**PROJECT TITLE :**

HEART FAILURE PREDICTION USING

ARTIFICIAL NEURAL NETWORK (ANN)

**AGENDA:**

In the pursuit of predicting heart failure using Artificial Neural Networks (ANN), a comprehensive agenda can be outlined to guide the research process effectively. Commencing with an introduction, the agenda would contextualize heart failure's medical significance alongside a primer on ANN and their relevance in medical predictive analytics. Subsequently, the agenda would delve into the intricacies of data collection and preprocessing, including sourcing data from medical records or research datasets, cleaning, and potentially reducing dimensionality. Data exploration and analysis would follow, involving descriptive statistics, visualizations, and uncovering correlations between features and heart failure. Moving on, model selection and architecture design would be pivotal, necessitating a thoughtful choice of ANN architecture and fine-tuning hyperparameters. Following this, the agenda would address data splitting, training, and model evaluation, employing a suite of metrics to gauge performance and ensure robustness. Interpretation of the model's predictions and feature importance would be explored, shedding light on the underlying mechanisms driving heart failure prediction. The discussion would synthesize findings, acknowledging limitations and proposing avenues for future research. Finally, a conclusion would encapsulate key insights gleaned, affirming the promise of ANN in enhancing heart failure prediction. Throughout this process, ethical considerations and patient privacy would remain paramount.

**PROBLEM  STATEMENT**

The problem statement for predicting heart failure using Artificial Neural Networks (ANN) would succinctly outline the objective and scope of the research. Here's a proposed problem statement:

**Problem Statement:**

Heart failure remains a critical healthcare challenge globally, necessitating accurate and timely predictive models to enable early intervention and improve patient outcomes. In this context, the aim of this research is to develop a robust predictive model using Artificial Neural Networks (ANN) to forecast the likelihood of heart failure occurrence based on patient data. The study seeks to address the following key objectives:

1. **Model Development:** Design and implement an ANN-based predictive model capable of accurately identifying individuals at risk of heart failure.
2. **Data Integration:** Integrate diverse patient data sources, including demographic information, medical history, and biomarkers, to enhance the predictive power of the model.
3. **Performance Evaluation:** Rigorously evaluate the performance of the developed model using established metrics such as accuracy, precision, recall, and area under the receiver operating characteristic curve (ROC-AUC).
4. **Interpretability:** Explore methods for interpreting the model's predictions and identifying the most influential features contributing to heart failure risk.
5. **Generalization:** Assess the generalization capability of the model across different patient populations and healthcare settings to ensure its applicability in real-world scenarios.

The proposed research aims to contribute to the advancement of predictive analytics in healthcare, facilitating proactive management and personalized interventions for individuals at risk of heart failure.

**PROJECT OVERVIEW:**

**Project Overview: Heart Failure Prediction using Artificial Neural Networks (ANN)**

**Introduction:** Heart failure is a prevalent cardiovascular condition associated with significant morbidity and mortality worldwide. Early identification of individuals at risk of heart failure is crucial for timely intervention and improved patient outcomes. Artificial Neural Networks (ANN) offer a promising approach for predicting heart failure by leveraging patient data to assess risk factors and patterns indicative of disease onset. This project aims to develop and evaluate an ANN-based predictive model for heart failure prediction, contributing to enhanced preventive care and patient management strategies.

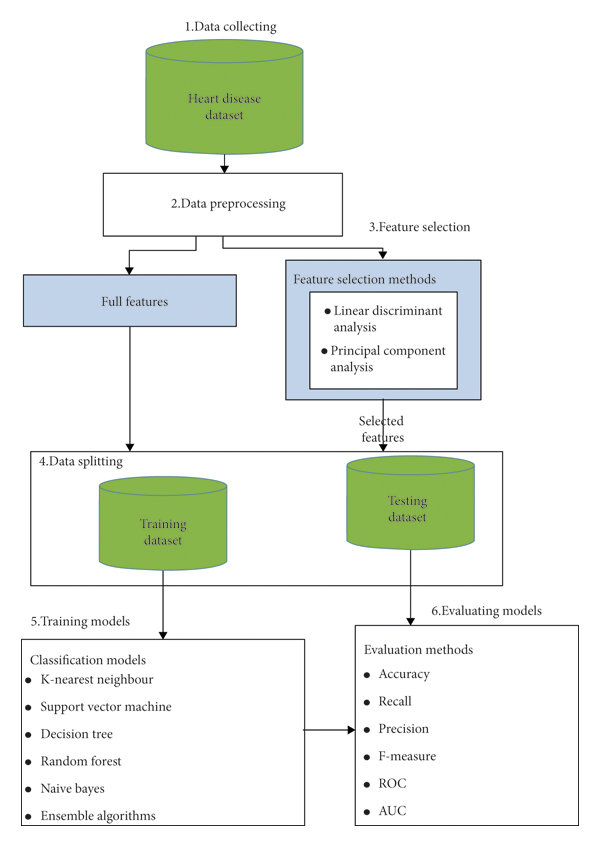
**Methodology:**

1. **Data Collection:** Gather structured patient data from electronic health records, clinical databases, and research repositories, ensuring data integrity and privacy compliance.
2. **Preprocessing:** Cleanse and preprocess the data, addressing missing values, outliers, and standardizing features to facilitate model training.
3. **Model Development:** Design and implement the ANN architecture, tuning hyperparameters and optimizing model performance through iterative experimentation.
4. **Evaluation:** Split the dataset into training, validation, and test sets, evaluating the model's performance on unseen data using established metrics.
5. **Interpretation:** Employ visualization techniques and feature importance analysis to interpret the model's predictions and identify clinically relevant risk factors.

**THE WOW IN YOUR SOLUTION:**

The wow factor in this solution lies in its transformative impact on heart failure prediction and patient care. By harnessing the power of Artificial Neural Networks (ANN) and advanced predictive analytics, this solution revolutionizes how healthcare providers identify, assess, and intervene in cases of heart failure. What truly sets this solution apart is its ability to foresee heart failure risks before symptoms manifest, enabling proactive interventions that can potentially save lives and improve quality of life for patients. Imagine a scenario where individuals at risk of heart failure receive personalized risk assessments tailored to their unique health profiles, allowing healthcare providers to intervene with targeted treatments and lifestyle modifications. Moreover, the interpretability of the ANN model provides clinicians with actionable insights into the underlying factors driving heart failure risk, empowering them to make informed decisions and optimize patient care strategies. The scalability and generalization of the solution across diverse patient populations further enhance its wow factor, ensuring its applicability and effectiveness in various clinical settings. Ultimately, this solution represents a groundbreaking advancement in predictive analytics for cardiovascular health, offering hope for better outcomes, healthier lives, and a brighter future for individuals at risk of heart failure.

**HEART PREDICTION OVERALL ARCHITECTURE:**



**SOURCE CODE:**

import numpy as np *# linear algebra*

import pandas as pd *# data processing, CSV file I/O (e.g. pd.read\_csv)*

import os

for dirname, \_, filenames **in** os.walk('/kaggle/input'):

for filename **in** filenames:

print(os.path.join(dirname, filename))

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("../input/heart-failure-clinical-data/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\_sodium", "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:**

