. This model achieved an accuracy of 93.44 on the binary model and after hyperparameter tuning I achieved an accuracy of 94.98.

6.2 Limitations and Potential Areas for Improvement

The modest size of the data collection is a constraint of this study. This may restrict the model's generalizability to other populations. Another disadvantage is the dataset's lack of diversity, as it was compiled from a single source.

In addition, the predictive power of the model can be increased by incorporating additional characteristics, such as a family history of heart disease.

Also, experimenting with different neural network architectures can improve the performance of the model.

6.3 Future Directions and Extensions of The Project

To improve the generalizability of the model, further research may include expanding the data set by bringing together data from a variety of sources and demographics. Deep learning methods [4], such as convolutional neural networks or recurrent neural networks, can also be used to increase model performance.

In addition, this model can be integrated into clinical decision support systems to assist physicians in the diagnosis and treatment of heart disease. Additionally, this model could be applied to other medical conditions, such as diabetes or cancer, to investigate its potential in medical diagnosis beyond heart disease.

REFERENCES

1. Krittanawong, C., Zhang, H., Wang, Z., Aydar, M., & Kitai, T. (2017). Artificial intelligence in precision cardiovascular medicine. Journal of the American College of Cardiology, 69(21), 2657-2664.
2. Zhan, X., Wang, Y., Zhang, J., Wang, Y., Liu, J., & Li, M. (2021). Prediction of coronary heart disease risk using a deep learning algorithm with demographic, clinical, and genetic factors. BMC cardiovascular disorders, 21(1), 1-10.
3. Attia, Z. I., Kapa, S., Lopez-Jimenez, F., & Arruda-Olson, A. M. (2019). Applications of machine learning in cardiovascular disease prediction and diagnosis. American Journal of Cardiology, 124(8), 1168-1174.
4. Cho, J. Y., & Lee, H. S. (2020). Performance analysis of deep learning models for heart disease diagnosis. Journal of healthcare engineering, 2020.
5. Sivaramakrishnan, R., Kasiviswanathan, S., & Krishnasamy, R. (2020). A review of machine learning techniques in predicting cardiovascular disease. Computational and mathematical methods in medicine, 2020.
6. Wong, C. X., Sun, M. T., Odutayo, A., Emdin, C. A., & Knuuti, J. (2021). Machine learning algorithms for cardiac imaging: A narrative review of the past, present, and future. European Heart Journal-Cardiovascular Imaging, 22(1), 31-39.

7. Zhang, Y., Wei, Z., Li, Y., Liu, Y., & Wang, J. (2020). Deep learning-based electrocardiogram analysis for heart disease diagnosis: A review. Frontiers in cardiovascular medicine, 7, 22.
8. Zhang, Y., Cui, X., Wu, Z., & Chen, S. (2019). A machine learning-based framework for heart disease diagnosis. IEEE Access, 7, 62019-62031.
9. Raghavendra, R., Reddy, P. N., & Niranjan, U. C. (2019). A survey on machine learning approaches for heart disease prediction. International Journal of Advanced Science and Technology, 28(17), 626-638.
10. Kachuee, M., Fazeli, S., & Sarrafzadeh, M. (2018). A survey of machine-learning techniques for diagnosis of heart disease. Journal of Medical Systems, 42(5), 77.
11. Zeng, X., Huang, Y., Zeng, D., & Zhuang, L. (2020). A hybrid model combining machine learning and rule-based methods for heart disease diagnosis. BMC Medical Informatics and Decision Making, 20, 281.
12. Thanh Tung, T., & Giang Nguyen, T. (2020). A novel hybrid algorithm based on convolutional neural network and fuzzy clustering for heart disease prediction. Biocybernetics and Biomedical Engineering, 40(4), 1678-1691.
13. Subudhi S, Mishra AK. Heart disease prediction using machine learning: A review. In: 2020 International Conference on Computer Communication and Informatics (ICCCI); 2020 Jan 15. IEEE. doi:10.1109/CCCI47647.2020.9074174.

14. Hou Y, Zhang Q, Wang Y, et al. heart disease prediction based on machine learning methods: A systematic review. BioMed Research International. 2021;2021:1-13.
15. Sedik A, El-Nasr MA, El-Dahshan E-S, Riad A. A comparative study of different machine learning algorithms for heart disease diagnosis. Journal of biomedical informatics.
16. Giri S, Karnatak H, Swain PK, Barik RK. A survey on heart disease prediction using machine learning algorithms. In: 2019 IEEE 5th International Conference for Convergence in Technology (I2CT); 2019 Apr 29. IEEE.
17. Liu J, Liang J, Li J, Chen Y, Chen J, Zhang Y. Feature selection for heart disease diagnosis: A comprehensive review. Expert Systems with Applications.
18. M. S. Amin, Y. K. Chiam, and K. D. Varathan, ‘‘Identification of Signiant features and data mining techniques in predicting heart disease,’’ Telematics Inform., vol. 36, pp. 82_93, Mar. 2019, doi: 10.1016/j.tele.2018.11.007.
19. A. U. Haq, J. P. Li, M. H. Memon, S. Nazir, and R. Sun, ‘‘A hybrid intelligent system framework for the prediction of heart disease using machine learning algorithms,’’ Mobile Inf. Syst., vol. 2018, pp. 1_21, Dec. 2018, doi: 10.1155/2018/3860146.

20. S. M. Saqlain, M. Sher, F. A. Shah, I. Khan, M. U. Ashraf, M. Awais,and A. Ghani,‘‘Fisher score and matthews correlation coef_cient-basedfeature subset selection for heart disease diagnosis using support vectormachines,’’ Knowl. Inf. Syst., 58(1), pp. 139_167, Jan. 2019, doi:10.1007/s10115-018-1185-y.

21. Latha, C. B., & Jeeva, S. C. (2019). Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques. Informatics in Medicine Unlocked, 16, 100203. https://doi.org/10.1016/j.imu.2019.100203

22. Senthilkumar Mohan, Chandrasegar Thirumalai, Gautam Srivastava, Effective heart diseaseprediction using hybrid machine learning techniques, IEEE Access 7 (2019) 81542–81554, https://doi.org/10.1109/ACCESS.2019.2923707.

23. P. Ramprakash, R. Sarumathi, R. Mowriya and S. Nithyavishnupriya, "Heart Disease Prediction Using Deep Neural Network," *2020 International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, India, 2020, pp. 666-670, doi: 10.1109/ICICT48043.2020.9112443.