

Hypertension Detection

Likhitha Devi Nekkanti - CS5300

May 1, 2023

Contents

1	Introduction	3
2	Hypertension Dataset	3
2.1	Data distribution graphs of each attribute before normalization	4
2.2	Data distribution graph of target feature before normalization	5
3	Data Cleaning	5
3.1	Splitting of data into training and validation	5
3.2	Data Normalization	5
3.3	Data distribution graphs of each attribute after normalization	6
4	Building a Model	7
4.1	Logistic Regression Model	7
4.1.1	Candidate Models	7
4.2	Learning Curve of Selected Neural Network Architecture	8
5	Model Evaluation	9
5.1	Test Accuracy	9
5.2	Custom Function and Keras Function	11
6	Feature Significance and Reduction	11
6.1	Feature Significance	11
6.2	Feature Reduction	12
7	Challenges Faced	12
8	Conclusion	13

1 Introduction

Millions of individuals around the world suffer from hypertension, also known as high blood pressure. Serious health issues like heart disease, stroke, and kidney failure can result from it. For the purpose of avoiding bad consequences and enhancing patient outcomes, early hypertension detection and treatment are essential.

Using various kinds of patient data, such as age, sex, blood pressure, cholesterol levels, and other factors, this project can predict the presence of hypertension. Healthcare professionals can quickly react and treat patients with hypertension in order to avoid problems and improve their health outcomes by correctly recognizing these patients.

In this project, various patient features and their corresponding hypertension status are present in a hypertension dataset that will be used. Our objective is to use this information to create a model that can accurately predict the presence of hypertension in new patients.

2 Hypertension Dataset

The dataset “HYPERTENSION” is taken from the Kaggle website. The dataset consists of 26083 rows and 14 columns or features. The first 13 features are taken as inputs and the target feature is taken as output. The names and the units in which each feature is measured is listed below:

- 1.Age: age of the patient (years).
- 2.Sex: gender of the patient (0 = female, 1 = male).
- 3.CP: chest pain type (0 = typical angina, 1 = atypical angina, 2 = non-anginal pain, 3 = asymptomatic).
- 4.Trestbps: resting blood pressure (mm Hg).
- 5.Chol: serum cholesterol (mg/dL).
- 6.Fbs: fasting blood sugar (0 = normal, 1 = high).
- 7.Restecg: resting electrocardiogram results (0 = normal, 1 = ST-T wave abnormality, 2 = left ventricular hypertrophy)
- 8.Thalach: maximum heart rate achieved (bpm).
- 9.Exang: exercise-induced angina (0 = no, 1 = yes).
- 10.Oldpeak: ST depression induced by exercise relative to rest (unitless)
- 11.Slope: slope of the peak exercise ST segment (0 = upsloping, 1 = flat, 2 = downsloping).
- 12.Ca: number of major vessels (0-3) colored by fluoroscopy
- 13.Thal: type of thalassemia (0 = normal, 1 = fixed defect, 2 = reversible defect).
- 14.Target: presence of heart disease (0 = no, 1 = yes).

2.1 Data distribution graphs of each attribute before normalization

The data distribution graphs of each attribute along with the "target" attribute plotted in histograms before performing normalization. In every graph, the maximum and minimum values along with their distribution according to their values are plotted in Figure-1.

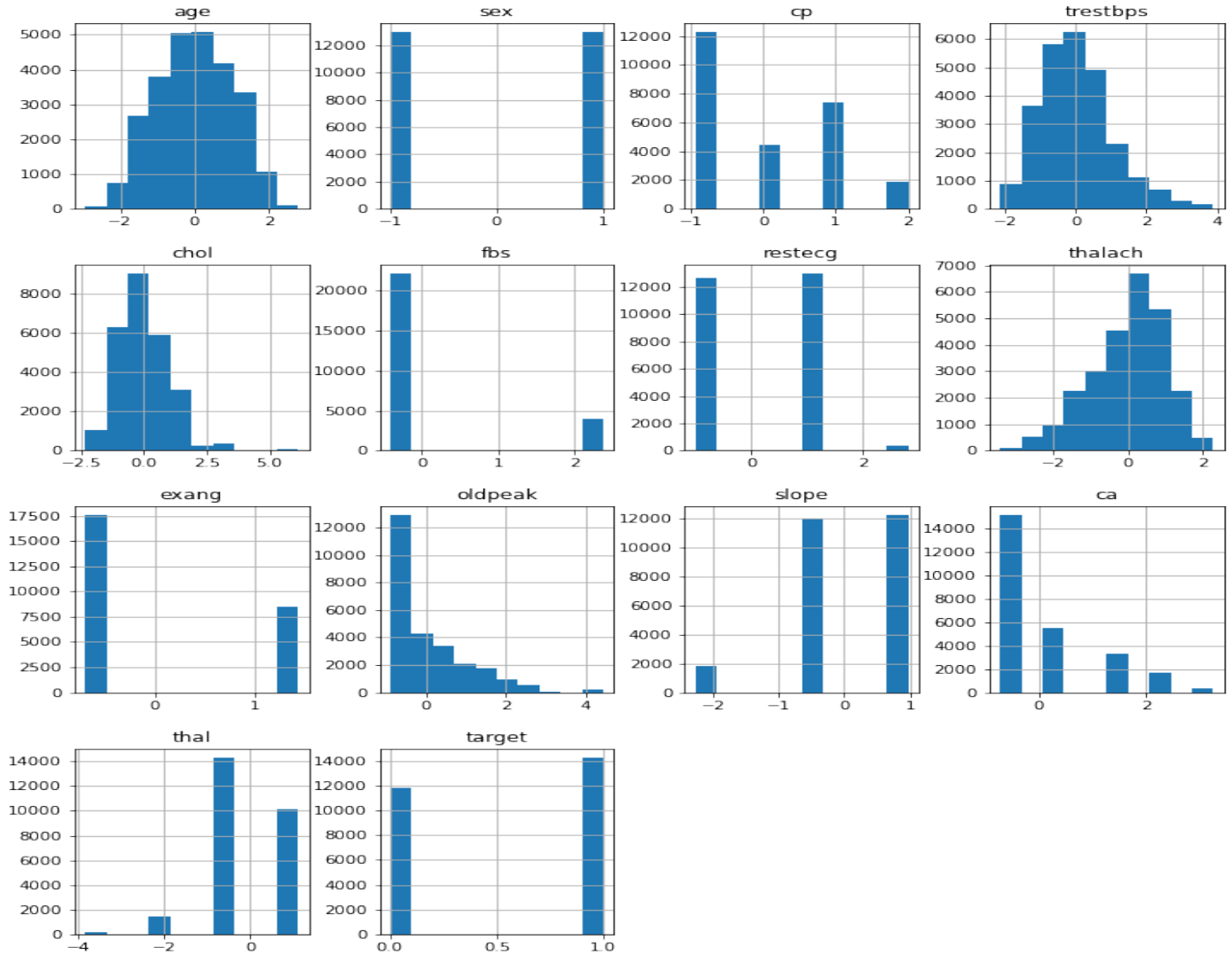


Figure 1: Input Data Distribution Histograms - Before Normalization

2.2 Data distribution graph of target feature before normalization

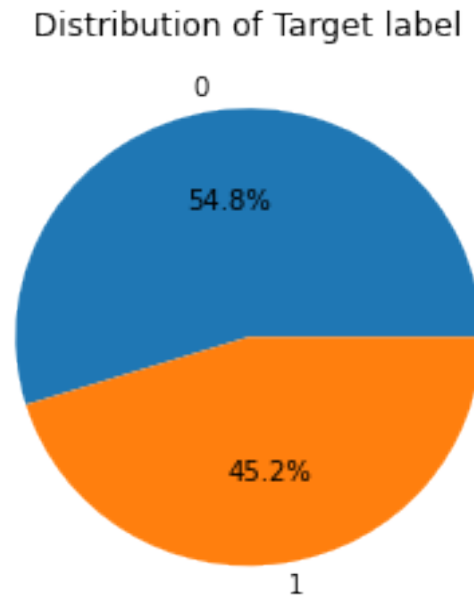


Figure 2: Output Data Distribution

3 Data Cleaning

The dropna eliminates all rows with at least one missing value from the DataFrame. The "sex" column consists of 25 missing records are present. By using dropna eliminates those 25 columns results total 26083 rows in final. As a result, the DataFrame can be used for additional analysis as it no longer has any missing information.

3.1 Splitting of data into training and validation

The data was randomly shuffled and then the dataset was split into training and validation, where 70% of the dataset was allocated for training and 30% was allocated for validation.

3.2 Data Normalization

Data normalization is the process of converting numerical data into a common format so that it may be more easily compared and understood. Reducing redundancy, minimizing inconsistencies, and enhancing data quality are the objectives of data normalization. The data is often scaled to a certain range or transformed into a standard distribution as part of normalization processes. This can make it simpler to compare distinct features in a dataset by reducing the impact of size or magnitude variations between them. Data normalization methods include Min-Max scaling, Z-score normalization, and Decimal scaling.

Mean Normalization Formula

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Z-Score Normalization

$$X_{normalized} = \frac{X - X_{mean}}{X_{standard deviation}}$$

3.3 Data distribution graphs of each attribute after normalization

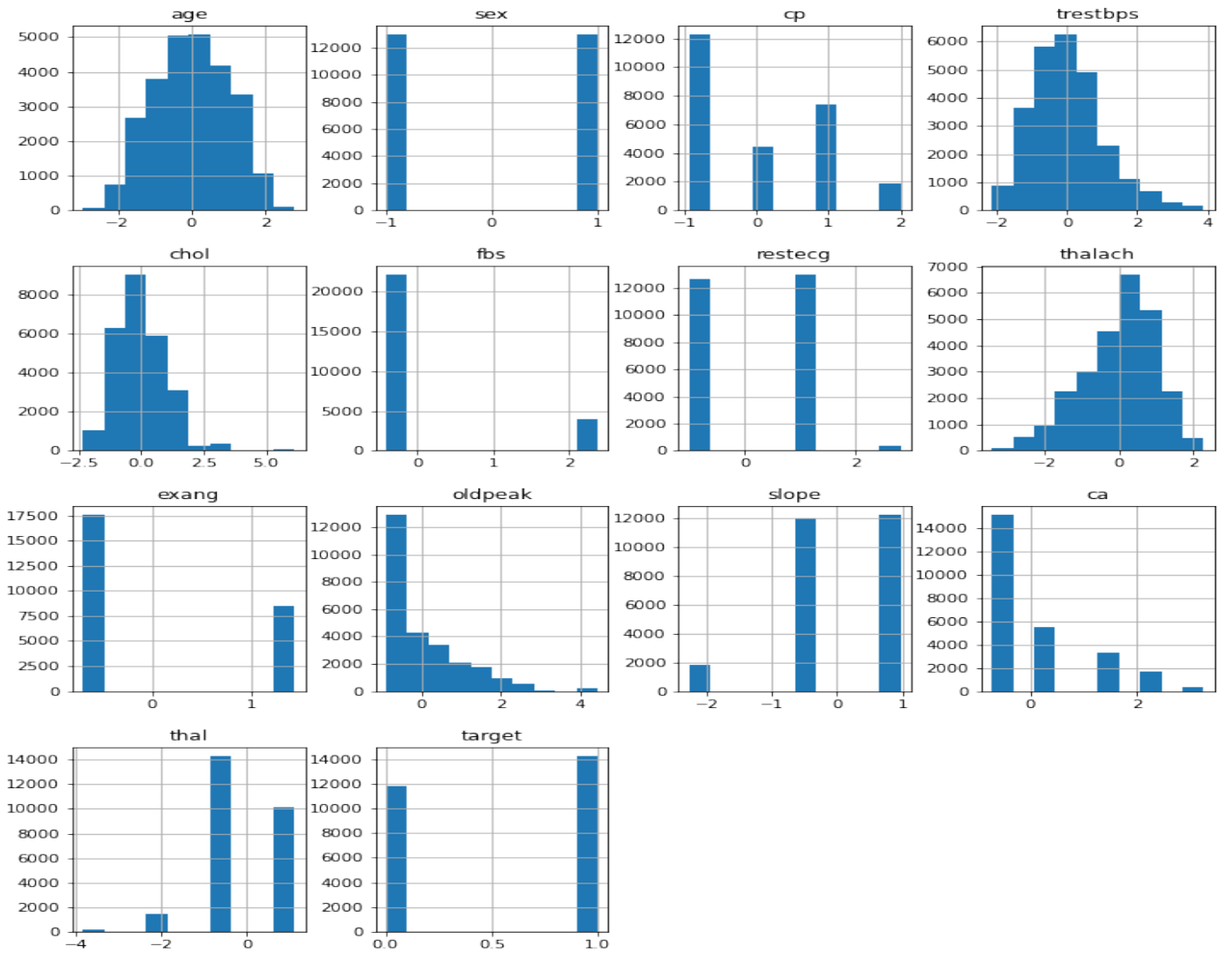


Figure 3: Input Data Distribution Histograms - Before Normalization

4 Building a Model

In this module, we use Artificial neural networks using Tensorflow to build the model. At first, we shuffle the rows of the dataset in order to avoid bias during the model-building phase. Secondly, we split the data into the training set and validation set. The first 30% data is the validation set and the remaining 70% data is the training set. And then we aim to obtain the highest accuracy using training and validation sets by gradually increasing the dense layers and changing the neurons.

NOTE: All models were compiled and fitted on May 3, 2023.

4.1 Logistic Regression Model

The logistic regression model consists of a single layer called the output layer. The single neuron in this layer implements the logistic function to generate the output. This logistic regression model is used as the baseline model for comparison. The number of hidden layers was then gradually increased to a maximum of 2 hidden layers.

4.1.1 Candidate Models

After applying the logistic regression model, we will gradually increase the number of dense layers by changing the number of neurons to obtain the maximum accuracy. In this phase, we achieve the best accuracy with a smaller model when compared to the previous phase.

Hidden	Epocs	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
0 Layer	10	3.2948	0.6838	3.3955	0.6899
1 Layer	10	0.2060	0.9114	0.1970	0.9140
2 Layer	10	8.8933e-04	1.0000	6.6193e-04	1.0000

Table 1: Performance comparison for different hidden layers

Above Table 1 is the demonstration of Training Loss, Training Accuracy, and Validation Loss, and Validation Accuracy of the model with different dense layers. At the 0th Layer, the Validation accuracy is at 0.8149, and for the 1st layer, the Validation accuracy is at 0.9431, and at the 2nd layer, the Validation accuracy is at 1.000. After adding 1,2 hidden layers we can observe that there is a gradual improvement in the Validation Accuracy.

4.2 Learning Curve of Selected Neural Network Architecture

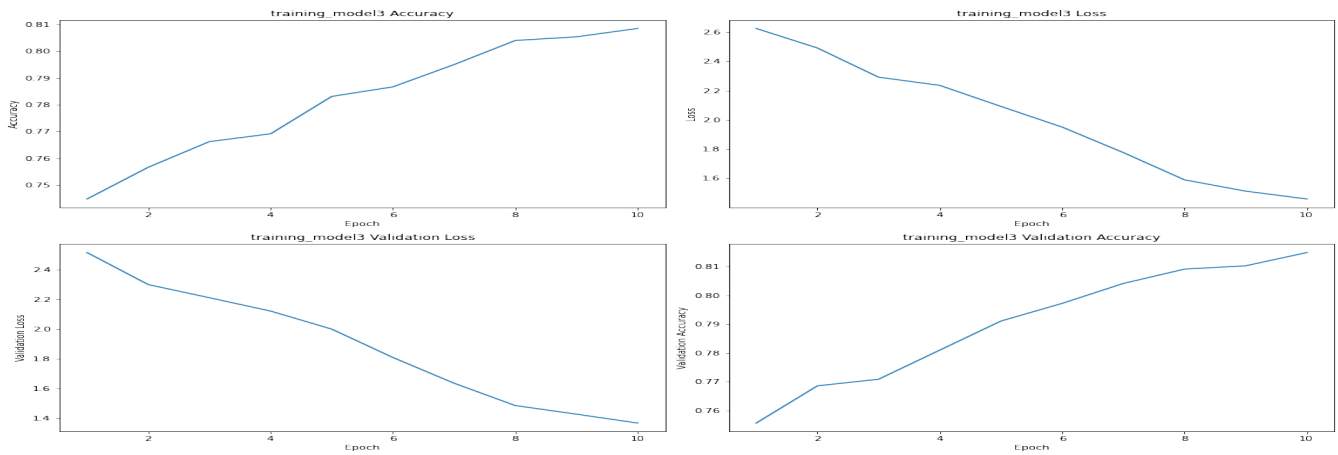


Figure 4: curve showing the change in loss/accuracy vs epoch

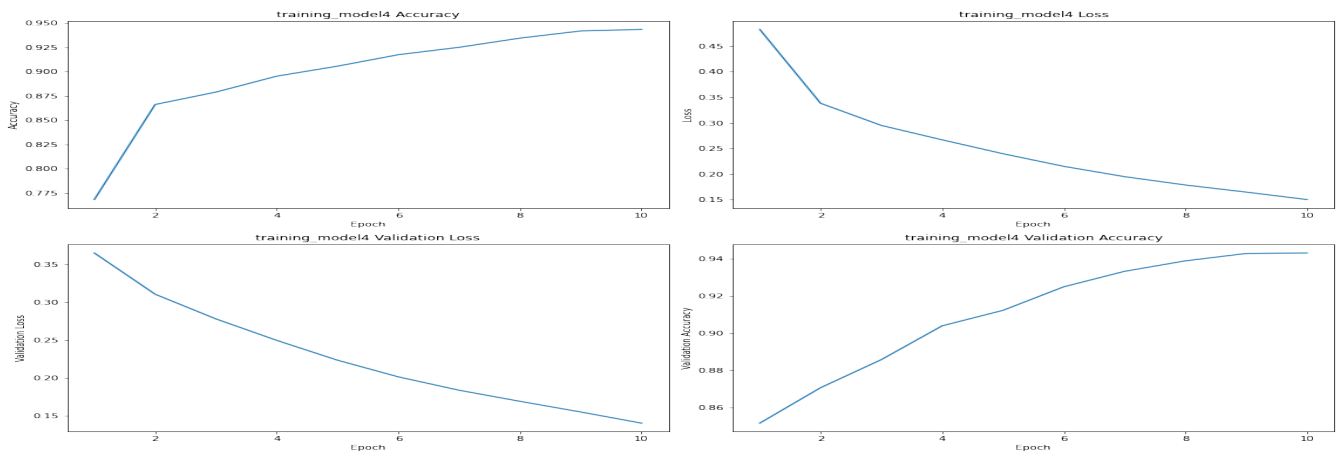


Figure 5: curve showing change in loss/accuracy vs epoch

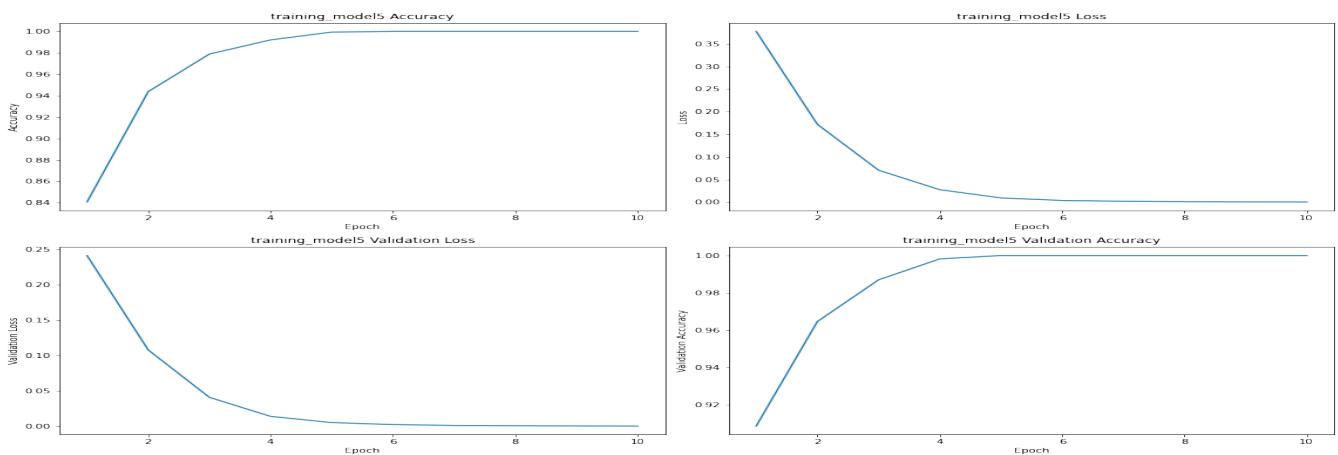


Figure 6: curve showing change in loss/accuracy vs epoch

5 Model Evaluation

The table given below shows the precision, recall and f1 score for the neural network model.

Precision indicates how accurately the model's optimistic predictions came true. It measures the proportion of true positives (TP) to the total of both true and false positives (FP).

Recall evaluates the model's accuracy in identifying positive cases. Recall is often referred to as sensitivity or the real positive rate. The ratio of true positives (TP) to the total of true positives plus false negatives (FN) is what determines this.

F1 score: The harmonic mean of recall and precision is the F1 score. It offers a balanced measurement by combining recall and precision into a single statistic.

Model	Precision	Recall	F1-Score
Baseline Model	73.50	68.67	71.00
1 Layer	89.12	96.18	92.52
2 Layer	100.00	100.00	100.00

5.1 Test Accuracy

It shows the trade-off between the true positive rate and the false positive rate. The ROC curve is a valuable tool to evaluate the accuracy of a binary classifier model. A model can be considered to have good predictive performance if its ROC curve lies at the top-left corner of the plot (i.e., has a high true positive rate and a low false positive rate).

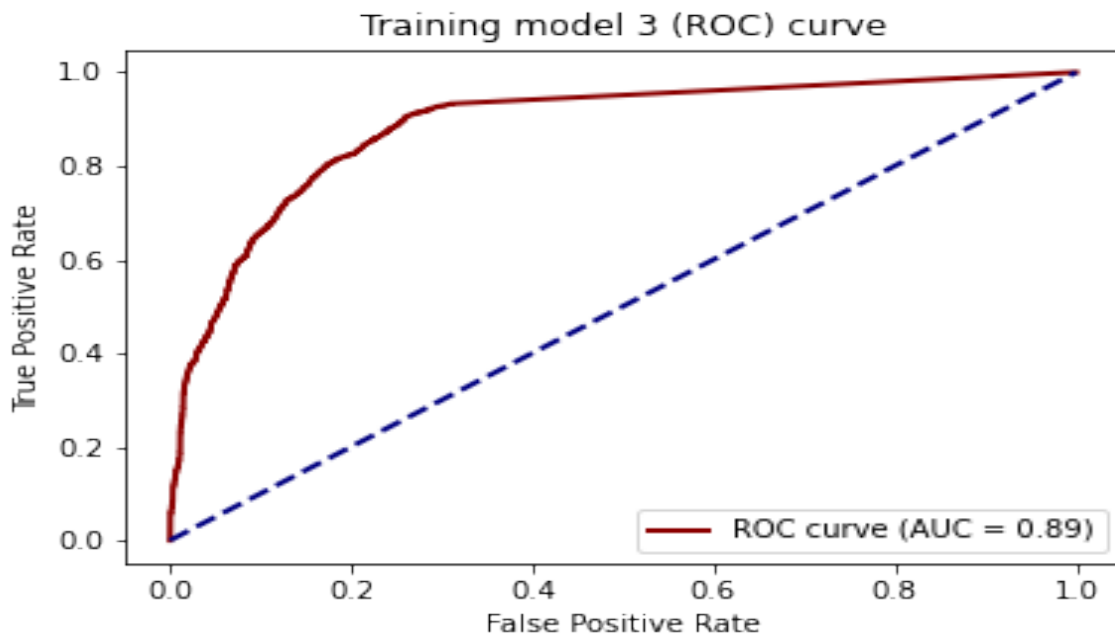


Figure 7: ROC Plot

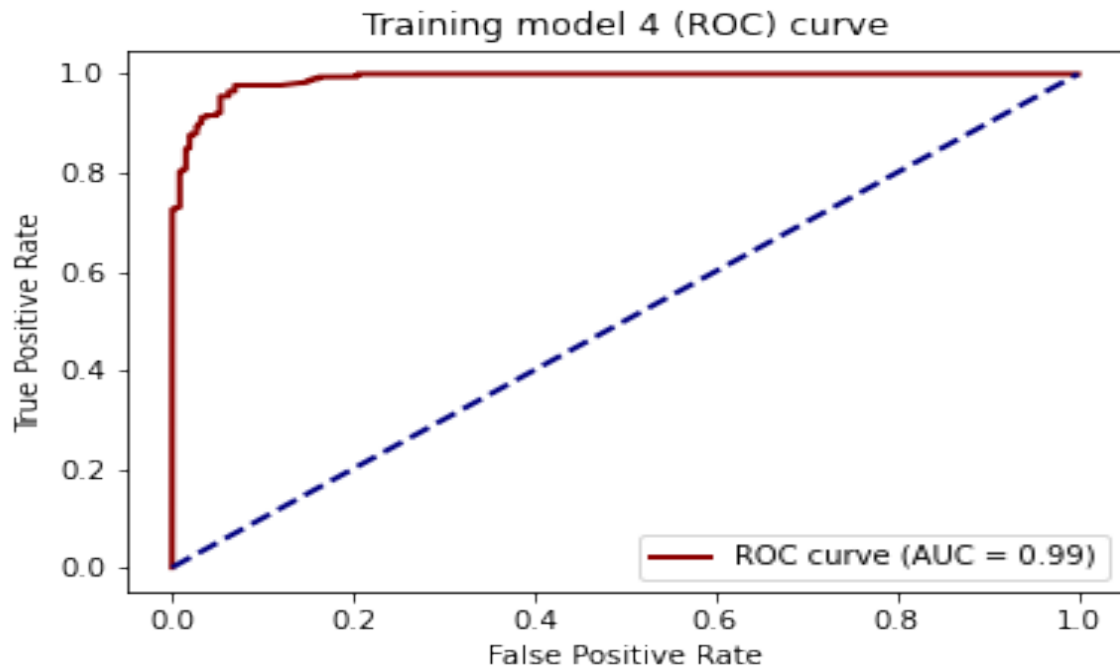


Figure 8: ROC Plot

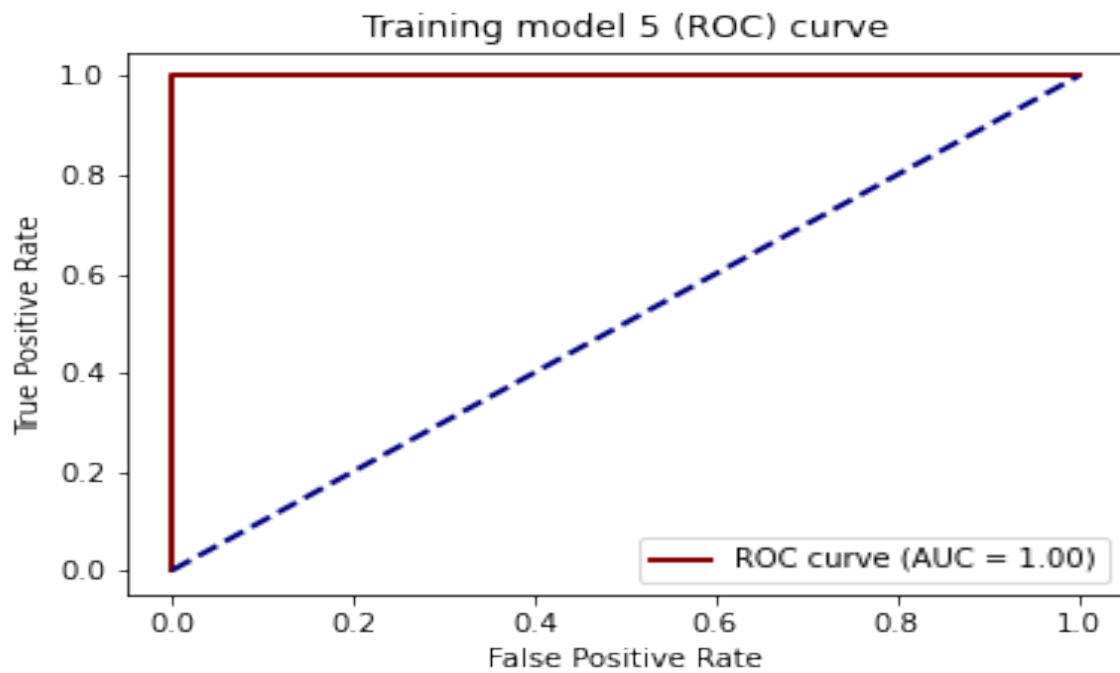
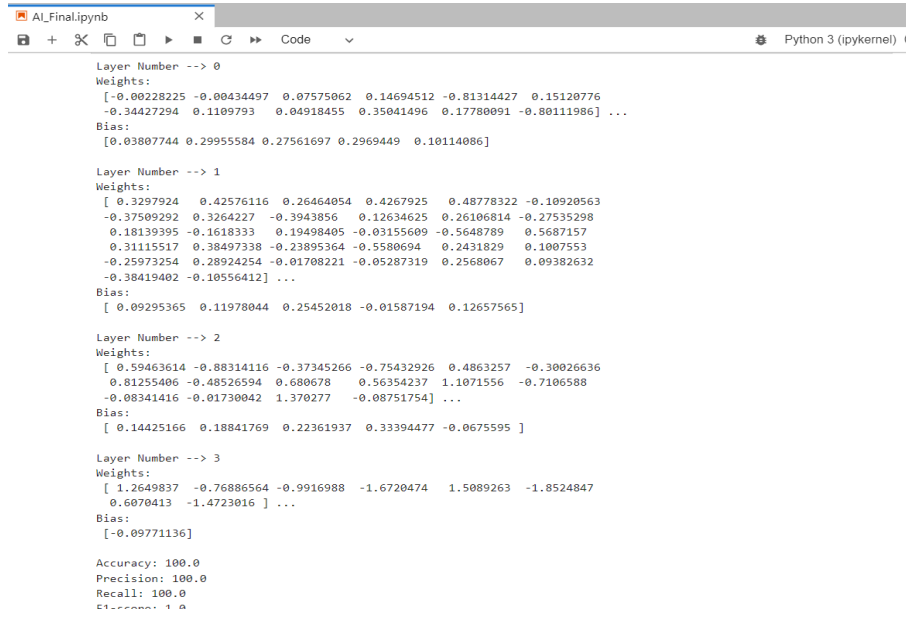


Figure 9: ROC Plot

5.2 Custom Function and Keras Function



```
Layer Number --> 0
Weights:
[[-0.00228225 -0.00434497  0.07575062  0.14694512 -0.81314427  0.15120776
 -0.34427294  0.1109793  0.04918455  0.35041496  0.17780091 -0.80111986] ...
Bias:
[0.03807744 0.29955584 0.27561697 0.2969449 0.10114086]

Layer Number --> 1
Weights:
[[ 0.3297924  0.42576116  0.26464054  0.4267925  0.48778322 -0.10920563
 -0.37509292  0.3264227  -0.3943856  0.12634625  0.26106814 -0.27535298
 0.18139395 -0.1618333  0.19498405 -0.03155609 -0.5648789  0.5687157
 0.31115517  0.38497338 -0.23895364 -0.5580694  0.2431829  0.1007553
 -0.25973254  0.28924254 -0.01708221 -0.05287319  0.2568067  0.09382632
 -0.38419402 -0.10556412] ...
Bias:
[ 0.09295365  0.11978044  0.25452018 -0.01587194  0.12657565]

Layer Number --> 2
Weights:
[[ 0.59463614 -0.88314116 -0.37345266 -0.75432926  0.4863257  -0.30026636
 0.81255406 -0.48526594  0.680678  0.56354237  1.1071556 -0.7106588
 -0.08341416 -0.01730042  1.370277  -0.08751754] ...
Bias:
[ 0.14425166  0.18841769  0.22361937  0.33394477 -0.0675595 ]

Layer Number --> 3
Weights:
[[ 1.2649837 -0.76886564 -0.9916988 -1.6720474  1.5089263 -1.8524847
 0.6070413 -1.4723016 ] ...
Bias:
[-0.09771136]

Accuracy: 100.0
Precision: 100.0
Recall: 100.0
----- 1.0
```

Figure 10: Significance of each input

6 Feature Significance and Reduction

To increase the precision and effectiveness of a model, feature significance and reduction are crucial machine learning strategies.

The importance of a feature is how well it predicts an outcome, or how useful it is as an input. Because not all features are equally important to the accuracy of the model, it is crucial to determine the essential features.

On the other hand, feature reduction describes the procedure of choosing a subset of the most crucial features out of the initial collection. This is done to make the data less dimensional, which can increase the model's effectiveness and prevent overfitting.

6.1 Feature Significance

We can see that the first 4 input features i.e. (age, sex, fbs, restecg) has the least importance or accuracy when compared to the remaining features. And next 3 features i.e. (trestbps, chol, slope) are somewhat more importance when compared to the first 4 features. The features with high importance are (thalach, oldpeak, exang, ca, cp, thal).

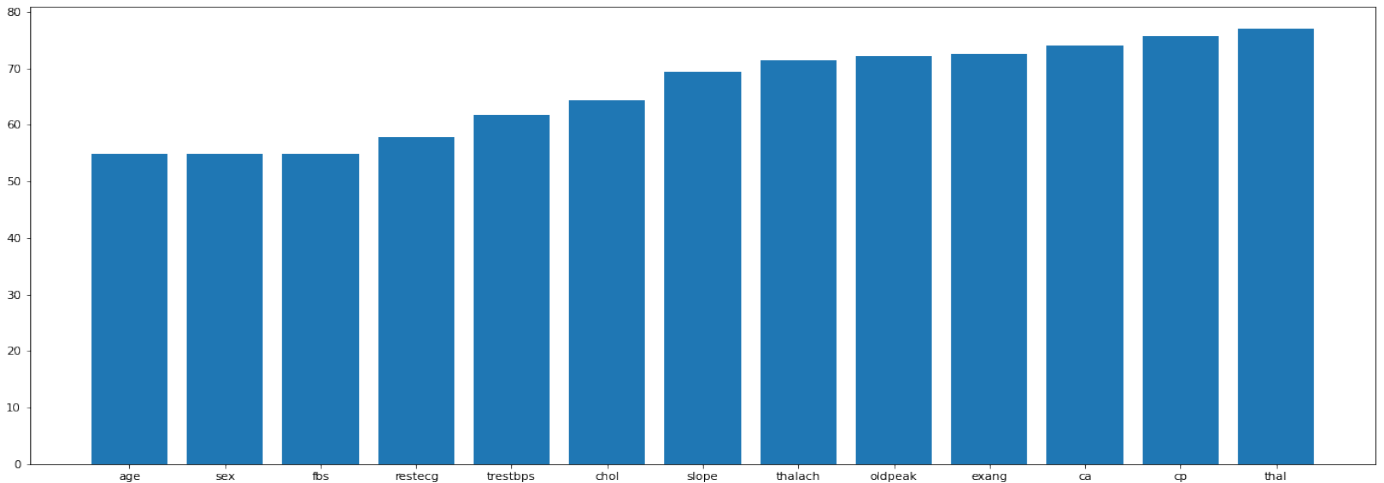


Figure 11: Significance of each input

6.2 Feature Reduction

There is no significant drop in the accuracy even after removing the first 4 input features i.e. (age, sex, fbs, restecg, trestbps, chol, slope, exang). And we can observe a slight drop in the accuracy after removing (thalach, oldpeak). And we can observe a large drop in the accuracy after removing (ca, cp, thal). Refer the graph below.

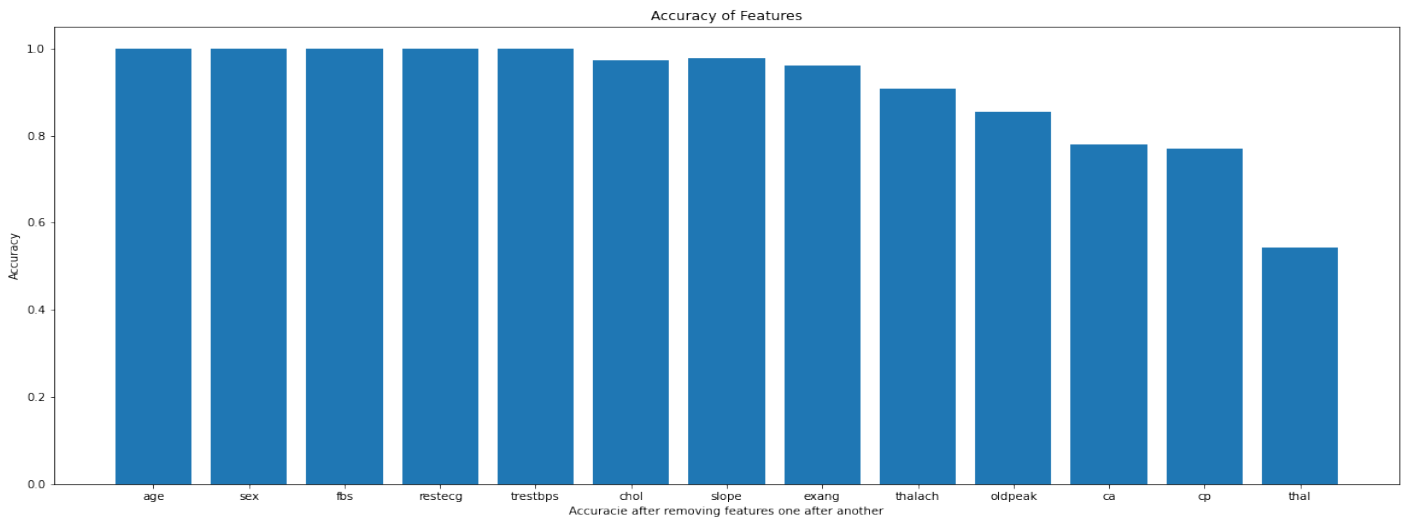


Figure 12: Performance after removing less important features

7 Challenges Faced

In the early phase of this project, choosing a dataset is a bit challenging. I tried different datasets with several combinations of dense layers and neurons, but still, I didn't get good accuracies. After changing several datasets

I choose the current hypertension dataset. I didn't use an early stopping technique which in consequence causing training error and time. In addition, I also faced some challenges in determining whether all the neural networks were sufficient. To solve this dilemma, I did a little research on the receiver operating characteristic curve (ROC) and confusion matrix.

8 Conclusion

In this project, I used Neural networks using tensor flow and keras to predict the presence of hypertension in new patients. In this project, we perform different machine learning techniques like data cleaning to remove missing rows and then the data is split into a validation set and training set. Then, I build a model using this training and validation set by applying artificial neural networks.

This project involves several features to predict the presence of hypertension. It is important to detect the presence of hypertension to avoid different health problems. This information can be used by healthcare professionals to identify patients who need urgent treatment to improve their health outcomes. In this project, I also found the significance of each feature. It is found that type of thalassemia which is a type of blood disorder is the main reason for hypertension in many members.