

**AIDS CLASSIFICATION USING ARTIFICIAL NEURAL NETWORKS AND DEEP LEARNING.**

NAME:- VINAYCHOWHAN NAIK LAVUDYA

ROLL NO:- **21911A3596**

SECTION:- **AI-B**

**FACULTY SIGN**

**ABSTRUCT**

Acquired Immunodeficiency Syndrome (AIDS) remains a significant global health challenge, necessitating advanced diagnostic and classification techniques to enhance patient management and treatment outcomes. Traditional methods of AIDS diagnosis and classification often rely on clinical and laboratory data, which can be time-consuming and resource-intensive. In recent years, the application of neural networks and deep learning has shown promising potential in automating and improving the accuracy of medical diagnoses, including AIDS classification.

**INTRODUCTION**

Acquired Immunodeficiency Syndrome (AIDS) continues to pose a significant global health challenge, affecting millions of individuals worldwide. It is caused by the Human Immunodeficiency Virus (HIV), which compromises the immune system, leading to life-threatening infections and cancers.

Recent advancements in artificial intelligence (AI) and machine learning (ML), particularly in neural networks and deep learning, offer transformative potential for medical diagnostics. Neural networks, inspired by the human brain's structure, can learn complex patterns and relationships within data, making them particularly suited for medical applications.

This study aims to explore the application of neural networks and deep learning in the classification of AIDS. By developing and evaluating various deep learning models, we seek to enhance diagnostic capabilities and provide a foundation for integrating these technologies into clinical practice. The objectives include constructing a comprehensive dataset, designing and implementing different neural network architectures, and rigorously evaluating their performance against established metrics.

DESCRIPTION

Acquired Immunodeficiency Syndrome (AIDS) results from the progression of the Human Immunodeficiency Virus (HIV) infection, which attacks and weakens the immune system. As the disease progresses, the body's ability to combat infections and certain cancers diminishes, leading to severe health complications. Accurate and early diagnosis of AIDS is paramount to managing the disease and improving patient outcomes.

The Role of Neural Networks and Deep Learning:

Neural networks are computational models inspired by the human brain's neural architecture. They consist of layers of interconnected nodes (neurons) that can learn to recognize patterns in data through training. Deep learning, a subset of neural networks, involves multiple hidden layers that enable the automatic extraction and abstraction of features from raw data. This makes deep learning particularly effective in complex tasks such as image and signal processing, natural language understanding, and, notably, medical diagnostics.

Methodology:

1**.Data Collection**: Gather a robust dataset from clinical databases, including electronic health records (EHRs), imaging repositories, and laboratory test results.

2.**Data Preprocessing**: Normalize data, handle missing values, and apply data augmentation techniques to improve model generalization.

3.**Model Implementation**: Develop various neural network architectures.

4.**Training**: Use a training dataset to teach the models the patterns associated with AIDS, adjusting weights through backpropagation to minimize error.

5.**Validation**: Apply cross-validation techniques to evaluate model performance and prevent overfitting. Test models on an independent dataset to assess real-world applicability.

Code

import pandas as pd

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

from sklearn.preprocessing import StandardScaler

import joblib

# Load the dataset

data = pd.read\_csv('/content/AIDS\_Classification.csv')

# Separate features and target

X = data.drop('z30', axis=1)  # Replace 'target' with your target column name

y = data['z30']

# Split the data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Standardize the data

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Save preprocessed data

pd.DataFrame(X\_train).to\_csv('X\_train.csv', index=False)

pd.DataFrame(X\_test).to\_csv('X\_test.csv', index=False)

pd.DataFrame(y\_train).to\_csv('y\_train.csv', index=False)

pd.DataFrame(y\_test).to\_csv('y\_test.csv', index=False)

# Save the scaler

joblib.dump(scaler, 'scaler.pkl')

import pandas as pd

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Dropout

# Load preprocessed data

X\_train = pd.read\_csv('X\_train.csv').values

y\_train = pd.read\_csv('y\_train.csv').values

X\_test = pd.read\_csv('X\_test.csv').values

y\_test = pd.read\_csv('y\_test.csv').values

# Define the model

model = Sequential([

    Dense(64, activation='relu', input\_shape=(X\_train.shape[1],)),

    Dropout(0.5),

    Dense(32, activation='relu'),

    Dropout(0.5),

    Dense(1, activation='sigmoid')  # Assuming binary classification

])

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

# Train the model

model.fit(X\_train, y\_train, validation\_data=(X\_test, y\_test), epochs=10, batch\_size=32)

# Save the model

model.save('aids\_classification\_model.h5')

Output:-

Epoch 1/10

54/54 [==============================] - 1s 6ms/step - loss: 0.6094 - accuracy: 0.6908 - val\_loss: 0.3597 - val\_accuracy: 0.8995

Epoch 2/10

54/54 [==============================] - 0s 2ms/step - loss: 0.3864 - accuracy: 0.8440 - val\_loss: 0.2384 - val\_accuracy: 0.9346

Epoch 3/10

54/54 [==============================] - 0s 2ms/step - loss: 0.3025 - accuracy: 0.9018 - val\_loss: 0.1952 - val\_accuracy: 0.9439

Epoch 4/10

54/54 [==============================] - 0s 3ms/step - loss: 0.2554 - accuracy: 0.9182 - val\_loss: 0.1860 - val\_accuracy: 0.9439

Epoch 5/10

54/54 [==============================] - 0s 2ms/step - loss: 0.2387 - accuracy: 0.9375 - val\_loss: 0.1820 - val\_accuracy: 0.9439

Epoch 6/10

54/54 [==============================] - 0s 3ms/step - loss: 0.2117 - accuracy: 0.9410 - val\_loss: 0.1822 - val\_accuracy: 0.9439

Epoch 7/10

54/54 [==============================] - 0s 2ms/step - loss: 0.2016 - accuracy: 0.9439 - val\_loss: 0.1842 - val\_accuracy: 0.9439

Epoch 8/10

54/54 [==============================] - 0s 2ms/step - loss: 0.2144 - accuracy: 0.9468 - val\_loss: 0.1769 - val\_accuracy: 0.9439

Epoch 9/10

54/54 [==============================] - 0s 2ms/step - loss: 0.2075 - accuracy: 0.9421 - val\_loss: 0.1745 - val\_accuracy: 0.9439

Epoch 10/10

54/54 [==============================] - 0s 2ms/step - loss: 0.1879 - accuracy: 0.9527 - val\_loss: 0.1743 - val\_accuracy: 0.9486

/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file via `model.save()`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my\_model.keras')`.

saving\_api.save\_model(

%%writefile app.py

import streamlit as st

import tensorflow as tf

import numpy as np

import joblib

# Load the trained model

model = tf.keras.models.load\_model('aids\_classification\_model.h5')

# Load the scaler

scaler = joblib.load('scaler.pkl')

# Streamlit app

st.title("AIDS Classification")

st.write("Input features for classification")

# Get feature names from the scaler

feature\_names = scaler.feature\_names\_in\_

# Create input fields for each feature

feature\_input = []

for feature\_name in feature\_names:

    feature\_value = st.number\_input(f"{feature\_name}")

    feature\_input.append(feature\_value)

if st.button('Classify'):

    # Preprocess the input

    input\_array = np.array(feature\_input).reshape(1, -1)

    input\_array = scaler.transform(input\_array)

    # Predict the class

    prediction = model.predict(input\_array)

    predicted\_class = 'Positive' if prediction >= 0.5 else 'Negative'

    confidence = prediction[0][0]

    st.write(f"Prediction: {predicted\_class} with confidence {confidence:.2f}")

! pip install streamlit -q

!wget -q -O - ipv4.icanhazip.com

35.247.11.114

! streamlit run app.py & npx localtunnel --port 8501

Collecting usage statistics. To deactivate, set browser.gatherUsageStats to false.

**You can now view your Streamlit app in your browser.**

Local URL: [**http://localhost:8501**](http://localhost:8501/)

Network URL: [**http://172.28.0.12:8501**](http://172.28.0.12:8501/)

External URL: [**http://34.106.83.13:8501**](http://34.106.83.13:8501/)

npx: installed 22 in 4.736s

your url is: [https://wise-fans-yell.loca.lt](https://wise-fans-yell.loca.lt/)

2024-06-11 04:48:27.239051: E external/local\_xla/xla/stream\_executor/cuda/cuda\_dnn.cc:9261] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN when one has already been registered

2024-06-11 04:48:27.239145: E external/local\_xla/xla/stream\_executor/cuda/cuda\_fft.cc:607] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT when one has already been registered

2024-06-11 04:48:27.240861: E external/local\_xla/xla/stream\_executor/cuda/cuda\_blas.cc:1515] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered

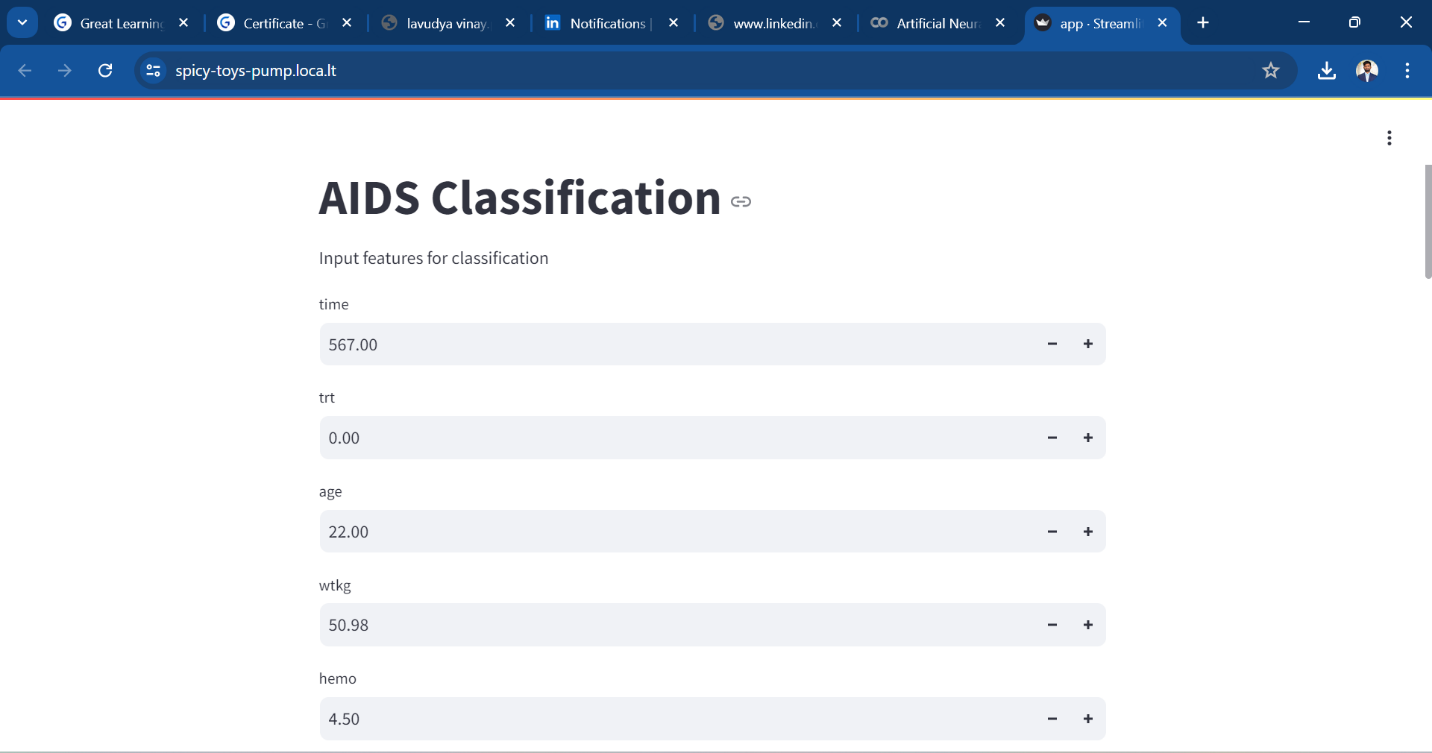
2024-06-11 04:48:28.510317: W tensorflow/compiler/tf2tensorrt/utils/py\_utils.cc:38] TF-TRT Warning: Could not find TensorRT

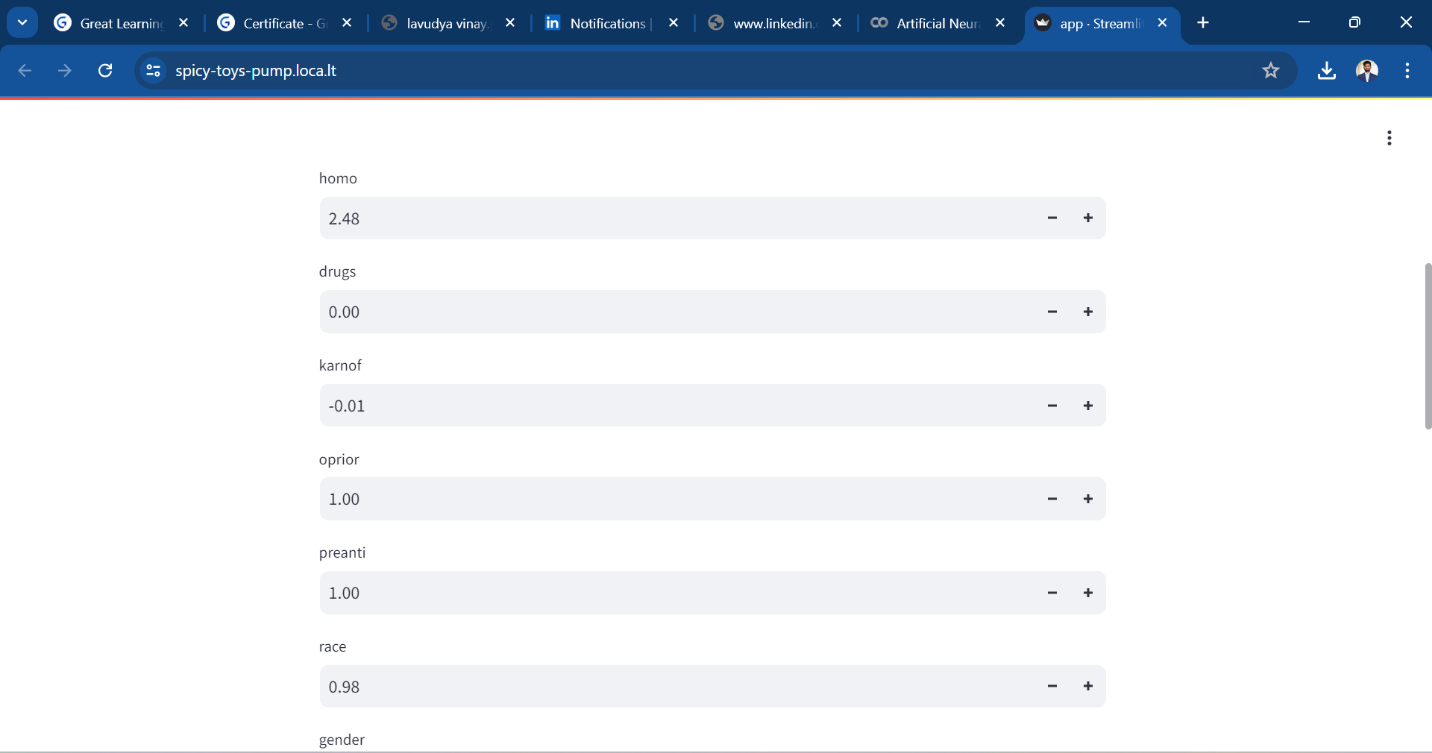
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but StandardScaler was fitted with feature names

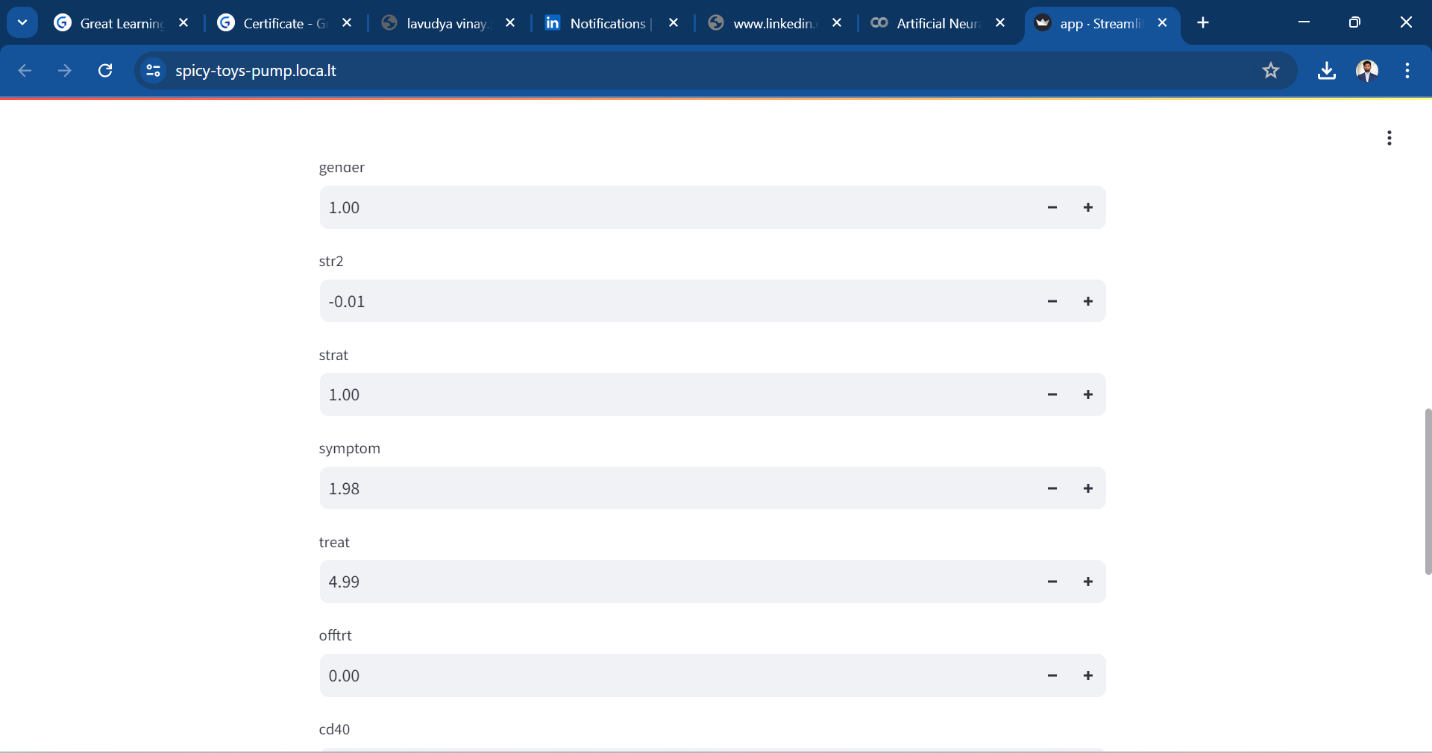
warnings.warn(

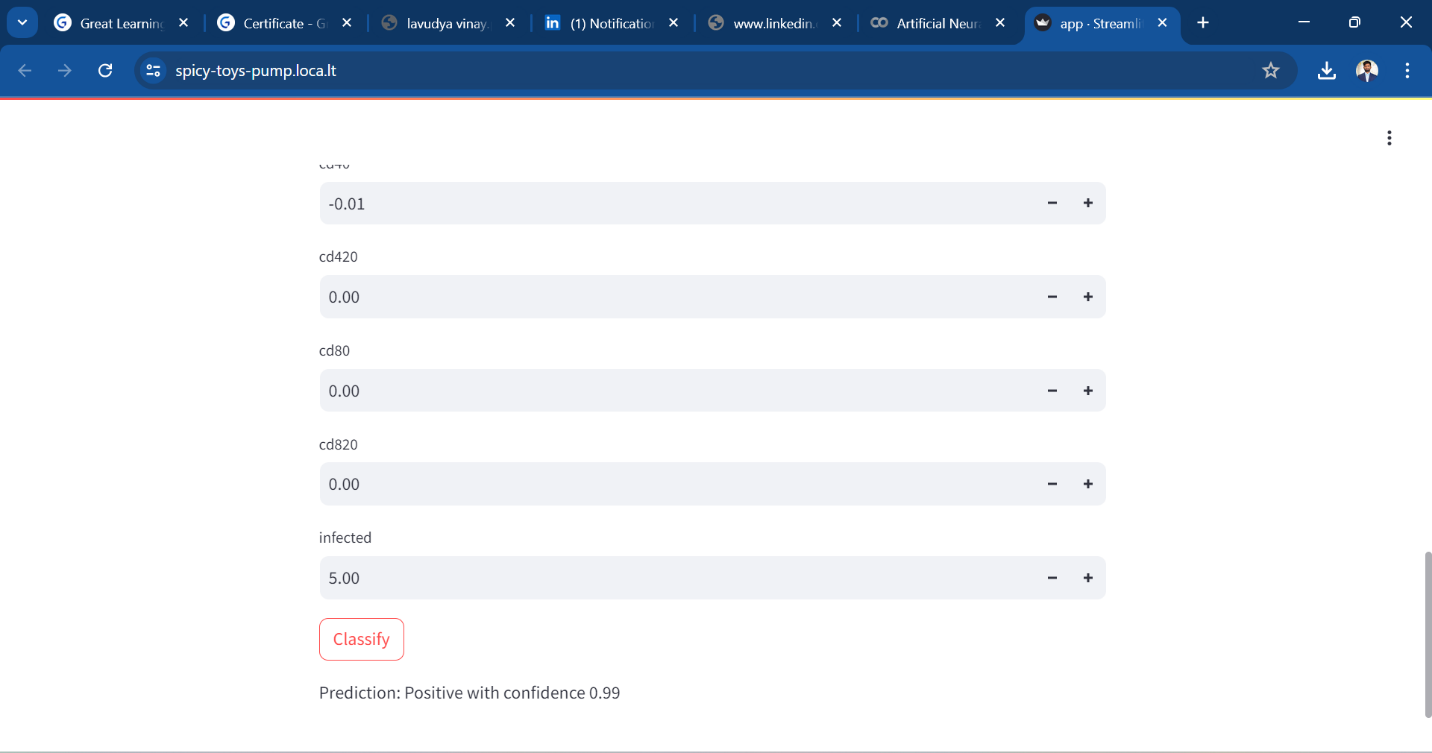
1/1 [==============================] - 0s 88ms/step

STREAMLIT









Conclusion:

The classification of AIDS (Acquired Immunodeficiency Syndrome) has evolved significantly over the years to improve diagnosis, treatment, and management of the disease. The classification systems, primarily developed by the Centers for Disease Control and Prevention (CDC) and the World Health Organization (WHO), provide standardized criteria to identify and categorize the progression of HIV (Human Immunodeficiency Virus) infection to AIDS.