

ASSIGNMENT-2

NAME: DANDA SAI PAVANI

REG NO: 20BCR7042

AI Assignment 2

Build an ANN model for Drug classification

This project aims to analyze the relationship between various medical parameters and drug effectiveness. The dataset consists of patient information, including age, sex, blood pressure levels (BP), cholesterol levels, sodium-to-potassium ratio (Na_to_K), drug type, and corresponding labels. The goal is to develop a model that can accurately predict the class or category of a given drug based on its features.

Dataset Link: <https://www.kaggle.com/prathamtripathi/drug-classification>

Task 1: Read the dataset and do data pre-processing

Task 2: Build the ANN model with (input layer, min 3 hidden layers & output layer)

Task 3: Test the model with random data

TASK-1

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler

dataset = pd.read_csv('drug200.csv')

dataset = dataset.dropna()

X = dataset.drop(['Drug'], axis=1)
y = dataset['Drug']

label_encoder = LabelEncoder()
X['Sex'] = label_encoder.fit_transform(X['Sex'])
X['BP'] = label_encoder.fit_transform(X['BP'])
X['Cholesterol'] = label_encoder.fit_transform(X['Cholesterol'])

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,
test_size=0.2, random_state=42)
```

TASK-2

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

model = Sequential()

model.add(Dense(units=64, activation='relu',
input_dim=X_train.shape[1]))
```

```

model.add(Dense(units=64, activation='relu'))
model.add(Dense(units=32, activation='relu'))
model.add(Dense(units=16, activation='relu'))

model.add(Dense(units=len(label_encoder.classes_),
activation='softmax'))

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

```

TASK-3

```

import numpy as np

random_data = np.array([[40, 0, 2, 0, 2.5]])

random_data_scaled = scaler.transform(random_data)

predictions = model.predict(random_data_scaled)
predicted_class =
label_encoder.inverse_transform([np.argmax(predictions)])

print('Predicted Drug Class:', predicted_class)

```

OUTPUT:

```

import numpy as np


random_data = np.array([[40, 0, 2, 0, 2.5]])

random_data_scaled = scaler.transform(random_data)

predictions = model.predict(random_data_scaled)
predicted_class = label_encoder.inverse_transform([np.argmax(predictions)])

print("Predicted Drug Class:", predicted_class)

```

 /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 491ms/step
Predicted Drug Class: ['HIGH']