7088CEM Artificial Neural Networks

Heart Failure Prediction Using CNN & ANN

Surabhi Jarabandahalli Rammurthy

*Student ID: 12801086*

*Data Science and Computational Intelligence Coventry University*

Coventry, United Kingdom [jarabandas@coventry.ac.uk](mailto:jarabandas@coventry.ac.uk)

***Abstract*— Heart failure is a chronic condition that affects millions worldwide. Early prediction helps in providing timely interventions and improve patient outcomes. In recent years deep learning techniques have shown great results in medical image analysis and prediction tasks. The current study evaluates prediction of Heart Failure using two different neural network models which are trained on dataset from an open-source which includes data obtained from observations. We compare different neural networks to select the best architecture. The models used are Artificial neural network and Convolutional neural network. Models will be evaluated using the training and testing data. The outcomes of this project demonstrate the effectiveness of the proposed model in predicting heart failure using clinical and demographic data. Physicians can use the suggested model as a decision-support tool to forecast patients' heart failure and deliver prompt interventions. This serves as a huge deal in terms of critical decision making and potentially reduces the risk in the early stages.**

***Keywords— Convolutional Neural Network, Artificial neural network, python programming, Error analysis, Decision making, Heart disease, Deep learning.***

1. INTRODUCTION

A new universe was unlocked by the invention of artificial intelligence, a ground-breaking development for humanity. It has made incredible advancements in every aspect of living, from simple chatbots to autonomous vehicles and robots.[1] The complicated decision-making process has been strengthened by artificial intelligence (AI), which also supports all computer-aided learning. In Artificial Intelligence the system learns from the previous experiences and data. The amount of data has been increasing rapidly; hence an efficient system is built using AI which has been advanced in many fields. Heart failure illness has become one of the major health problems of our time. The quality of life has been decreasing because of these chronic diseases. These diseases are mainly caused due to the hard breathing and failure of motor function system. According to a study, there will be three to five patients with cardiovascular disease for every 100 patients, making heart failure disease an extremely prevalent disorder. The number of people suffering from this chronic disease is expected to be more than twenty-six million according to the statistics obtained worldwide. This has been found mostly among American and European countries. All the factors directly affect the life span of the person hence early detection of the disease is an important factor which can be achieved through artificial neural network and convolutional neural network.

To identify clinically important parameters associated with CHD in patients, such as heart rate and axis deviation, several studies have used statistical and machine learning and artificial intelligence models on echocardiography images and electrocardiography signals. In the research, boosted algorithms like gradient boost and logit boost have been statistical and machine learning models on echocardiography images and electrocardiography signals. In the research, boosted algorithms like gradient boost and logit boost have been used to forecast FFR and cardiovascular events. Using the 30-day readmission electronic data for patients with heart failure and developed prediction models to identify the prevalence of cardiovascular disease.

Convolutional neural networks are used for classification, segmentation, auto correlation, and image analysis. In order to classify images, such as images of road signs, convolutional neural networks are frequently used. These networks have demonstrated better signal processing accuracy. Convolutional neural networks will be used to better comprehend natural language processing. A convolutional neural network's design will take the shape of a u because the output should take the form of the input, improving images.

1. LITERATURE SURVEY

In the paper [2] (T P, Kumari & N K, 2022) the author has discussed about predicting heart failure by feature ranking analysis where an abundant quantity of data has been produced everyday by health care industry which led to development of various algorithms and correlation techniques. Multiple features were extracted in order to increase the accuracy rate such as blood pressure, heart rate etc. But the author focuses on extracting the best features among the available ones by the application of machine learning and artificial intelligence techniques, The goal is defined to create innovative products for business to achieve key goals. The best feature extraction of this model will be implemented in the current model to increase the performance and prediction.

The author in [3] (Zhang, Kambhampati, Davis, Goode & Cleland, 2022) has depicted the missing value imputation with multiclass classification for clinical heart failure Data. The clinical data could have missed and wrong values. The best-known scheme to overcome this issue us imputation. In these studies, a clinical heart failure data collection was utilized where many imputation techniques were compared

and analyzed for their performance by the application of various algorithms. As per the author the model has given high accuracy and performance.

The paper [4] (Baviskar V., 2022) has built an augmented ensemble heart failure prediction model using Multi parametric analysis. These conditions are caused by a broad range of abnormalities in various body parameters. The current models are limited in terms of accuracy, precision when applied to different type of datasets. The goal of this model is to elevate the performance of underlying text proposes design of an augmented ensemble learning model by the application of several machine learning, artificial neural network, recurrent neural network techniques. This procedure aids in choosing the top-performing model for a specific feature pattern, enhancing the system's general effectiveness. The author has depicted that the model has achieved the accuracy of 99.5% on various datasets. In the current model we aim to increase the precision even when different datasets are applied from the above survey.

In the paper [5]( Yazid, M.,2017) the author aims to predict the mortality of patients admitted for ADHF (acute decompensated heart failure). The model uses the artificial neural network for the process of prediction. A multi -layer feed forward network and back propagation algorithm is used with several variations. Here the scaled conjugate algorithms are used as they work well for a variety of case study sets and work especially well for networks with a lot of weight and bias. Neural network with 32 hidden layers has better performance compared to others. As per the author the model has achieved 91% accuracy. We incorporate the feature of neural network methodology implemented on large datasets to the current model.

The paper [6] (Saravanan & Swaminathan, 2022) depicts the hybrid machine learning model for heart failure. The prevalence of heart disease is dramatically rising as a result of the large number of heart failure cases that go undiagnosed. In order to take necessary precautions a model has been built using machine learning algorithms k-means and support vector machine. With the aid of k-means clustering, the data will first be divided into six groups, and these clustered data will then be used to make predictions about heart failure. The model has outperformed many previous models with the precision and accuracy. We adopt the prediction of likeliness of heart diseases prediction at during the early stages.

1. DATASET

The heart failure dataset includes clinical and laboratory data on 299 heart failure individuals. A total of 1200 observations were produced by combining several easily accessible databases. This dataset has been taken from an open- source Kaggle. A total of 1200 observations were produced by combining several easily accessible databases. One objective variable—the binary outcome of death from heart failure—and 12 clinical features make up the dataset.

All the features are utilized to combine the datasets which helps in research and experimentations.

The main features included in the dataset are as follows.

|  |  |
| --- | --- |
| Age | Age of patient(years) |
| anaemia | Whether the patient has anemia (boolean) |
| creatinine\_phosphokinase | Level of CPk enzyme(mcg/L) |
| diabetes | Diabetes condition(boolean) |
| Ejection\_fraction | proportion of blood that is lost during each contraction of the heart(%) |
| High blood pressure | whether or not the subject has high blood pressure(boolean) |
| Platelets | number of platelets per milliliter of blood (Kilopatelets/mL) |
| Serum\_creatinine | serum creatinine concentration in the blood (mg/dL) |
| Serum\_sodium | systemic sodium concentration in the blood (mEq/L) |
| sex | Gender of the patient(Male,Female,Others) |
| smoking | Whether the patient smokes or not (boolean) |
| time | Follow-up period (days) |

The current model helps in the early detection of risk so that necessary actions can be taken to prevent the mortality of the patients.

Below mentioned is the link for the dataset in Kaggle.

[https://www.kaggle.com/code/theeprologue/heart-failureprediction-logistic-regression-ann/data](https://www.kaggle.com/code/theeprologue/heart-failureprediction-logistic-regression-ann/data%20)

1. METHEDOLOGY

The current model has been built using python coding language. The steps followed in this implementation has been depicted below. This code has been executed using jupyter notebook which is a part if Anaconda software. Necessary libraries and packages have been used in order to implement the model. Following sections will depicts the methods and procedures followed for the implementation.

DATA PRE-PROCESSING

It is the crucial step of implementation which involves the process of converting the raw data into a data suitable for the analysis. This process typically clean, transform and reshape the data into usable format.

We will first load all the essential libraries for our model. We will load the packages required for performing the data pre-processing from sklearn

1. DATA LOADING

The CSV file consists of all the heart failure clinical dataset having the biological features of the patients, The dataset can be used to train and test the implemented models. The pandas library has been used to read the csv file as ‘pd.read\_csv()’ function which creates a data frame.

The various tools and techniques offered by the panda’s library can be used to process, evaluate, and use the data for machine learning tasks after it has been loaded into the data frame object.

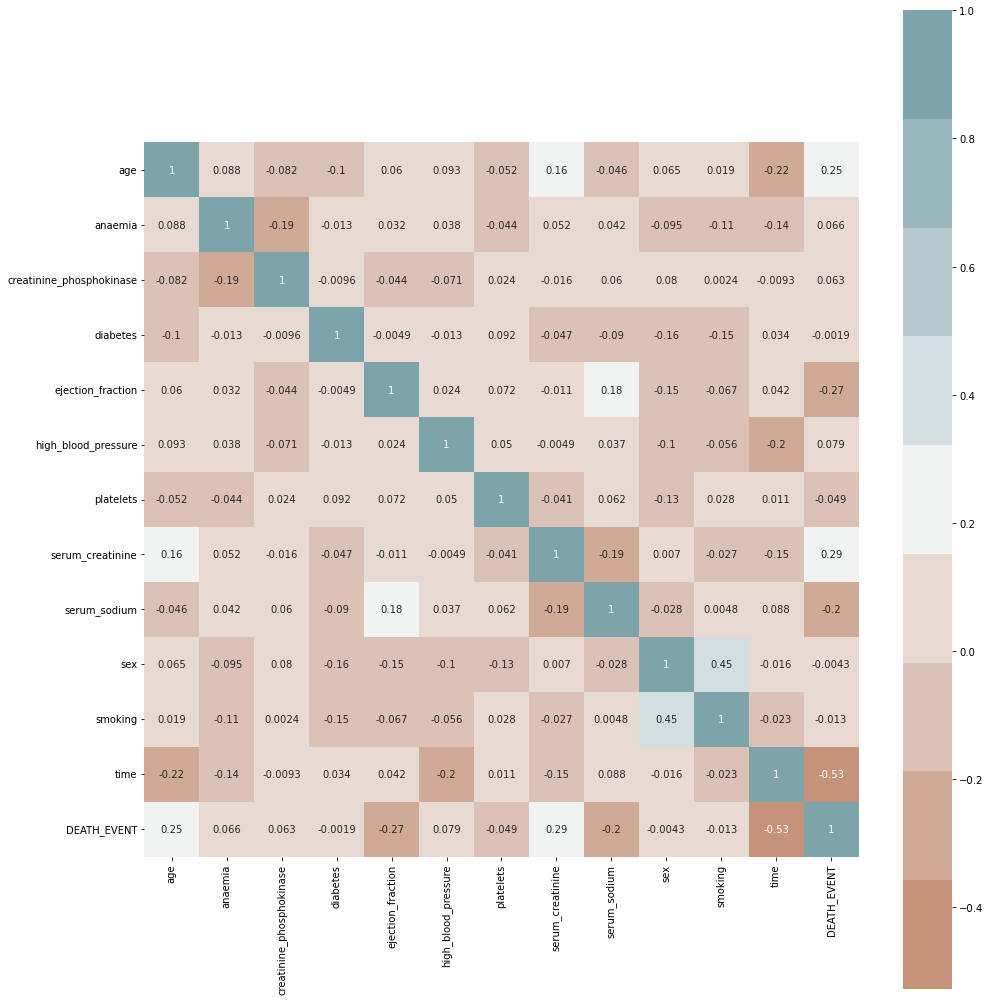
1. ANALYSIS OF DATA

The analysis is done as We'll choose the goal variable and see if the data is imbalanced. The goal variable's count plot can be used to assist with this. For determining the relationship between the characteristics, we can use a correlation matrix. Here we determine the relationships and patterns between the variables and the visualizations are done to see these trends in the data.

Below we have plotted the variables Count against the Death\_Event and the correlation matrix of all features with the het map is defined.



*Fig:1*

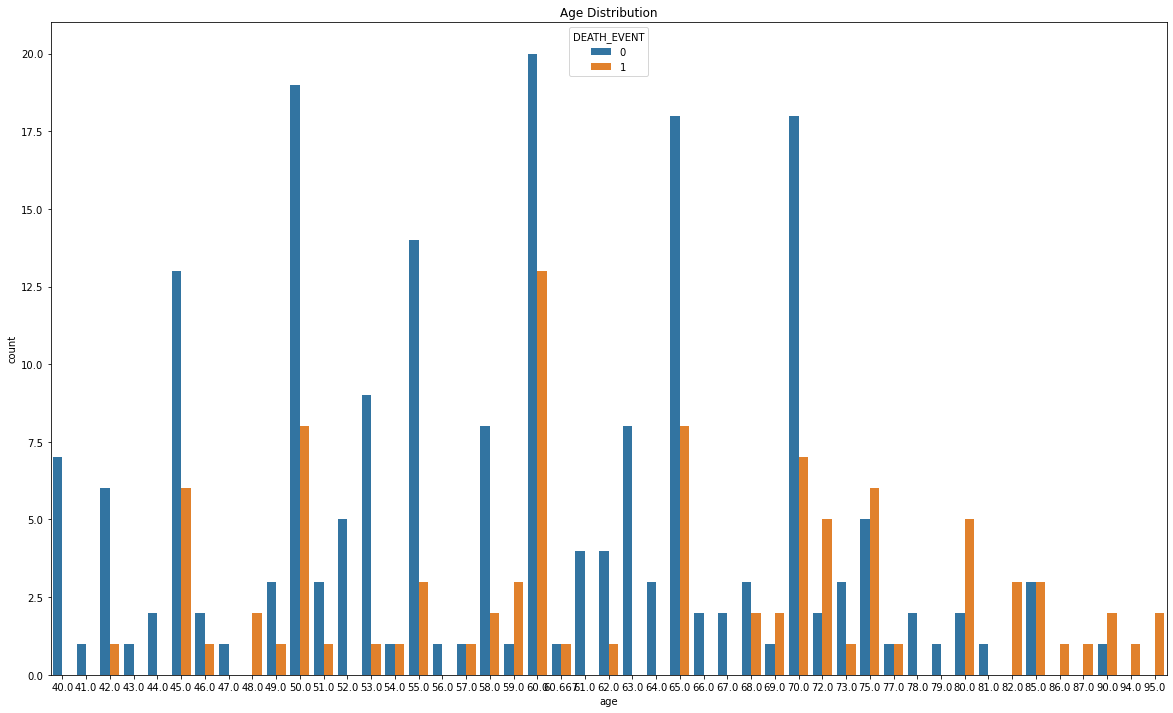


*Fig:2*

The correlation matrix shows the co-efficient between different variables. This statistical measure indicates the a two-variable linear relationship's strength and path.

1. VIZUALIZING THE DATA

To view the data as plots, we will execute the data visualization. The facts can be more easily understood visually. A couple of plotted plots for the data are shown below for the ‘age distribution’. Data visualization should be easy to understand, concise, and straightforward. Additionally, it ought to be customized for the target market and the analysis's objectives.

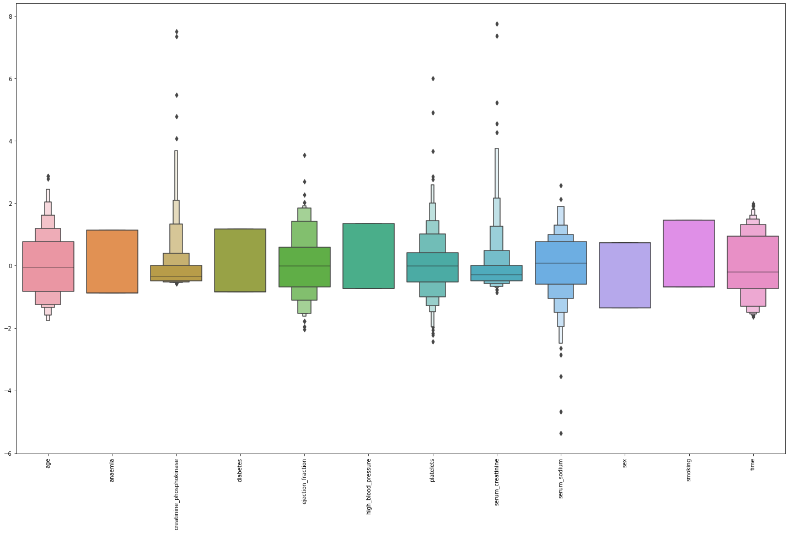


*Fig:3*

1. FEATURE SCALING

Feature scaling is used to standardize or normalize a dataset's assortment of features and factors. Since distance measurements are sensitive to feature scale, it is possible for some features with a bigger scale to predominate over those with a smaller scale if the features are not scaled.

The StandardScaler() method will be used to preprocess the data. For each of the statistics in this column, the mean and standard deviation will be calculated. The scaled characteristics are shown in the figure.



*fig:4*

1. DATA SPLITTING

Data splitting is the process of dividing the data into subsets of two or more. In the typical machine learning or ANN models this data is slitted into training and testing dataset by using the train\_test\_split function. The ratio of these datasets depends on many factors such as size, complexity and time. These datsets will be used to train the implemented models and test those models for the accuracy and performance.

Further predictions will be done by obtaining the results from these models. The feature of updating and training the model with new data is also done over time.

METHODS AND ARCHITECTURE SELECTION

1. ARTIFICIAL NEURAL NETWORK

A form of machine learning algorithm called an artificial neural network (ANN) is modeled after the structure and operation of the human brain. It is utilized to discover intricate connections and patterns in data, making it appropriate for a wide range of uses, involving diagnosis in medicine.

We use ANN for our model as ANN can classify new patients as having or not having heart problems based on their signs, history of illness, and other pertinent factors by learning from a big dataset of patients with known outcomes also it can detect the root cause and associated risks that cause heart attacks. This model is useful in heart disease as it includes many factors as: age, gender, blood pressure, cholesterol, and family background. It is capable of capturing many complex relationships and able to provide great insights from the data.

All-inclusively ANN is chosen as they can correctly classify patients based on their risk of getting heart disease and can learn from vast amounts of complex data.



*fig:5*

We refer to the network as the deep Neural Network as the number of hidden levels rises. (DNN). The graphical depiction of the neural network model is shown in figure 5. Once the pre-processing of the data is completed, we will build the model by utilizing the training data. Using the uniform kernel start, I trained the model using the activation function as RELU. The Model contains 3 hidden layers where each one of the layer consists of 16,8,4 number of nodes. Once the process of training the model is

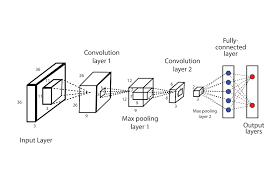
finished, we make use of ADAM optimizer for the purpose of compilation. Following that, this model will be trained using training data with a batch size of 32 and 500 epochs. The predictions will be done using this about the accuracy and the precision of the model which will described in the next section.

1. CONVOLUTIONAL NEURAL NETWORK

Convolutional neural network is mainly utilized for the purpose of image classification. However in the context of heart failure detection ECG (electrocardiogram) signals can be used to teach a CNN to recognize patterns that point to heart failure. The CNN would be programmed to acquire features that are important for differentiating between healthy and ill patients from the raw ECG signal as input.

CNN for heart failure detection involves obtaining a labeled dataset of ECG signal measurements, preprocessing the data, creating a suitable network architecture, training the CNN, and assessing the CNN's success on a different test set. A CNN can achieve high accuracy in identifying heart failure from ECG signals with enough data and careful tuning of the network design and training parameters.

Below figure shows the overview of convolutional neural network.



*fig:6*

The inclusion of manual data handling will aid us in avoiding issues with garbage in and out. CNN is trained in such a way that without using the manual features this will learn the original input data in a hierarchal process which makes CNN to learn on its own as human brain.

In order to implement the CNN model, we have utilized many packages from tensor flow. RELU and sigmoid activation functions are used for the purpose of implementing the model. This will make it simple to understand a neuron's activity rates, and the RELU function will help us make our model run more quickly.

We do not have the activation unit in the output as the output layer will produce the arbitrary output by itself. For the purpose of implementation, we have 4 hidden layers in the model. The model is optimized and produces

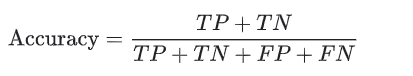
good results which will be discussed briefly in the next section.

V. EXPERIMANETAL RESULS

Following model implementation, test data will be used to determine the model's correctness as now the models have been implemented, this section will explain about the results obtained from the models. Accuracy is one of the metrics to evaluate the classification models. It is the fraction of predictions that our model got correct values. As we are predicting a binary classification problem the accuracy of the model will be calculated in terms of negatives and positives.

ANN RESULTS:

With the help of ANN model, I was able to get an Accuracy of 68%.



Where TP is the True Positives, which measures the extent of model for which the model will correctly predict the positive class.

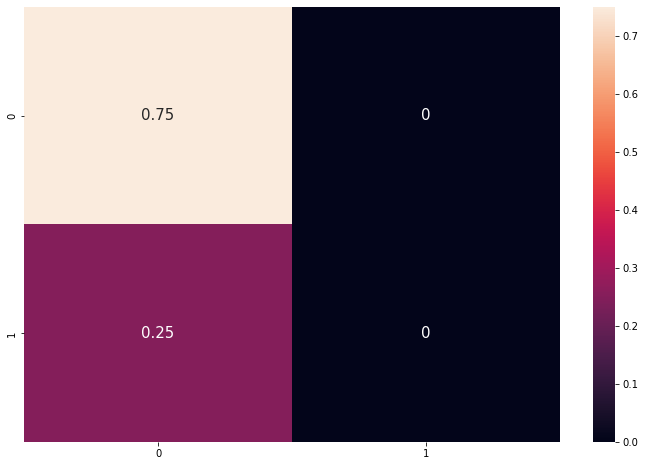
TN is the True Negative where the model will predict the negative case when actual case is also negative.

FP is the False Positive for which the model will predict the case belongs to one case, but the model will predict the case belongs to another class.

FN is the False Negative for which the model predicts the class as negative, but it is actually positive.

The below plot known as confusion matrix is plotted against these values which will help us in analyzing the results in visualized way.

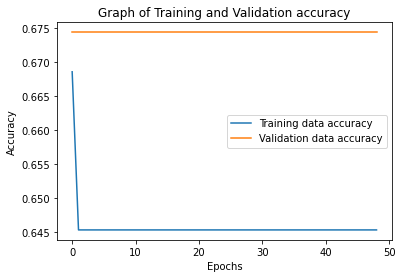
Confusion Matrix:



*Fig: 9*

Although accuracy is an important factor for calculating the performance of the model, it alone won’t be sufficient for deciding the performance of the model. Another most important factor is to use a graph that will make us understand the progress of Neural Networks which is an Accuracy Curve. The curve is plotted for both validation and training accuracy

Below figure shows the accuracy curve using ANN model:

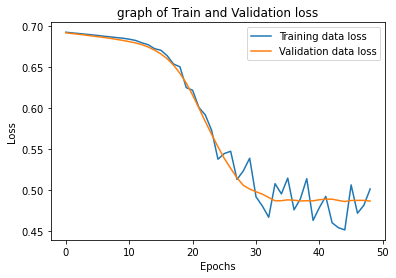
**

*fig: 7*

We can observe that the gap between the validation accuracy and training accuracy will give the clear indication for the overfitting. The more the gap the more will be the overfitting.

The other most important plot to debug the neural network is Loss Curve. It will give the plot for the training process and the direction in which the neural network is used. It shows how bad our model is performing with respect to training and validation losses.

Below figure shows the loss curve using ANN model:

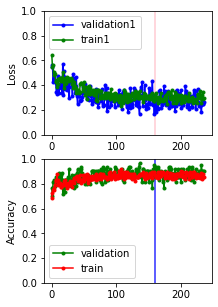


*fig: 8*

CNN RESULTS:

According to evaluation metrics, the suggested method outperformed the current methodology, and the proposed model was able to achieve an accuracy of 96%.

The figure below represents the accuracy and the loss curve of the CNN model.



*fig:9*

CONCLUSION

This project is implemented by cleaning the data initially and further it is divided into 2 sets where the model is implemented and trained on the slitted training and testing dataset. From all these results, it is observed that CNN model tends to perform more than ANN model and accurate for solving the classification problem. Overall the CNN model is can be trained and implemented in real world scenario.

VI. DISCUSSION AND FUTURE WORK

The Heart failure prediction using ANN and CNN has shown promising results.

* ANN are type of models which will predict patterns using biological data as inputs. These models can be trained with the large datasets and could be used in real- world scenarios.
* CNN would extract the images and the features of them to predict the conditions of the patient. As this model has given great accuracy this can be trained and used in identifying patterns associated with heart failure.

Future Work:

To take advantage of each algorithm's advantages, researchers could look into the use of hybrid models that blend ANN and CNN. The use of transfer learning, which starts with previously learned models and trains new models on smaller data sets, could be investigated by researchers.

This strategy might aid in enhancing the precision of heart failure forecast models with scant data.

VII. ETHICAL AND LEGAL ISSUES

This model is tending to raise many ethical and legal issues which has be taken into consideration.

Informed consent: when the data of a patient is taken, we should make sure that we have taken the consent of the patient. Transparency and accuracy: The results obtained by the model has to accurate and should be conveyed to the concerned healthcare providers correctly without any missing values which lead to further complications.

Privacy and confidentiality: The data obtained from the patient should not be disclosed to any third party at in any case it has to be kept secure. Any loss of data might cause issues hence it should be backed up and kept on a trusted source.

VIII REFRENCES

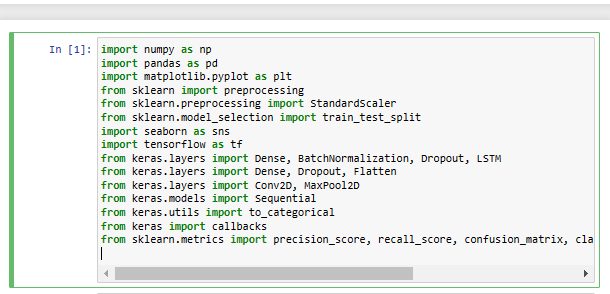
1. <https://arxiv.org/ftp/arxiv/papers/1909/1909.00489.pdf>

1. Pushpavathi, T. P., Kumari, S., & Kubra, N. K. (2021, January). Heart Failure Prediction by Feature Ranking Analysis in Machine Learning. In 2021 6th International Conference on Inventive Computation Technologies (ICICT) (pp. 915-923). IEEE.
2. Zhang, Y., Kambhampati, C., Davis, D. N., Goode, K., & Cleland, J. G. (2012, May). A comparative study of missing value imputation with multiclass classification for clinical heart failure data. In 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery (pp. 2840-2844). IEEE.
3. Baviskar, V., Dwivedi, Y., Mishra, M., Verma, M., & Chatterjee, P. (2022, April). Design of an Augmented Ensemble Heart Failure Prediction Model using Multi Parametric Analysis. In 2022 IEEE 7th International conference for Convergence in Technology (I2CT) (pp. 1-5). IEEE.
4. Yazid, M. H. A., Talib, S., Satria, M. H., & Abd Ghazi, A. (2017, September). Neural network on mortality prediction for the patient admitted with ADHF (acute decompensated heart failure). In 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI) (pp. 1-6). IEEE.
5. Saravanan, S., & Swaminathan, K. (2021, October). Hybrid K-Means and Support Vector Machine to Predict Heart Failure. In 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC) (pp. 1678-1683). IEEE

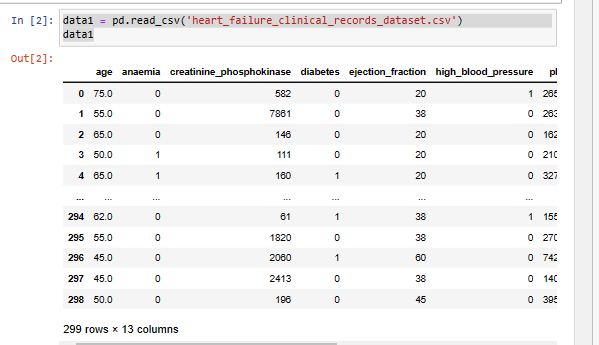
<https://www.kaggle.com/code/andreariba/heart-failurecnn>

<https://www.kaggle.com/code/andreariba/heart-failure-cnn>

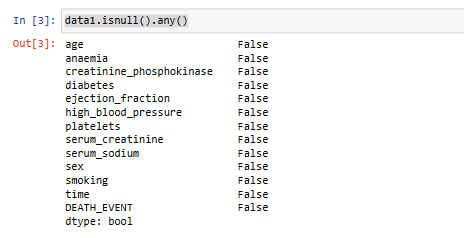
**Appendix 1**



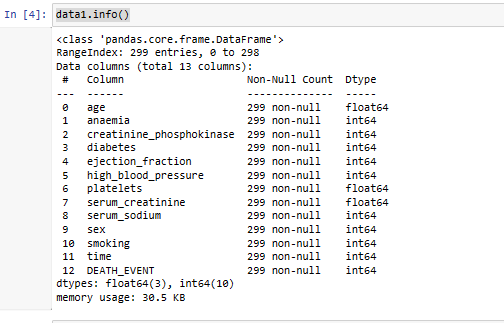
Importing the libraries.



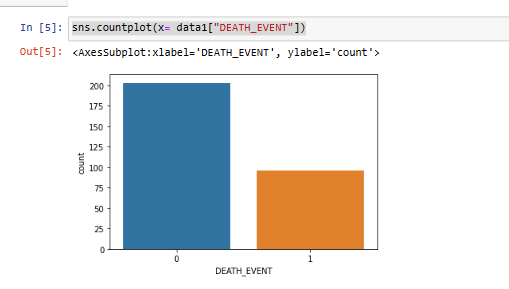
loading the files to view the data.



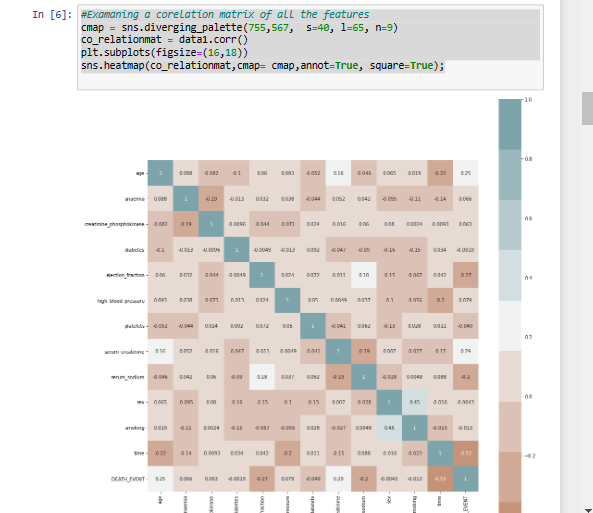
Here we are checking if there are any null values present



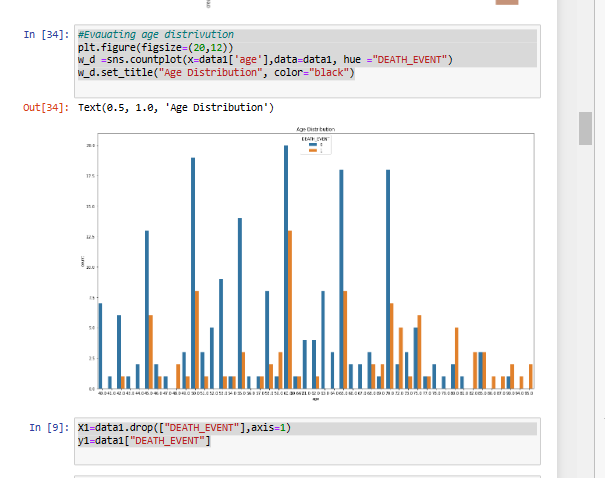
**Checking values of the data set**



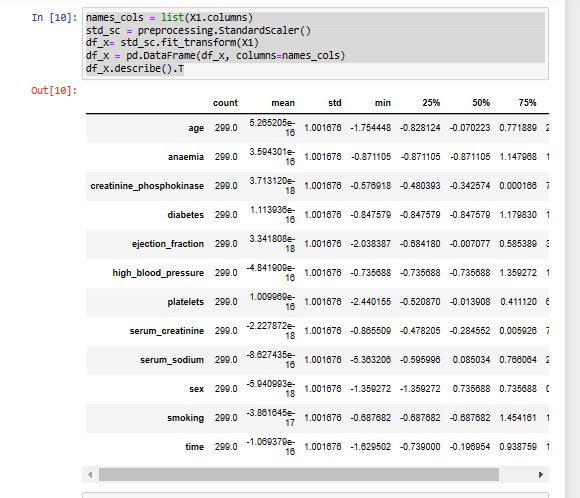
**Plot of the dataset a count plot**

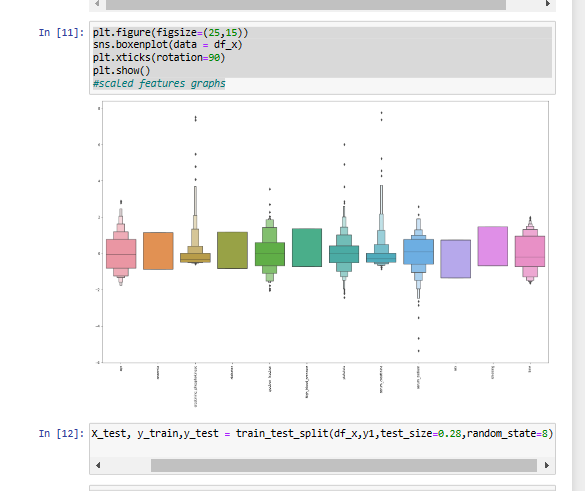


**Here we plot a confusion matrix to see the data from negative to positive values**

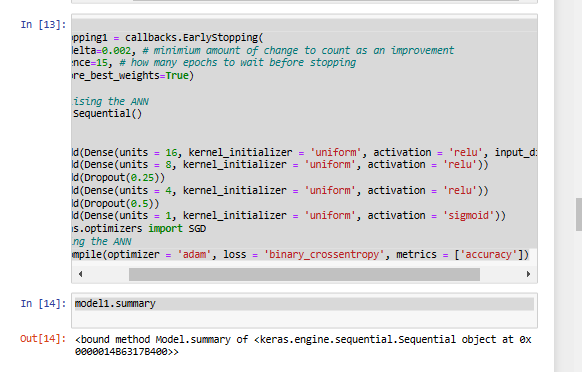


**The above figure shows the age distribution plot in data exploration.**

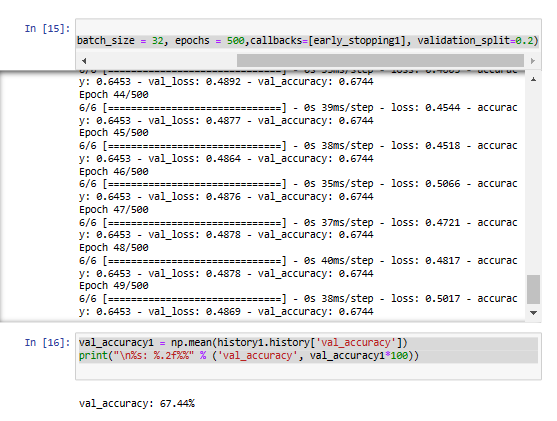




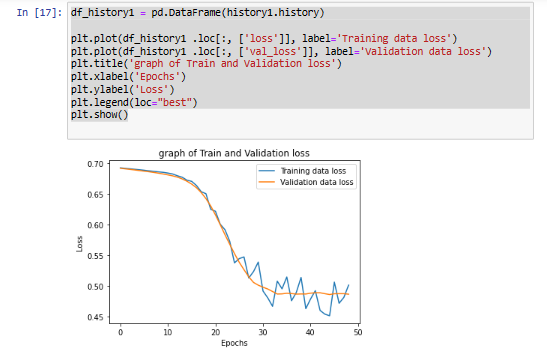
**Here we split the data into train and test data in order to train the model**

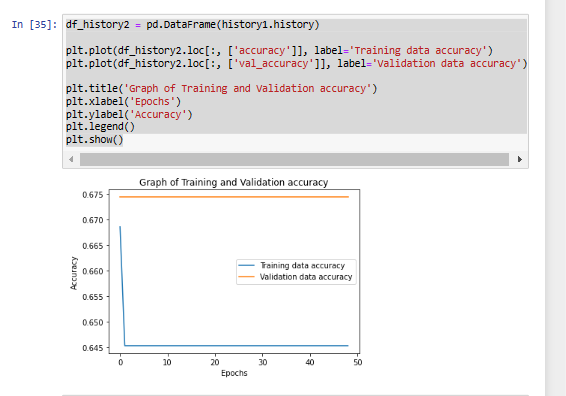


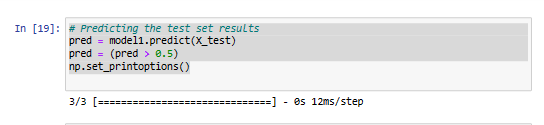
**Building the model using ANN with 2 hidden nodes**

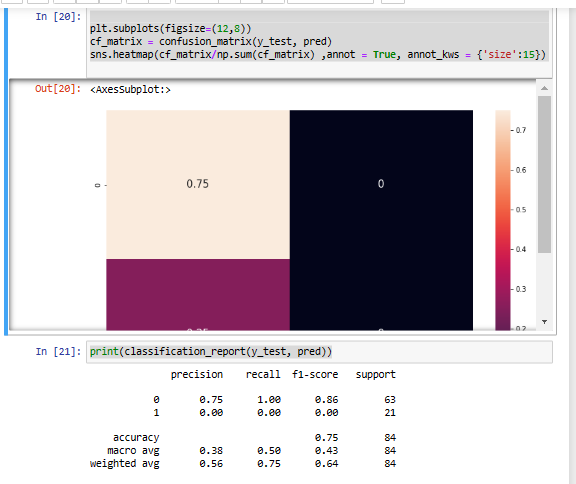


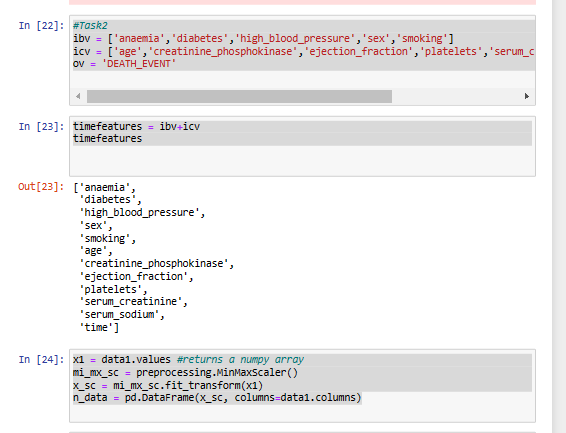
**We execurte in a batch with the value of epoch being 500**

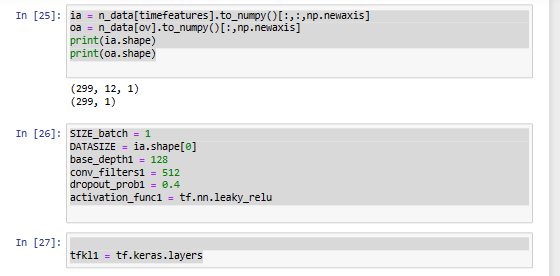




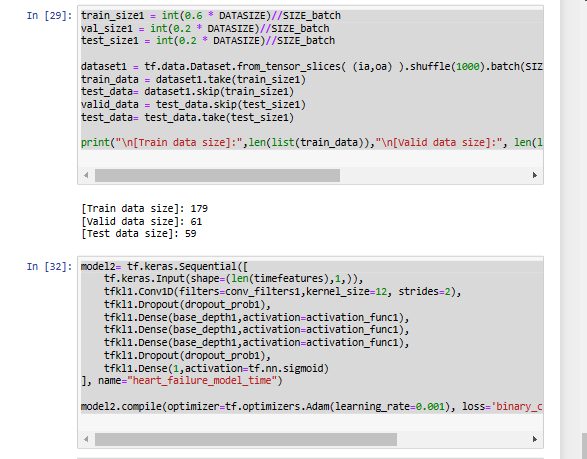


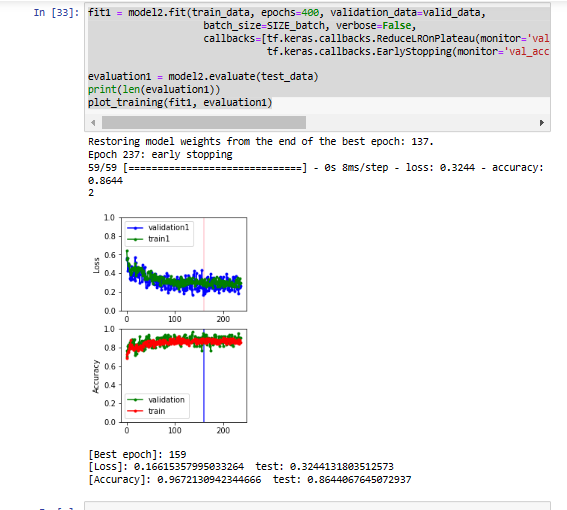












**APPENDIX 2- Code**

**Below provided are the links to the code**

[**https://github.com/surabhijr/HeartFailureDetectionSystem**](https://github.com/surabhijr/HeartFailureDetectionSystem)

**[1]**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn import preprocessing

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

import seaborn as sns

import tensorflow as tf

from keras.layers import Dense, BatchNormalization, Dropout, LSTM

from keras.layers import Dense, Dropout, Flatten

from keras.layers import Conv2D, MaxPool2D

from keras.models import Sequential

from keras.utils import to\_categorical

from keras import callbacks

from sklearn.metrics import precision\_score, recall\_score, confusion\_matrix, classification\_report, accuracy\_score, f1\_score

[2]

data1 = pd.read\_csv('heart\_failure\_clinical\_records\_dataset.csv')

data1

[3]

data1.isnull().any()

[4]

data1.info()

[5]

sns.countplot(x= data1["DEATH\_EVENT"])

[6]

#Examaning a corelation matrix of all the features

cmap = sns.diverging\_palette(755,567, s=40, l=65, n=9)

co\_relationmat = data1.corr()

plt.subplots(figsize=(16,18))

sns.heatmap(co\_relationmat,cmap= cmap,annot=True, square=True);

[7]

#Evauating age distrivution

plt.figure(figsize=(20,12))

w\_d =sns.countplot(x=data1['age'],data=data1, hue ="DEATH\_EVENT")

w\_d.set\_title("Age Distribution", color="black")

[8]

[9]

X1=data1.drop(["DEATH\_EVENT"],axis=1)

y1=data1["DEATH\_EVENT"]

[10]

names\_cols = list(X1.columns)

std\_sc = preprocessing.StandardScaler()

df\_x= std\_sc.fit\_transform(X1)

df\_x = pd.DataFrame(df\_x, columns=names\_cols)

df\_x.describe().T

[11]

plt.figure(figsize=(25,15))

sns.boxenplot(data = df\_x)

plt.xticks(rotation=90)

plt.show()

#scaled features graphs

[12]

X\_train, X\_test, y\_train,y\_test = train\_test\_split(df\_x,y1,test\_size=0.28,random\_state=8)

[13]

#task 1

early\_stopping1 = callbacks.EarlyStopping(

min\_delta=0.002, # minimium amount of change to count as an improvement

patience=15, # how many epochs to wait before stopping

restore\_best\_weights=True)

# Initialising the ANN

model1 = Sequential()

# layers

model1.add(Dense(units = 16, kernel\_initializer = 'uniform', activation = 'relu', input\_dim = 12))

model1.add(Dense(units = 8, kernel\_initializer = 'uniform', activation = 'relu'))

model1.add(Dropout(0.25))

model1.add(Dense(units = 4, kernel\_initializer = 'uniform', activation = 'relu'))

model1.add(Dropout(0.5))

model1.add(Dense(units = 1, kernel\_initializer = 'uniform', activation = 'sigmoid'))

from keras.optimizers import SGD

# Compiling the ANN

model1.compile(optimizer = 'adam', loss = 'binary\_crossentropy', metrics = ['accuracy'])

[14]

model1.summary

[15]

history1 = model1.fit(X\_train, y\_train, batch\_size = 32, epochs = 500,callbacks=[early\_stopping1], validation\_split=0.2)

[16] val\_accuracy1 = np.mean(history1.history['val\_accuracy'])

print("\n%s: %.2f%%" % ('val\_accuracy', val\_accuracy1\*100))

[17]

df\_history1 = pd.DataFrame(history1.history)

plt.plot(df\_history1 .loc[:, ['loss']], label='Training data loss')

plt.plot(df\_history1 .loc[:, ['val\_loss']], label='Validation data loss')

plt.title('graph of Train and Validation loss')

plt.xlabel('Epochs')

plt.ylabel('Loss')

plt.legend(loc="best")

plt.show()

[18]

df\_history2 = pd.DataFrame(history1.history)

plt.plot(df\_history2.loc[:, ['accuracy']], label='Training data accuracy')

plt.plot(df\_history2.loc[:, ['val\_accuracy']], label='Validation data accuracy')

plt.title('Graph of Training and Validation accuracy')

plt.xlabel('Epochs')

plt.ylabel('Accuracy')

plt.legend()

plt.show()

[18]

# Predicting the test set results

pred = model1.predict(X\_test)

pred = (pred > 0.5)

np.set\_printoptions()

[19]

plt.subplots(figsize=(12,8))

cf\_matrix = confusion\_matrix(y\_test, pred)

sns.heatmap(cf\_matrix/np.sum(cf\_matrix) ,annot = True, annot\_kws = {'size':15})

[20]

print(classification\_report(y\_test, pred))

[21]

#Task2

ibv = ['anaemia','diabetes','high\_blood\_pressure','sex','smoking']

icv = ['age','creatinine\_phosphokinase','ejection\_fraction','platelets','serum\_creatinine','serum\_sodium','time']

ov = 'DEATH\_EVENT'

[22]

timefeatures = ibv+icv

timefeatures

[23]

x1 = data1.values #returns a numpy array

mi\_mx\_sc = preprocessing.MinMaxScaler()

x\_sc = mi\_mx\_sc.fit\_transform(x1)

n\_data = pd.DataFrame(x\_sc, columns=data1.columns)

[24]

ia = n\_data[timefeatures].to\_numpy()[:,:,np.newaxis]

oa = n\_data[ov].to\_numpy()[:,np.newaxis]

print(ia.shape)

print(oa.shape)

[25]

SIZE\_batch = 1

DATASIZE = ia.shape[0]

base\_depth1 = 128

conv\_filters1 = 512

dropout\_prob1 = 0.4

activation\_func1 = tf.nn.leaky\_relu

[26]

tfkl1 = tf.keras.layers

[27]

#FUNCTION TO PLOT THE TRAINING

def plot\_training(fit, evaluation):

best\_epoch1 = fit.epoch[fit.history['val\_loss'].index(min(fit.history['val\_loss']))]

fig, ax = plt.subplots(2,1,figsize=(3,5))

ax[0].plot(fit.epoch,fit.history['val\_loss'],'.-',color='blue', label='validation1')

ax[0].plot(fit.epoch,fit.history['loss'],'.-',color='green', label='train1')

ax[0].set(ylabel='Loss',ylim=[0,1])

ax[0].axvspan(best\_epoch1-0.5,best\_epoch1+0.5, alpha=0.5, color='pink')

#ax[0].autoscale(False)

ax[0].scatter(best\_epoch1, evaluation[0],s=2, zorder=1,color='yellow')

ax[0].legend()

ax[1].plot(fit.epoch,fit.history['val\_accuracy'],'.-',color='green', label='validation')

ax[1].plot(fit.epoch,fit.history['accuracy'],'.-',color='red', label='train')

ax[1].set(ylabel='Accuracy',ylim=[0,1])

ax[1].axvspan(best\_epoch1-0.5,best\_epoch1+0.5, alpha=0.5, color='blue')

#ax[1].autoscale(False)

ax[1].scatter(best\_epoch1, evaluation[1],s=2, zorder=1,color='green')

ax[1].legend()

plt.show()

print("[Best epoch]:", best\_epoch1)

print("[Loss]:", min(fit.history['val\_loss']), " test:", evaluation[0])

print("[Accuracy]:", max(fit.history['val\_accuracy']), " test:", evaluation[1])

[28]

train\_size1 = int(0.6 \* DATASIZE)//SIZE\_batch

val\_size1 = int(0.2 \* DATASIZE)//SIZE\_batch

test\_size1 = int(0.2 \* DATASIZE)//SIZE\_batch

dataset1 = tf.data.Dataset.from\_tensor\_slices( (ia,oa) ).shuffle(1000).batch(SIZE\_batch)

train\_data = dataset1.take(train\_size1)

test\_data= dataset1.skip(train\_size1)

valid\_data = test\_data.skip(test\_size1)

test\_data= test\_data.take(test\_size1)

print("\n[Train data size]:",len(list(train\_data)),"\n[Valid data size]:", len(list(valid\_data)),"\n[Test data size]:", len(list(test\_data)))

[29]

model2= tf.keras.Sequential([

tf.keras.Input(shape=(len(timefeatures),1,)),

tfkl1.Conv1D(filters=conv\_filters1,kernel\_size=12, strides=2),

tfkl1.Dropout(dropout\_prob1),

tfkl1.Dense(base\_depth1,activation=activation\_func1),

tfkl1.Dense(base\_depth1,activation=activation\_func1),

tfkl1.Dense(base\_depth1,activation=activation\_func1),

tfkl1.Dropout(dropout\_prob1),

tfkl1.Dense(1,activation=tf.nn.sigmoid)

], name="heart\_failure\_model\_time")

model2.compile(optimizer=tf.optimizers.Adam(learning\_rate=0.001), loss='binary\_crossentropy',metrics=['accuracy'])

[30]

fit1 = model2.fit(train\_data, epochs=400, validation\_data=valid\_data,

batch\_size=SIZE\_batch, verbose=False,

callbacks=[tf.keras.callbacks.ReduceLROnPlateau(monitor='val\_loss', factor=0.5, patience=10, min\_lr=0.000001),

tf.keras.callbacks.EarlyStopping(monitor='val\_accuracy', min\_delta=0.0, patience=100, verbose=1, mode='auto', restore\_best\_weights=True)])

evaluation1 = model2.evaluate(test\_data)

print(len(evaluation1))

plot\_training(fit1, evaluation1)