In [1]:
```
import tensorflow as tf
from tensorflow import keras
```

In [2]:
```
print("Tensorflow version: ",tf.__version__)
print("Keras version: ",keras.__version__)
```

```
Tensorflow version:  2.18.0
Keras version:  3.6.0
```

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df = pd.read_csv("wine.csv")
df.head()
```

Out[3]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56

In [5]:
```
df.shape
```

Out[5]:
```
(1599, 12)
```

In [6]:
```
print("Contain Null: ",df.isnull().sum())
```

```
Contain Null:  fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   object 
dtypes: float64(11), object(1)
memory usage: 150.0+ KB
```

In [8]:
```
df.describe()
```

Out[8]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	1
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	

In [10]:
```
categorical_columns = df.select_dtypes(include=['object']).columns
print("Categorical variables:", categorical_columns)
```

```
Categorical variables: Index(['quality'], dtype='object')
```

In [12]:
```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
df['quality'] = encoder.fit_transform(df['quality'])
```

In [14]:
```
x = df.drop(columns=['quality'])
y = df['quality']
```

In [18]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state=
x_train,x_val,y_train,y_val = train_test_split(x_train,y_train,test_size=0.2,ran
```

In [19]:
```
print(f"Train size: {x_train.shape}, Validation size: {x_val.shape}, test size:
```

```
Train size: (895, 11), Validation size: (224, 11), test size: (480, 11)
```

In [20]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
x_train_scaled = scaler.fit_transform(x_train)
x_val_scaled = scaler.fit_transform(x_val)
x_test_scaled = scaler.fit_transform(x_test)
```

In [27]:
```
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense

hidden1 = Dense(64,activation='relu',input_shape = (x_train_scaled.shape[1],))
hidden2 = Dense(32,activation='relu')
output = Dense(1,activation='sigmoid')
```

```
c:\Users\tarpi\AppData\Local\Programs\Python\Python312\Lib\site-packages\keras\sr
c\layers\core\dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim`
argument to a layer. When using Sequential models, prefer using an `Input(shape)`
object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

In [42]:
```
#model = Sequential([hidden1,hidden2,output])

from tensorflow.keras.layers import Dropout
from tensorflow.keras.regularizers import l2

model = Sequential([
    Dense(64, activation='relu', input_shape=(x_train_scaled.shape[1],), kernel_
    Dropout(0.5),
    Dense(32, activation='relu', kernel_regularizer=l2(0.01)),
    Dropout(0.5),
    Dense(1, activation='sigmoid')
])
```

```
c:\Users\tarpi\AppData\Local\Programs\Python\Python312\Lib\site-packages\keras\sr
c\layers\core\dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim`
argument to a layer. When using Sequential models, prefer using an `Input(shape)`
object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

In [43]:
```
model.summary()
```

Model: "sequential_2"

Layer (type)	Output Shape	Param #
dense_9 (Dense)	(None, 64)	768
dropout (Dropout)	(None, 64)	0
dense_10 (Dense)	(None, 32)	2,080
dropout_1 (Dropout)	(None, 32)	0
dense_11 (Dense)	(None, 1)	33

Total params: 2,881 (11.25 KB)

Trainable params: 2,881 (11.25 KB)

Non-trainable params: 0 (0.00 B)

In [44]:
```
model.compile(loss='binary_crossentropy',
              optimizer= 'adam',
              metrics=['accuracy'])
```

In [51]:
```
# history = model.fit(x_train_scaled,y_train,validation_data=(x_val_scaled,y_val

from tensorflow.keras.callbacks import EarlyStopping
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weig
history = model.fit(
    X_train_scaled, y_train,
    validation_data=(X_val_scaled, y_val),
    epochs=50,
    batch_size=32,
    callbacks=[early_stopping]
)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[51], line 6
      3 from tensorflow.keras.callbacks import EarlyStopping
      4 early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_be
st_weights=True)
      5 history = model.fit(
----> 6     X_train_scaled, y_train,
      7     validation_data=(X_val_scaled, y_val),
      8     epochs=50,
      9     batch_size=32,
     10     callbacks=[early_stopping]
     11 )

NameError: name 'X_train_scaled' is not defined
```

In [52]:
```
from tensorflow.keras.callbacks import ReduceLROnPlateau
lr_scheduler = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=3, min
```

In [53]:
```
weights,biases = model.layers[0].get_weights()
print("Weights: ",weights)
print("Biases: ",biases)
```

```
Weights:  [[ 5.16635412e-03  7.49213025e-02  1.80279315e-02 -1.12767164e-02
   5.58731053e-03 -3.07337323e-04  1.34011758e-02  5.68314120e-02
   2.95740664e-02 -1.95619445e-02 -1.75791595e-03 -4.11349647e-02
   1.38315922e-02  2.41307020e-02  3.23316045e-02  1.45635586e-02
  -1.98109038e-02 -4.26980443e-02 -4.97252941e-02 -1.27053098e-03
   1.25936652e-03  3.43547314e-02 -5.23776514e-03 -1.71231925e-02
   9.40294936e-02  1.88970137e-02  3.53873298e-02  4.26217215e-03
  -4.74165678e-02 -1.08114462e-02  2.26835590e-02 -3.12178712e-02
   8.05771723e-02  1.18560146e-03  3.31630814e-03  6.43332442e-03
  -4.85900454e-02 -3.22167352e-02 -2.80781724e-02  3.56128551e-02
   5.59836999e-03 -1.22960201e-02  3.60772852e-03 -5.97596020e-02
   2.92488001e-02 -5.97728824e-04  2.84725185e-02  1.08349966e-02
  -5.15822470e-02  2.43353061e-02  1.78451054e-02 -1.02960346e-02
  -5.69564253e-02  6.90129325e-02 -4.31588851e-02 -1.88686028e-02
  -2.80935820e-02 -1.35076959e-02  8.59933421e-02 -1.95212383e-02
   1.25443591e-02  2.23892909e-02  4.56159413e-02 -9.17572901e-03]
 [ 1.18535627e-02 -5.96555658e-02 -3.92260216e-02  1.05116792e-01
  -4.75710407e-02 -9.06451233e-03 -2.27503814e-02 -8.01783651e-02
   7.53343403e-02 -4.75370834e-05  1.61745679e-02  4.52795029e-02
  -1.49511714e-02  2.36595497e-02  1.06932916e-01  9.26397890e-02
   9.25138686e-03  4.22055423e-02  6.21627308e-02  8.55766609e-02
   4.40517515e-02  1.66063942e-02  9.78528615e-03  1.71584133e-02
  -5.27949594e-02 -7.34149590e-02 -5.61680645e-02 -1.18849427e-01
   6.43080175e-02  8.75435546e-02  1.39407143e-01  6.53173849e-02
  -5.59898615e-02 -1.02405250e-03  3.06048244e-03 -3.61557566e-02
   2.48456970e-02  5.91557510e-02  1.18123945e-02 -7.55344927e-02
  -3.39160077e-02 -6.28185868e-02 -1.45175308e-02  1.39955178e-01
  -5.66966273e-02  6.30121753e-02 -6.08407222e-02  7.27792270e-04
   3.95073788e-03 -4.85852249e-02  1.01716006e-02 -7.71117359e-02
   8.42060670e-02 -5.98011166e-02  7.74988309e-02  1.59908086e-02
  -2.48444956e-02  1.31377298e-02 -4.42154072e-02  3.26757529e-03
  -6.35785237e-02 -4.80168611e-02 -8.22592229e-02  1.26781203e-02]
 [ 2.46592313e-02  2.54248828e-02  3.52259912e-02 -2.13926862e-04
   7.82851654e-04  6.69452595e-03  2.83603165e-02 -5.70109636e-02
   1.91321187e-02  1.08461101e-02  6.02329103e-03  4.58476803e-04
   2.74612401e-02  1.35399446e-01  8.05680361e-03  4.08983827e-02
  -8.92243907e-03  7.65547007e-02  4.11324911e-02  1.47923799e-02
   7.38755707e-03  1.59539934e-02  1.15490044e-02  4.61495034e-02
  -5.11809513e-02  3.06982715e-02  7.41478428e-03 -1.15999421e-02
   4.32895683e-02  4.78596520e-03  8.01937953e-02  3.65008041e-02
   2.51928661e-02 -2.75097461e-03  3.00995423e-03 -2.63399687e-02
   6.58893660e-02  1.49929831e-02 -1.95787326e-02 -7.15672374e-02
  -1.56192798e-02  3.98543626e-02 -1.36473598e-02  8.01265910e-02
  -1.19837662e-02  3.66784893e-02 -2.55796034e-02 -9.39310621e-03
   1.65351219e-02  4.74612750e-02  2.87794834e-03  3.52594964e-02
   1.68186612e-02  2.35988467e-04  2.69991718e-02  8.72353930e-03
  -7.44688092e-03  7.26713985e-02  1.38574755e-02  3.77319865e-02
   8.82961671e-04 -1.27522564e-02 -3.23356800e-02 -2.92794756e-03]
 [ 2.20218524e-02 -3.09704579e-02 -9.76935495e-03 -1.42055349e-02
  -2.29634922e-02 -1.97456265e-03 -1.43416086e-02 -7.41210356e-02
   3.53235548e-04  1.98339466e-02  1.91185623e-03  1.08404597e-02
  -4.81344899e-03  1.88393146e-02 -2.20028479e-02  4.06564698e-02
   1.45543600e-03 -1.68878287e-02  3.21845561e-02  5.77240735e-02
   4.15793294e-03  9.19019710e-03 -2.95748864e-03 -1.12673792e-03
  -2.19796617e-02 -2.96799373e-02 -2.09637242e-03 -5.87354675e-02
  -1.29152099e-02 -3.26629020e-02 -1.63531508e-02 -4.58202288e-02
  -2.92089395e-02 -1.96026871e-04 -1.19848340e-03 -1.95166729e-02
  -1.33124413e-02 -1.38627589e-02 -8.73834826e-03 -6.23907708e-02
   5.29351365e-03 -3.63318957e-02  1.01333186e-02 -5.49104251e-02
  -5.47340959e-02  2.32909061e-03 -2.13248003e-02 -2.00676452e-03
```

```
  1.60969608e-03 -2.10455209e-02 -6.13372587e-03 -7.73959905e-02
 -3.46956030e-02 -4.48629782e-02 -1.62216816e-02 -1.92035716e-02
  1.60588063e-02  3.12620285e-03 -1.22796381e-02  5.03905583e-03
 -3.46673653e-03  1.91695690e-02 -5.39496494e-03 -4.69652237e-03]
[-1.49117438e-02 -7.60521293e-02 -2.09727827e-02  1.48863653e-02
 -2.37033982e-02 -5.85586671e-03 -7.74869276e-03 -6.07759953e-02
 -6.28397390e-02  1.36820814e-02  2.91958973e-02 -1.01308012e-03
 -9.53172706e-03  2.99757980e-02  4.50810753e-02 -1.81368347e-02
 -1.48209054e-02 -6.68545067e-02 -4.02341001e-02  5.87928705e-02
 -5.09758526e-03  3.18752080e-02  2.60586813e-02  3.78156155e-02
 -4.19640318e-02 -3.45935374e-02 -4.83628623e-02 -5.67787662e-02
 -4.16801218e-03 -5.22475457e-03  8.20813552e-02  1.65696647e-02
 -4.15335372e-02 -5.48518947e-05  1.73656847e-02 -1.63750071e-02
 -7.06245378e-02  2.99566449e-03  3.02743036e-02 -1.60288494e-02
  1.55217918e-02 -2.49047782e-02 -2.75501912e-03  9.42505151e-03
 -1.56277493e-02 -5.15218824e-02  5.79091115e-03 -4.29712981e-03
 -2.49581300e-02 -2.97113080e-02  5.18324748e-02 -6.21021762e-02
 -1.10233156e-02 -4.49900366e-02  3.34809087e-02  1.07588414e-02
 -1.27689084e-02 -4.39068526e-02 -6.02925234e-02  2.33564731e-02
 -2.04817653e-02 -1.80389956e-02 -4.50447723e-02  1.30494805e-02]
[-3.74135785e-02  6.14085384e-02 -6.11027004e-03  8.01897049e-02
  2.09884569e-02  6.85053458e-03 -1.93769275e-03  4.63620536e-02
 -8.80447626e-02 -6.82853395e-03  3.05244382e-02  3.41498926e-02
 -7.63144717e-03  3.57349627e-02  5.05038723e-02  2.69608852e-02
 -2.38798708e-02 -4.80153225e-02 -6.03087582e-02 -1.80991851e-02
 -2.60261334e-02  2.81603336e-02 -1.56170060e-03  5.95235564e-02
  4.69523668e-02  2.42885835e-02  2.28357539e-02  5.70304468e-02
  5.60064241e-02  7.95620531e-02  1.52103016e-02  5.86116724e-02
  6.43964335e-02  2.23989878e-03  1.50762042e-02  1.20107019e-02
 -1.25186369e-01  7.52125829e-02  8.03228319e-02  2.74788383e-02
 -1.62378177e-02  2.44445503e-02 -6.96700392e-03 -5.49649522e-02
  2.51497291e-02 -2.09502596e-02  2.86973622e-02  7.80991511e-03
  1.77719686e-02 -1.01760402e-02  7.86908269e-02  6.44559488e-02
 -3.10608782e-02  7.66506046e-02  1.02853253e-02  5.42925596e-02
  8.19076896e-02 -6.71006590e-02  4.03597429e-02 -2.64063533e-02
 -3.52313300e-03  3.05982530e-02  2.27973610e-02  3.12729739e-02]
[-1.35324970e-02 -4.44524139e-02  9.47273616e-03  3.78991701e-02
 -2.25047413e-02 -4.30171192e-03  1.31817227e-02 -3.66016924e-02
  3.19766626e-02  1.36846229e-02  4.39971983e-02  9.38714296e-02
  2.23588361e-03 -6.11186288e-02 -3.47588764e-04  2.38135718e-02
 -1.10088242e-02 -1.12742197e-03  5.58510162e-02  5.62865362e-02
 -1.20276138e-02  8.76123756e-02  2.44071223e-02  4.60056812e-02
 -5.24489135e-02 -3.87948342e-02 -6.67676777e-02 -2.00112779e-02
  1.12016037e-01  2.99321581e-02  5.11723906e-02  8.59567970e-02
 -2.24137772e-02  1.41100038e-03  1.27938464e-02 -3.35280709e-02
  5.39917313e-02  5.73613979e-02  6.89288750e-02 -6.74812272e-02
 -3.60506400e-02 -7.55466614e-03 -2.35759988e-02  8.34840387e-02
 -3.07863075e-02 -8.26663710e-03 -3.42021324e-02  8.15967377e-03
  8.85458216e-02 -4.22136225e-02  4.75559086e-02 -2.09296811e-02
  5.30123711e-02 -3.25874612e-02  1.17914893e-01  1.53891891e-01
 -4.67169248e-02 -1.10841393e-02 -2.59698480e-02  4.43110168e-02
 -7.92637095e-02 -8.42895582e-02 -6.14605434e-02  2.83464491e-02]
[-6.77600829e-03 -1.47768363e-01 -3.64898629e-02 -7.10493559e-03
 -4.86509092e-02 -2.05599256e-02 -2.39693262e-02 -5.77977439e-03
 -1.13542043e-02 -6.45643333e-04  1.26484679e-02  1.41345765e-02
  2.74306769e-03 -4.17519957e-02  1.39620975e-02  2.96586799e-03
 -1.55878644e-02  1.34403631e-02  1.23155043e-02  3.98375187e-03
 -6.56198757e-03  1.14666037e-02  4.22651786e-03  1.26683824e-02
 -3.74361463e-02 -1.01228774e-01 -1.46158144e-01 -8.55681971e-02
  3.84487733e-02  2.44735666e-02  6.76279329e-03 -1.87138654e-03
```

```
  -8.60079080e-02  4.20560653e-04  1.04363244e-02  7.30377948e-03
   1.91694926e-02  2.46381722e-02 -5.13077248e-03  3.60622481e-02
   1.34973181e-03 -1.14996389e-01 -2.95178555e-02  5.52113494e-03
  -1.54014053e-02 -2.62442660e-02 -1.39178447e-02 -3.32720391e-02
   4.02536383e-03 -8.23020265e-02  1.42841451e-02 -6.31630793e-02
   2.90858801e-02 -1.01605132e-01 -2.26780698e-02  1.41453352e-02
  -9.15881693e-02 -1.12544699e-02 -8.60828832e-02 -3.08421068e-03
  -9.68459586e-04 -1.04250260e-01 -1.28939539e-01  1.29838651e-02]
 [ 1.55844735e-02 -8.75736326e-02 -1.61077250e-02 -6.26069456e-02
  -2.66150311e-02 -9.79938917e-03 -2.23055594e-02  2.14200914e-02
   5.10814637e-02  1.73947625e-02 -4.66286344e-03  6.60311570e-03
  -1.88323471e-03  4.61704843e-02 -4.63869683e-02 -5.62845618e-02
  -5.29760122e-03 -5.17871277e-03  2.81843869e-03 -2.71400306e-02
   1.34571325e-02  1.43917399e-02  3.43112797e-02 -8.47566465e-04
   6.33192062e-02 -6.42636493e-02 -1.97632052e-02 -8.38829130e-02
  -2.45570242e-02 -9.82472301e-02  8.84340890e-03 -1.18389893e-02
  -1.61599163e-02  4.35731234e-03 -3.51074967e-03  1.22807268e-02
   6.66288733e-02 -2.50940267e-02 -1.50969829e-02  1.41042937e-02
   5.95356338e-03 -4.33110856e-02  2.27274420e-03  2.13538297e-02
   2.06327029e-02  2.27182801e-03  1.62758380e-02  1.20041356e-03
  -9.13288072e-03  1.15876021e-02 -2.13763844e-02 -3.85361724e-02
  -1.97834559e-02  3.60488705e-02 -1.46805253e-02  3.13835032e-02
  -6.13329411e-02  6.65304139e-02 -6.42165542e-02  4.80922014e-02
   1.47818234e-02 -1.37999840e-02 -3.84277627e-02  6.83547696e-03]
 [-6.64904341e-02  7.57396519e-02  1.72678977e-02  2.31996756e-02
   3.61831784e-02  5.60282776e-03  1.86306741e-02  9.31516886e-02
  -1.25065178e-01 -9.04491730e-03  4.07328196e-02  7.75145739e-02
   3.12994421e-03 -1.61363631e-01 -7.39937201e-02  1.09231407e-02
  -3.10564507e-02 -2.02229470e-01 -1.10750481e-01 -1.09498851e-01
  -8.03627893e-02  5.75902797e-02 -2.05059145e-02  7.38279819e-02
   9.85343680e-02  4.59094159e-02  1.07975230e-01  9.35153738e-02
   7.13290423e-02  5.02621979e-02 -1.05130300e-01 -2.47020321e-03
   4.65821922e-02  2.88712233e-03  1.44322924e-02  4.20200787e-02
  -1.94423646e-01  5.83231822e-02  5.25320731e-02  8.82996097e-02
   3.94620150e-02  3.14157531e-02  2.73431316e-02 -1.51764095e-01
   7.28559792e-02 -1.67877391e-01  6.55340627e-02  1.08564394e-02
  -5.28159626e-02  6.25112280e-02  2.01959852e-02  6.34007305e-02
  -7.75713921e-02  7.22106919e-02  4.42420878e-03  5.02242707e-02
   2.28603408e-02 -1.72846273e-01  5.08585088e-02 -6.24750331e-02
   7.83215240e-02  3.27628143e-02  9.38492194e-02  2.89575234e-02]
 [-4.38618055e-03  1.96476221e-01  3.61643173e-02 -7.34404996e-02
   1.40457809e-01  2.56982706e-02  3.53703983e-02  1.32272765e-01
  -2.47702803e-02  4.50445525e-03 -2.07647737e-02 -1.93889905e-02
   1.45374518e-02 -1.05931945e-01 -4.86775227e-02 -5.17215803e-02
   1.76019836e-02 -7.40573555e-02 -6.32324442e-02 -7.68880472e-02
   4.72307438e-03  7.34326104e-03  3.86971980e-03 -4.49103154e-02
   3.81691381e-02  1.51537642e-01  1.76031455e-01  1.48658052e-01
  -9.19750035e-02 -8.27411786e-02 -7.78435394e-02 -1.23942152e-01
   1.69758081e-01 -7.65408855e-04 -1.00421226e-02  4.15584669e-02
  -2.82985028e-02 -9.83653069e-02 -7.77998641e-02  1.08907379e-01
   2.74054874e-02  1.24947041e-01  1.98699571e-02 -1.08151846e-01
   3.75455245e-02 -4.52877097e-02  4.56204526e-02  4.73019369e-02
  -8.29950050e-02  1.61511332e-01 -2.82358639e-02  1.78811744e-01
  -8.40233937e-02  1.61005124e-01 -1.15738206e-01 -9.37258005e-02
   1.42095298e-01 -6.56785769e-03  1.57885879e-01 -6.41423883e-03
   1.79372728e-02  1.55503184e-01  1.74292311e-01 -2.28238553e-02]]
Biases:  [-8.2836309e-03  6.8993926e-02 -1.3974565e-02  8.1053274e-03
  6.9938146e-02  5.7150158e-03 -2.8981099e-02  1.5524106e-01
  6.8768970e-04 -3.5031843e-03 -4.9810257e-02 -6.3235790e-02
  4.2389785e-03  4.4642750e-02  1.7113268e-02  6.7936070e-02
```

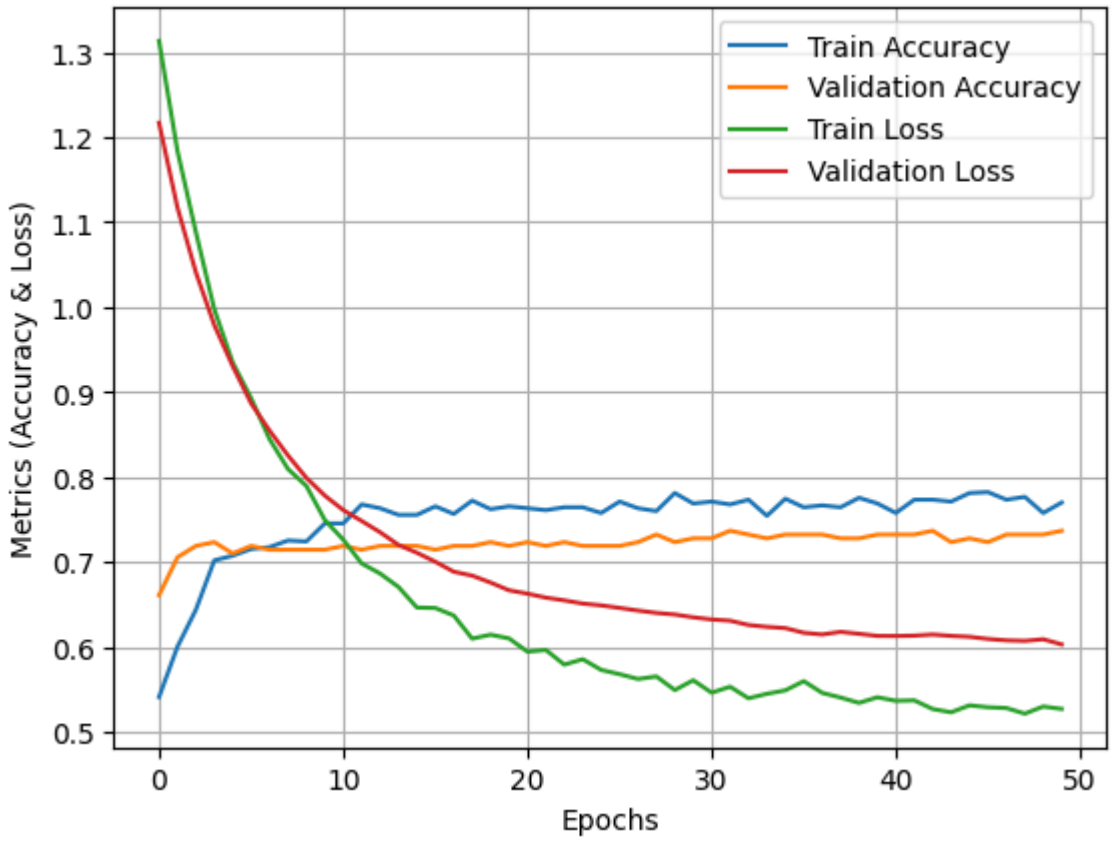
```
-4.2076509e-02  2.2531871e-02  2.8388117e-02  6.5327495e-02
 1.9566822e-03 -1.9919703e-02 -1.1803136e-02 -4.4720884e-02
 1.0195395e-01  4.6971511e-02  5.0547071e-02  7.6928720e-02
-4.4525437e-02  1.5578771e-02  7.5428791e-02  5.7468273e-02
 5.7145156e-02 -5.5584352e-02 -1.4883115e-02  9.3691133e-02
 2.2411814e-02 -4.8214313e-02 -8.1701418e-03  1.4547954e-01
 2.6513608e-02  4.2562246e-02  2.2281120e-02  8.4866015e-03
 9.5946200e-02  2.3957158e-02  9.2157252e-02 -7.7043820e-05
 7.1254350e-02  5.4263365e-02 -7.3383830e-02  1.1454501e-01
 3.4916379e-02  1.3769649e-01  7.9432130e-02  3.2861877e-02
 5.8214284e-02  1.8147571e-02  5.9700094e-02  1.1882979e-02
 1.0293108e-01  8.7663785e-02  3.2110479e-02 -2.1403797e-02]
```

In [54]:
```
history_df = pd.DataFrame(history.history)
print(history_df.head())
```

```
   accuracy      loss  val_accuracy  val_loss
0  0.540782  1.313497      0.660714  1.216986
1  0.600000  1.183901      0.705357  1.117237
2  0.643575  1.088298      0.718750  1.041002
3  0.701676  0.997498      0.723214  0.978810
4  0.707263  0.934622      0.709821  0.930460
```

In [55]:
```
import matplotlib.pyplot as plt

# Plot accuracy
plt.plot(history_df['accuracy'], label='Train Accuracy')
plt.plot(history_df['val_accuracy'], label='Validation Accuracy')
plt.plot(history_df['loss'], label='Train Loss')
plt.plot(history_df['val_loss'], label='Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Metrics (Accuracy & Loss)')
plt.legend(loc='best')
plt.grid()
plt.show()
```



In [56]:
```
from sklearn.metrics import classification_report

test_loss,test_accuracy = model.evaluate(x_test_scaled,y_test,verbose=0)
print(f"Test Loss: {test_loss}, Test Accuracy: {test_accuracy}")

y_pred = (model.predict(x_test_scaled)>0.5).astype("int32")
print(classification_report(y_test, y_pred))
```

```
Test Loss: 0.5666950941085815, Test Accuracy: 0.7083333134651184
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 971us/step
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 971us/step
              precision    recall  f1-score   support

           0       0.65      0.76      0.70       213
           1       0.78      0.67      0.72       267

    accuracy                           0.71       480
   macro avg       0.71      0.71      0.71       480
weighted avg       0.72      0.71      0.71       480
```

In [ ]:

In [ ]: