TED ROGERS SCHOOL OF MANAGEMENT

Heart Disease Classification using Neural Network and PyTorch

Ayushi Pitchika

Project Objective

- Machine learning and neural network models can analyze vast medical data, offering comparable accuracy to traditional methods while expediting diagnostic processes. By leveraging extensive medical datasets, algorithms can automate intricate data analysis while continually refining their predictive accuracy.

- This study investigates the development of such a deep neural network model, employing a classification approach within a machine learning framework to identify heart disease presentation. The classification goal of the study is to predict whether the patient presents heart disease or not, leveraging PyTorch capabilities to set up a Deep Neural Network.

Data Exploration

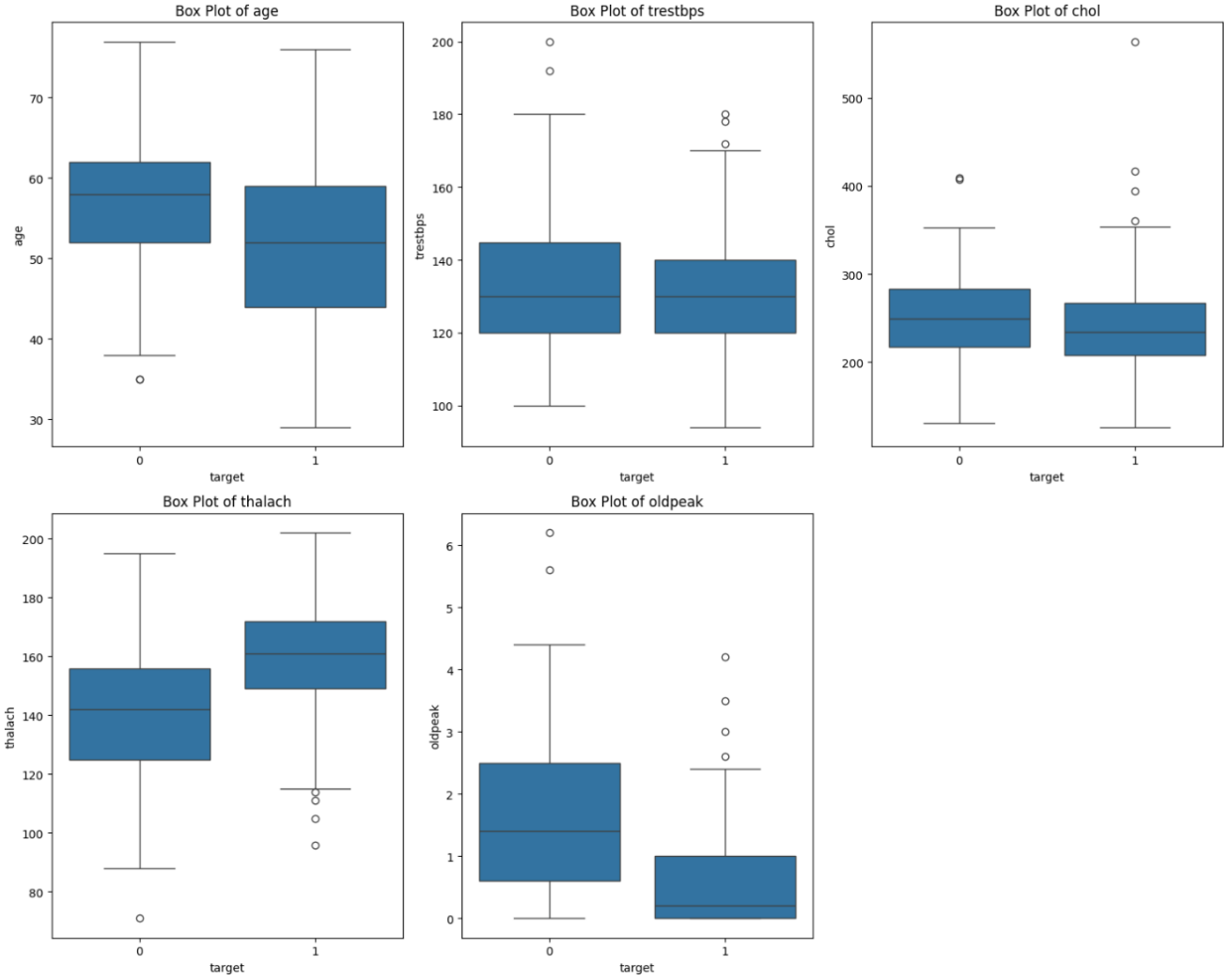
- The dataset was imported as a pandas dataframe. It consists of 14 columns, including the target variable, and 303 data points. The target variable has a binary classification with 1=yes and 0=no. There is a mix of categorical and numerical features in the dataset.

```
data.describe()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000

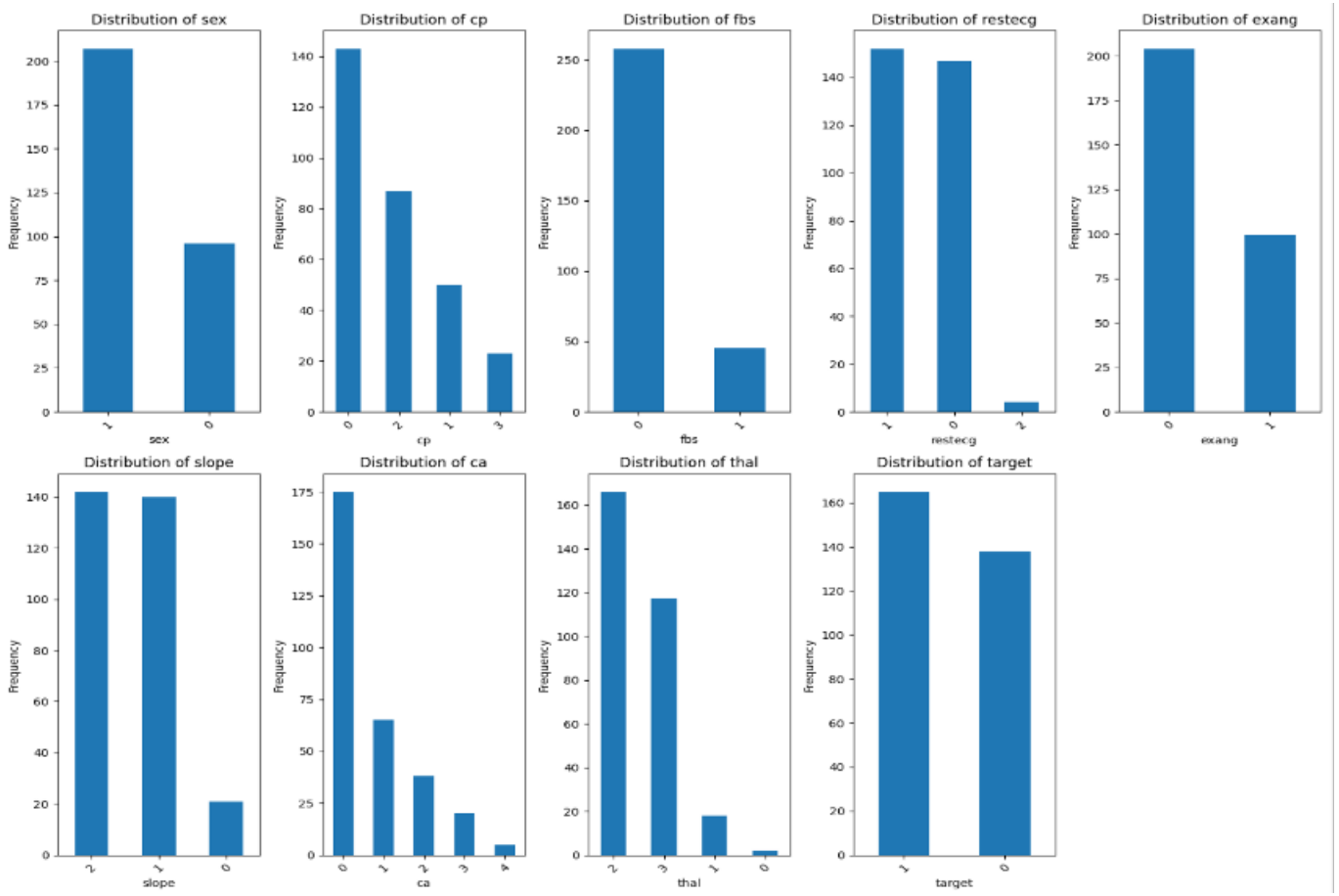
Data Exploration

- Box plots were generated to understand the central tendency, spread, and distributional characteristics of continuous variables, as well as identify potential outliers and asymmetries in the data.
  - 'oldpeak' has a skewed distribution
  - differences in the spread of the positive and negative target values for age
  - positive group has a higher mean ‘thalach’ value than the negative
  - 'trestbps', 'chol' and 'oldpeak' have a few outliers



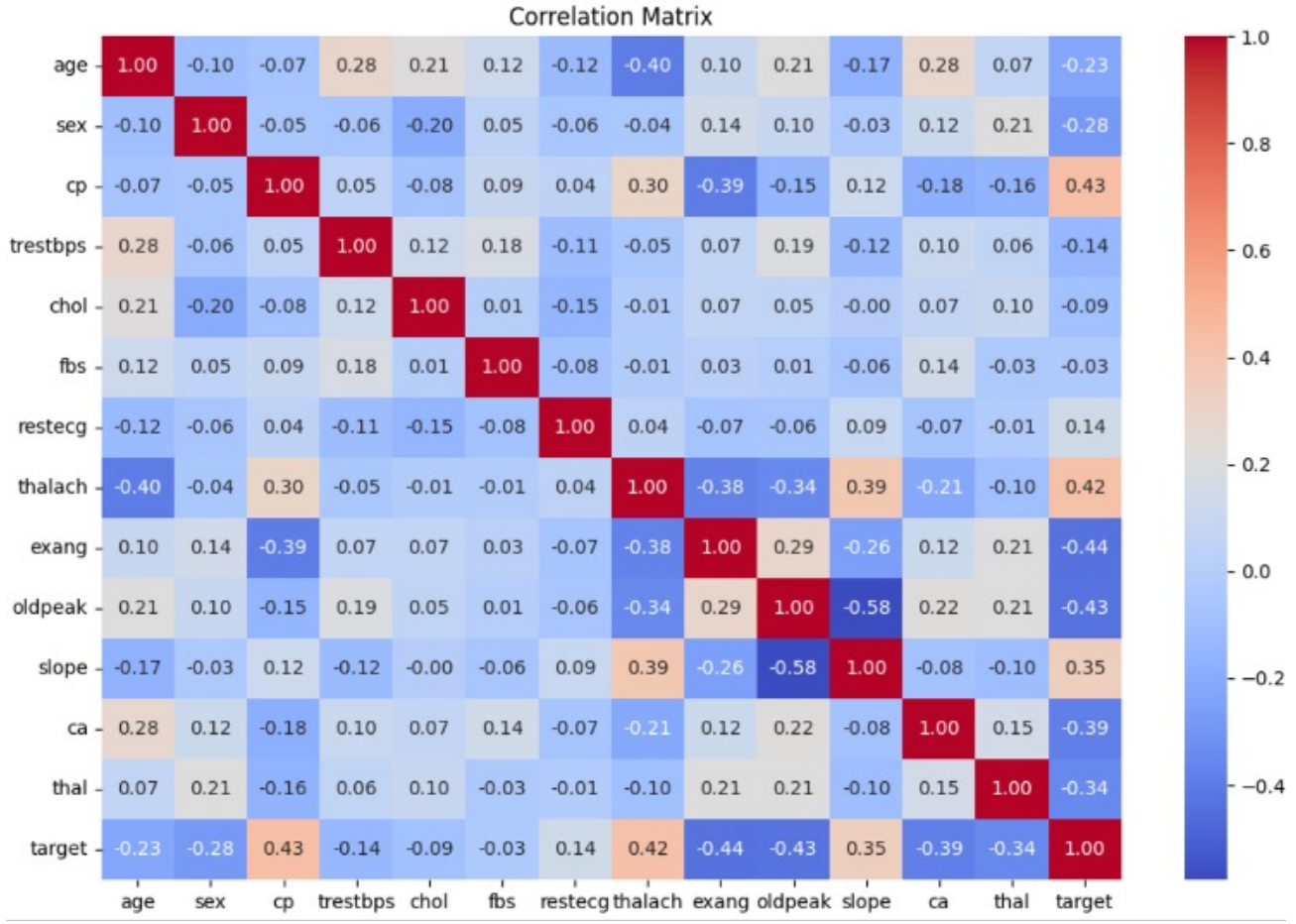
Data Exploration

- Bar plots are plotted for categorical features to investigate the distribution of the various categories.
  - dataset provided is well balanced as seen from the bar plot for the target variable
  - double the number of males than females in the dataset
  - Similar distribution is seen in the ‘exang’ feature



Data Exploration

- Correlation matrix to explore correlations between features and the target variable.
  - 'cp' and 'thalach' have a high positive correlation with the target and 'exang' and 'oldpeak' have high negative correlations
  - 'fbs' seems to be minimally correlated to the target
  - since no two features have a correlation of -/+ 1, it can be deduced that none of the features are redundant



Data Preprocessing and Cleaning Steps Undertaken

- There was no need to handle missing values and the categorical variables were already encoded into numerical format.
- Feature scaling was performed since neural networks are sensitive to the scale of input features. Based on the distribution of data and sensitivity to outliers, Z-score normalization (standardization) method was used.
- The dataset was split into training and testing sets with a 70-30 split as specified. The split was also stratified to ensure that it maintained the distribution of the target classes to avoid data leakage.

Architecture of Neural Network

The neural network architecture for HeartDiseaseModel() is a feedforward neural network with configurable parameters for size of the input, the number of hidden layers, the size of each hidden layer, and the activation function used.

The output layer consists of a single node, which is used for binary classification (predicting the presence or absence of heart disease). It uses a sigmoid activation function to output probabilities between 0 and 1, representing the likelihood of the positive class.

Hyperparameter tuning	**Value 1**	**Value 2**	**Value 3**
Number of nodes in each layer	32	64	128
Number of layers	1	2	3
Activation Function	Rectified Linear Unit (ReLU)	Sigmoid	Leaky Rectified Linear Unit (LeakyReLU)

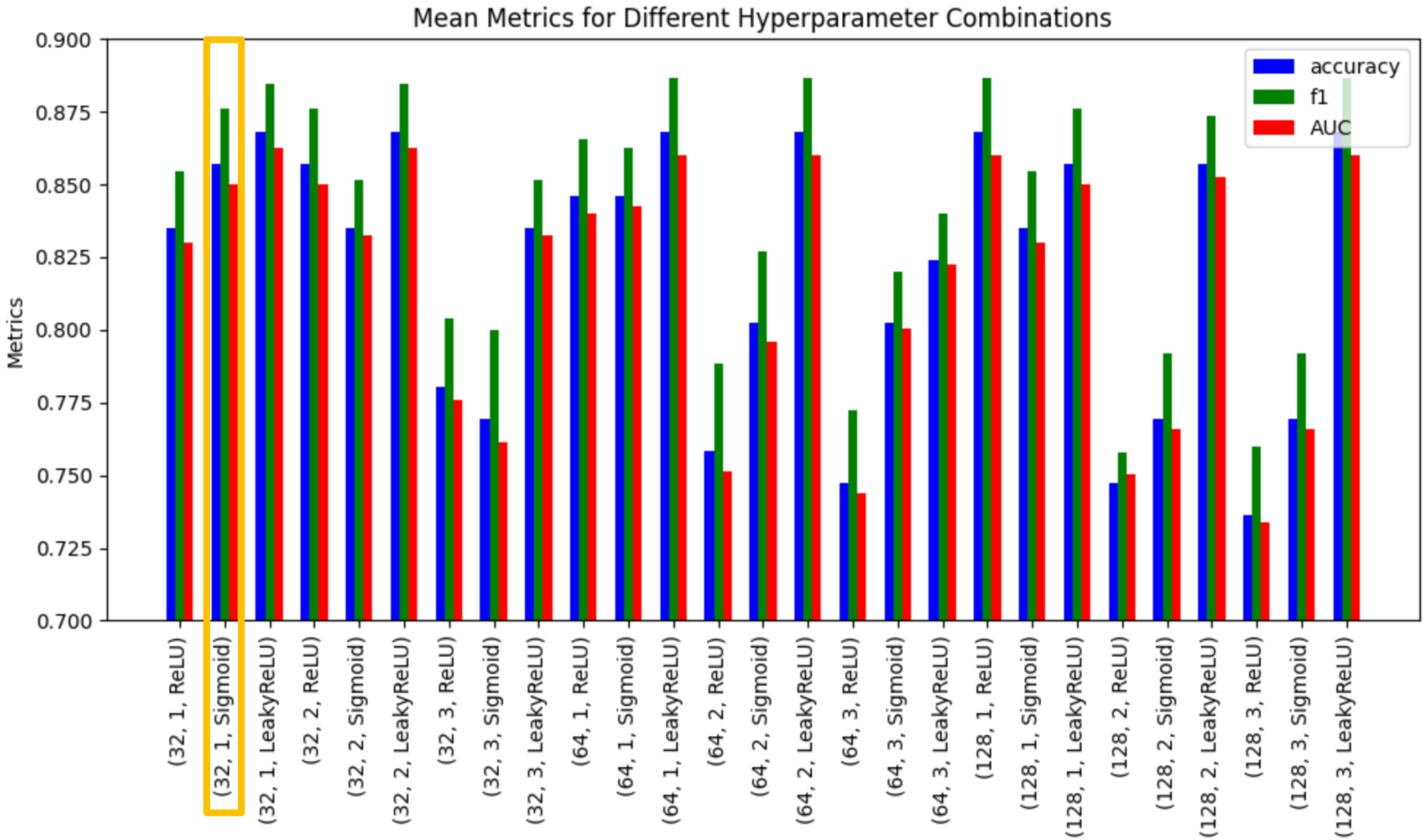
Evaluation of Model

Based on highest accuracy, f1, recall and high AUC scores, the best neural network model is determined to have the following architecture:

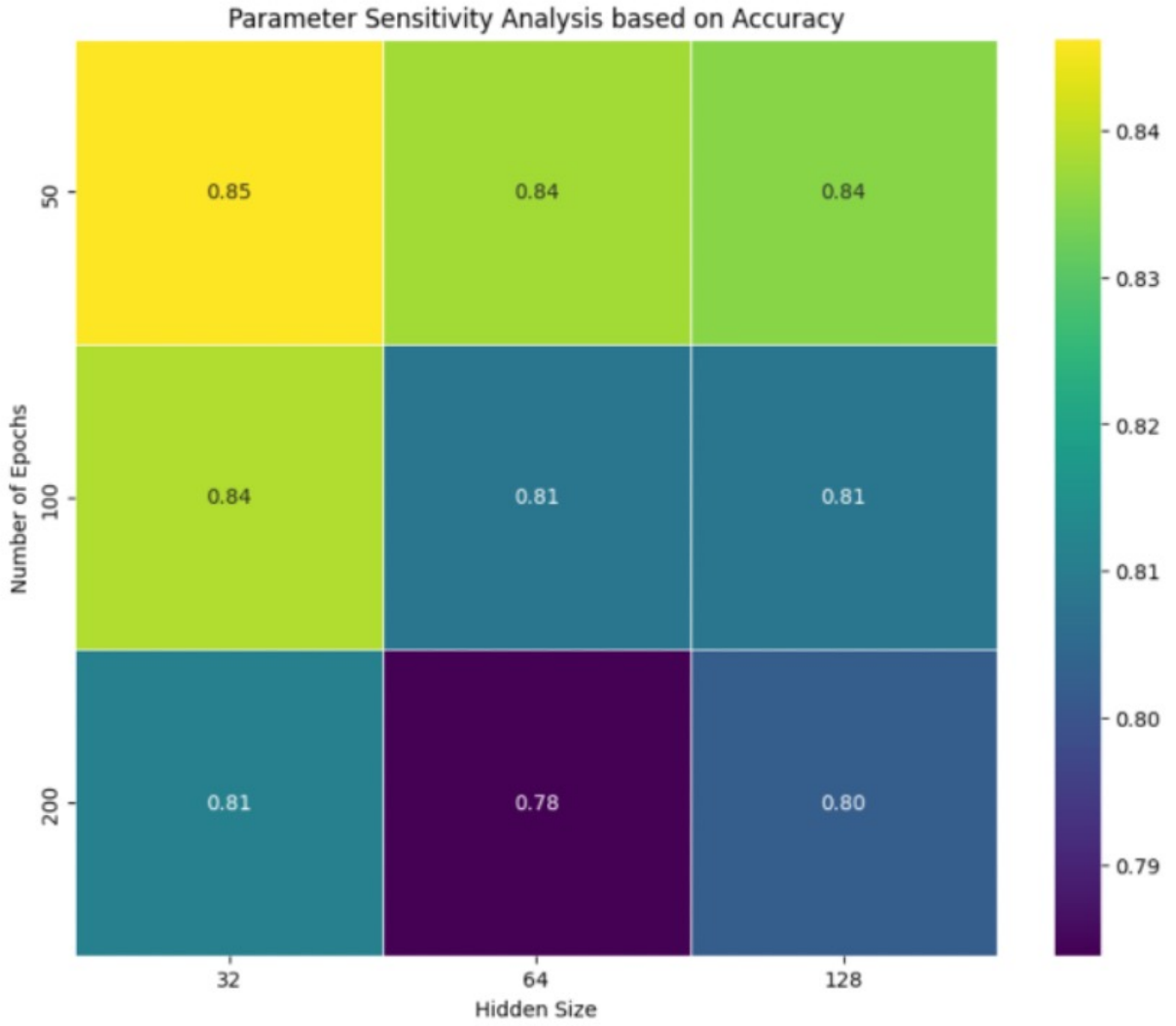
32 nodes per layer, 1 layer and Sigmoid activation function

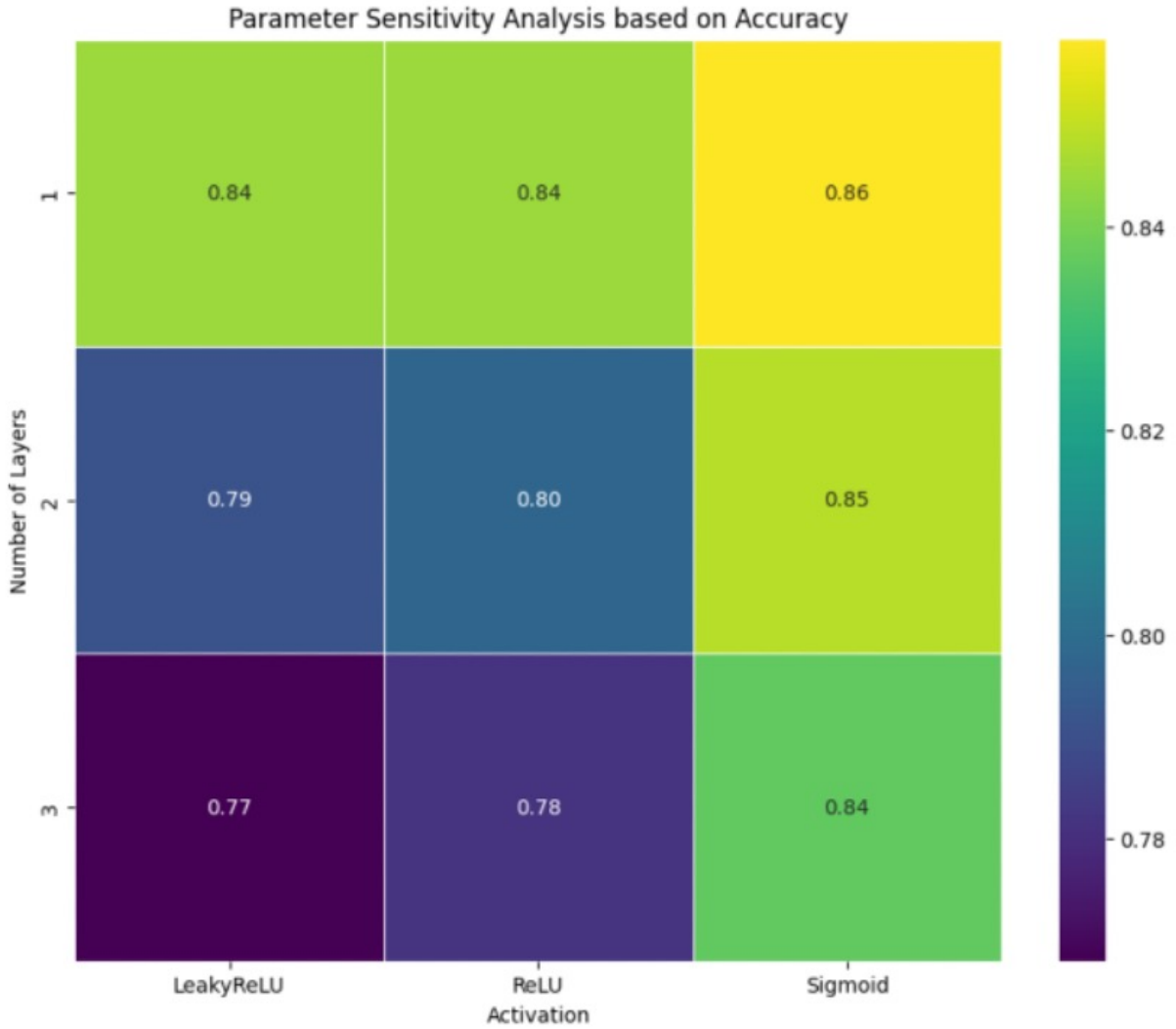
The performance metrics for this model are as follows:

- Accuracy: 86.81%
- f1: 88.46%
- AUC score: 86.24%



Parameter Sensitivity Analysis





Parameter Sensitivity Analysis

• Change in number of nodes: Based on the heatmap, it is observed that the lower the number of nodes, the higher the accuracy. The accuracy changes by 1-3% as the number of nodes changes from 32 to 128.

• Change in number of epochs: Based on the heatmap, it is observed that the lower the number of nodes, the higher the accuracy. The accuracy changes by 2-4% as the number of epochs changes from 50 to 200.

• Change in number of layers: Based on the heatmap, it is observed that the lower the number of hidden layers, the higher the accuracy. The accuracy changes by 3-7% as the number of layers changes from 1 to 3.

• Activation function: Based on the heatmap, it is observed that the sigmoid activation function has the highest accuracy, followed by the LeakyReLU and then ReLU.

Discussions

- High values in performance metrics indicates the model is effective in distinguishing between healthy and diseased hearts.
- It is crucial to assess how well the model generalizes to unseen data through techniques like k-fold cross-validation.
- Learning rate scheduling and dropout regularization can also help optimize training may also help boost performance.
- Due to the smaller size of the dataset, a simpler architecture is better able to predict the classification goal.

Conclusion and Future Work

- Created a feedforward NN archit4ecture with hyperparameters for the number of hidden layers, the size of each hidden layer, and the activation function
- Trained model on training set and then tested using preciously separated test datapoints.
- Metrics such as accuracy, f1 and AUC score to evaluate the model performance and recommend best model.
- Parameter sensitivity analysis to determine how changes affect the model performance.
- Tables and visualizations created to report the findings to stakeholders in a clear and concise manner.
- Recommendation for future work would be to validate the findings using additional datasets or cross-validation techniques to ensure the robustness of the conclusions