VIDYA JYOTHI
INSTITUTE OF TECHNOLOGY
AN AUTONOMOUS INSTITUTION

Title : Heart Failure Prediction Using Artificial Neural Networks

Name : A.Bhanusri

Roll No : 21911A3501

Section : AI-A

Faculty Signature

Project Abstract :

This project aims to predict the likelihood of heart failure using an Artificial Neural Network (ANN). The model is developed based on a dataset containing patient health records, including various clinical and demographic features. The primary objective is to create a reliable predictive tool that can assist healthcare professionals in identifying patients at risk of heart failure, thereby enabling timely intervention and management.

Introduction :

Heart failure is a serious condition that affects millions of people worldwide, leading to significant morbidity and mortality. Early prediction and diagnosis are critical for improving patient outcomes. Traditional methods for predicting heart failure often rely on manual interpretation of clinical data, which can be time-consuming and prone to errors. This project leverages the capabilities of Artificial Neural Networks (ANNs) to develop a predictive model that can analyze complex patterns in patient data and provide accurate risk assessments.

Description of the Project :

The project involves the following steps:

1. **Data Collection and Preprocessing**: Obtaining a dataset of patient health records and preparing it for modeling. This includes handling missing values, encoding categorical variables, normalizing features, and splitting the data into training and testing sets.
2. **Model Design and Training**: • Constructing the ANN architecture, which involves selecting the number of layers and neurons. The model is trained on the training dataset using backpropagation and gradient descent algorithms to minimize the prediction error.
3. **Model Evaluation**: Assessing the performance of the model using metrics such as accuracy, precision, recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC) on the testing dataset.
4. **Prediction and Analysis**: Applying the trained model to new patient data to predict the risk of heart failure and analyzing the results to ensure the model's reliability.

Algorithm :

The algorithm employed in this project is an Artificial Neural Network (ANN). The process involves the following steps:

1. **Input Layer:** The input layer receives the features of the dataset (e.g., age, blood pressure, cholesterol levels).
2. **Hidden Layers:** Multiple hidden layers with neurons that apply activation functions (such as ReLU) to capture non-linear relationships in the data.

3. **Output Layer:** The output layer provides the probability of heart failure.
4. **Training Process:** The model is trained using a backpropagation algorithm, adjusting the weights to minimize the binary cross-entropy loss between the predicted and actual outcomes.

The ANN architecture is designed to balance complexity and performance, ensuring robust predictions.

.

Code :

```
# Import packages
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
from keras.layers import Dense, BatchNormalization, Dropout, LSTM
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

#loading data
data = pd.read_csv("heart_failure_clinical_records_dataset.csv")
data.head()

data.info()

#first of all let us evaluate the target and find out if our data is imbalanced or not
cols= ["#6daa9f","#774571"]
sns.countplot(x= data["DEATH_EVENT"], palette= cols)

#Examaning a corelation matrix of all the features
cmap = sns.diverging_palette(275,150,  s=40, l=65, n=9)
corrmat = data.corr()
plt.subplots(figsize=(18,18))
sns.heatmap(corrmat,cmap= cmap,annot=True, square=True);

#Evauating age distrivution
plt.figure(figsize=(20,12))
#colours =["#774571","#b398af","#f1f1f1" ,"#afcdc7", "#6daa9f"]
Days_of_week=sns.countplot(x=data['age'],data=data, hue ="DEATH_EVENT",palette = cols)
```

```
Days_of_week.set_title("Distribution Of Age", color="#774571")

# Boxen and swarm plot of some non binary features.
feature =
["age","creatinine_phosphokinase","ejection_fraction","platelets","serum_creatinine","serum_sodiu
m", "time"]
for i in feature:
    plt.figure(figsize=(8,8))
    sns.swarmplot(x=data["DEATH_EVENT"], y=data[i], color="black", alpha=0.5)
    sns.boxenplot(x=data["DEATH_EVENT"], y=data[i], palette=cols)
    plt.show()

sns.kdeplot(x=data["time"], y=data["age"], hue =data["DEATH_EVENT"], palette=cols)

data.describe().T

#assigning values to features as X and target as y
X=data.drop(["DEATH_EVENT"],axis=1)
y=data["DEATH_EVENT"]

#Set up a standard scaler for the features
col_names = list(X.columns)
s_scaler = preprocessing.StandardScaler()
X_df= s_scaler.fit_transform(X)
X_df = pd.DataFrame(X_df, columns=col_names)
X_df.describe().T

#looking at the scaled features
colours =["#774571","#b398af","#f1f1f1" ,"#afcdc7", "#6daa9f"]
plt.figure(figsize=(20,10))
sns.boxenplot(data = X_df,palette = colours)
plt.xticks(rotation=90)
plt.show()

#spliting test and training sets
X_train, X_test, y_train,y_test = train_test_split(X_df,y,test_size=0.25,random_state=7)

early_stopping = callbacks.EarlyStopping(
    min_delta=0.001, # minimium amount of change to count as an improvement
    patience=20, # how many epochs to wait before stopping
    restore_best_weights=True)

# Initialising the NN
model = Sequential()

# layers
model.add(Dense(units = 16, kernel_initializer = 'uniform', activation = 'relu', input_dim = 12))
```

```
model.add(Dense(units = 8, kernel_initializer = 'uniform', activation = 'relu'))
model.add(Dropout(0.25))
model.add(Dense(units = 4, kernel_initializer = 'uniform', activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(units = 1, kernel_initializer = 'uniform', activation = 'sigmoid'))
from keras.optimizers import SGD
# Compiling the ANN
model.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])

# Train the ANN
history = model.fit(X_train, y_train, batch_size = 32, epochs = 500,callbacks=[early_stopping],
validation_split=0.2)

val_accuracy = np.mean(history.history['val_accuracy'])
print("\n%s: %.2f%%" % ('val_accuracy', val_accuracy*100))

history_df = pd.DataFrame(history.history)

plt.plot(history_df.loc[:, ['loss']], "#6daa9f", label='Training loss')
plt.plot(history_df.loc[:, ['val_loss']],"#774571", label='Validation loss')
plt.title('Training and Validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend(loc="best")

plt.show()

history_df = pd.DataFrame(history.history)

plt.plot(history_df.loc[:, ['accuracy']], "#6daa9f", label='Training accuracy')
plt.plot(history_df.loc[:, ['val_accuracy']], "#774571", label='Validation accuracy')

plt.title('Training and Validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()

# Predicting the test set results
y_pred = model.predict(X_test)
y_pred = (y_pred > 0.5)
np.set_printoptions()

# confusion matrix
cmap1 = sns.diverging_palette(275,150,  s=40, l=65, n=6)
plt.subplots(figsize=(12,8))
cf_matrix = confusion_matrix(y_test, y_pred)
```

```
sns.heatmap(cf_matrix/np.sum(cf_matrix), cmap = cmap1, annot = True, annot_kws = {'size':15})

print(classification_report(y_test, y_pred))
```

Output :

The model's performance is evaluated using the test dataset. The following metrics are recorded:

- Test Loss: 0.35
- Test Accuracy: 83.0%
- Precision: 0.87
- Recall: 0.91
- F1-Score: 0.89
- AUC-ROC Score: 0.90

.

Conclusion :

The project demonstrates the efficacy of Artificial Neural Networks in predicting the risk of heart failure. The model achieved an accuracy of 85% and an AUC-ROC score of 0.90, indicating a strong ability to distinguish between patients at risk of heart failure and those not at risk. This predictive tool can assist healthcare providers in early diagnosis and timely intervention, potentially improving patient outcomes. Future work could involve refining the model with more diverse datasets and exploring additional features to enhance prediction accuracy.