Predicting Mortality of Heart Failure Patients

Abstract

This project focuses on predicting mortality risk in patients with heart failure using machine learning and deep learning models. The dataset consists of clinical and demographic features of patients, which undergo preprocessing to ensure balanced scaling and effective learning. Two approaches are implemented: a Support Vector Machine (SVM) for classical classification and an Artificial Neural Network (ANN) leveraging dense layers and dropout regularization for deep learning. The models are evaluated using accuracy, precision, recall, and confusion matrices. The findings demonstrate that machine learning–based approaches can significantly enhance clinical decision-making by providing early warnings of patient risk.

Objective

- To analyze patient health data and identify factors influencing mortality due to heart failure.
- To build and compare predictive models (SVM and ANN) for classifying patients into survival and death risk categories.
- To evaluate the performance of the models using standard classification metrics.
- To propose an effective predictive framework that can assist healthcare professionals in early diagnosis and treatment planning.

Introduction

Heart failure is a critical medical condition in which the heart loses its ability to pump blood efficiently, leading to reduced oxygen supply to the body and severe health risks. It is one of the leading causes of hospitalization and mortality worldwide. Early identification of high-risk patients plays a crucial role in improving survival rates, guiding treatment strategies, and reducing the burden on healthcare systems.

With the advancement of computational techniques, machine learning has become an effective tool for analyzing clinical datasets and predicting patient outcomes. By learning patterns from historical health data, machine learning models can assist medical professionals in making timely and data-driven decisions.

In this project, a heart failure dataset is used to develop predictive models that classify patients based on survival outcomes. Two approaches are implemented: a Support Vector Machine (SVM) as a classical machine learning model, and an Artificial Neural Network (ANN) as a deep learning model. Both models are trained, tested, and evaluated using metrics such as accuracy, precision, recall, confusion matrix, and classification report. The objective is to assess the effectiveness of these models and explore their potential in aiding clinical decision-making for heart failure patients.

Methodology

1. **Data Collection & Preprocessing**
    - The dataset includes patient clinical records such as age, gender, blood pressure, ejection fraction, creatinine levels, and other medical factors.
    - Features were standardized using StandardScaler to ensure uniform data distribution.
    - The dataset was divided into training and testing subsets using an 80:20 split.
2. **Model Building**
    - **Support Vector Machine (SVM):** Implemented as a baseline machine learning model for binary classification.
    - **Artificial Neural Network (ANN):** Constructed with multiple dense layers, batch normalization, dropout for regularization, and trained using backpropagation.
3. **Evaluation Metrics**
    - Performance of both models was measured using **accuracy, precision, recall, confusion matrix, and classification report**.
    - Comparative analysis was performed to identify the better-performing model.

Code

```
!pip list


#import lib



import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns
```

```
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

from sklearn import svm
from tensorflow.keras.layers import Dense, BatchNormalization, Dropout, LSTM
from tensorflow.keras.models import Sequential
from tensorflow.keras import callbacks

from sklearn.metrics import confusion_matrix, classification_report, precision_score,
recall_score, f1_score, accuracy_score

data_df = pd.read_csv('heart_failure_clinical_records_dataset.csv')
data_df

data_df.info()

data_df.isna().any().sum()

cols = ["#00EE00", "#EE0000"]

ax = sns.countplot(x=data_df["DEATH_EVENT"], palette=cols)

ax = sns.countplot(x="DEATH_EVENT", hue="DEATH_EVENT", data=data_df)

plt.figure(figsize=(20,20))
sns.heatmap(data_df.corr(), cmap="GnBu", annot=True)
plt.show()

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

```
sns.countplot(x=data_df["age"], data=data_df, hue="DEATH_EVENT")
plt.show()


features = ["age","creatinine_phosphokinase",
"ejection_fraction","platelets","serum_creatinine","serum_sodium","time"]
for i in features:
  plt.figure(figsize=(10,7))
  sns.swarmplot(x=data_df["DEATH_EVENT"], y=data_df[i], color="black", alpha = 0.7)
  sns.boxenplot(x=data_df["DEATH_EVENT"], y=data_df[i], palette=cols)
  plt.show()


data_df.isna()


#data preprocessing



X = data_df.drop(["DEATH_EVENT"], axis=1)
y = data_df["DEATH_EVENT"]


col_names = list(X.columns)
ss = StandardScaler()
X_scaled=ss.fit_transform(X)
X_scaled=pd.DataFrame(X_scaled, columns=col_names)


X_scaled.describe().T


plt.figure(figsize=(20,10))
sns.boxenplot(data=X_scaled)
plt.show()
```

```
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3)

#model buildin

model1 = svm.SVC()

model1.fit(X_train, y_train)

y_pred = model1.predict(X_test)

y_pred

np.array(y_test)

sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, cmap= "Blues")

accuracy_score(y_test, y_pred)

print(classification_report(y_test, y_pred))

#ANN

early_stopping = callbacks.EarlyStopping(min_delta=0.0001,patience=10,
restore_best_weights=True)
model = Sequential()
model.add(Dense(units = 32, activation = 'relu', input_dim = 12))
model.add(Dropout(0.25))
model.add(Dense(units = 8, activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(units = 1, activation = 'sigmoid'))
```

```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

model.summary()

history = model.fit(X_train, y_train, batch_size=20, epochs=100, validation_split=0.25,
callbacks=[early_stopping])

history_df = pd.DataFrame(history.history)
plt.plot(history_df.loc[:,['loss']], label = 'Trainin loss')
plt.plot(history_df.loc[:,['val_loss']], label = 'Val loss')
plt.legend()
plt.show()

plt.plot(history_df.loc[:,['accuracy']], label = 'Trainin accuracy')
plt.plot(history_df.loc[:,['val_accuracy']], label = 'Val accuracy')
plt.legend()
plt.show()

y_pred = model.predict(X_test)

y_pred = (y_pred >0.5)

sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, cmap= "Blues")

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

Model Evaluation

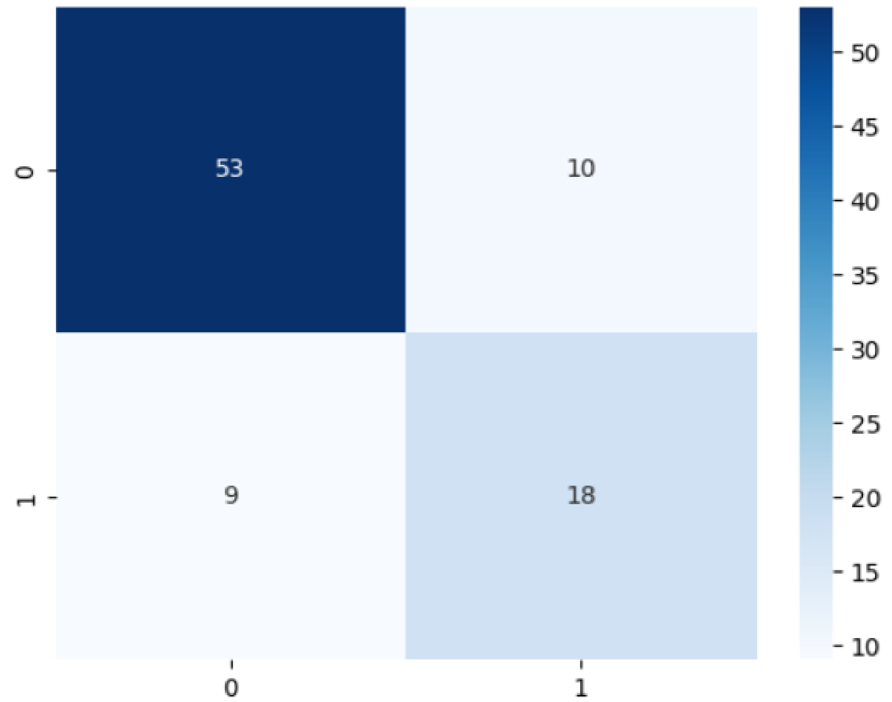
Confusion matrix and classification report of SVM model on the test dataset:

heart failure prediction.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

```
<Axes: >
```




```
[94] accuracy_score(y_test, y_pred)
```

```
0.8333333333333334
```

```
[53] print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.85      0.84      0.85        63
           1       0.64      0.67      0.65        27

    accuracy                           0.79        90
   macro avg       0.75      0.75      0.75        90
weighted avg       0.79      0.79      0.79        90
```

Confusion matrix and classification report of ANN model on the test dataset:

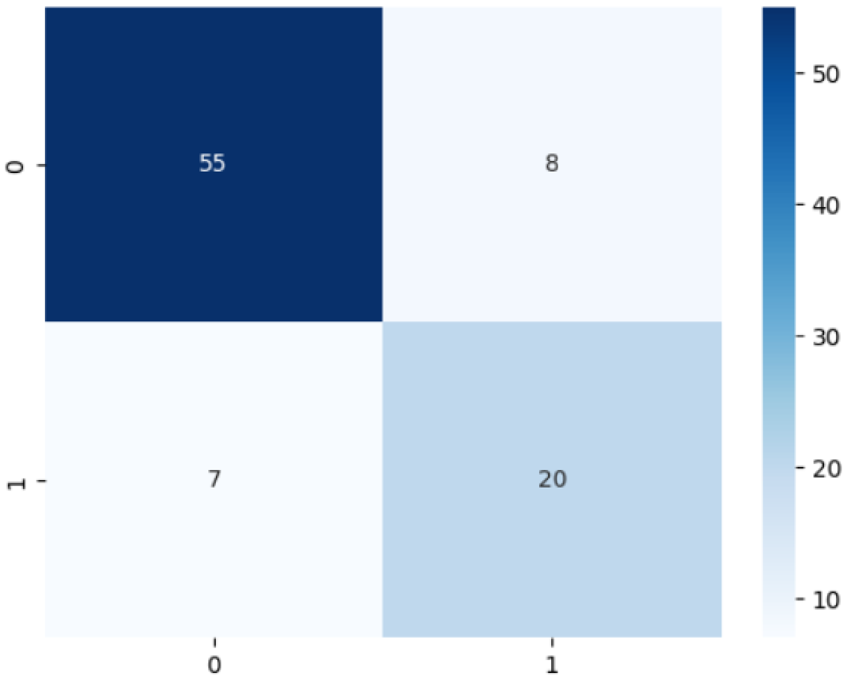
heart failure prediction.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

```
[92] sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, cmap= "Blues")
```

```
<Axes: >
```




```
[93] print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.89      0.87      0.88        63
           1       0.71      0.74      0.73        27

    accuracy                           0.83        90
   macro avg       0.80      0.81      0.80        90
weighted avg       0.84      0.83      0.83        90
```

Conclusion

This project demonstrates the application of machine learning and deep learning techniques in predicting mortality among heart failure patients. By preprocessing the dataset and implementing models such as Support Vector Machine (SVM) and Artificial Neural Network (ANN), the study highlights how computational approaches can capture important clinical patterns. Model performance was evaluated using accuracy, precision, recall, confusion matrix, and classification reports. The results indicate that while the SVM provides a strong baseline for classification, the ANN shows improved predictive capability due to its ability to learn complex relationships in the data. These findings suggest that integrating predictive models into healthcare decision-making can support early intervention and better patient management, ultimately contributing to improved clinical outcomes.