In [ ]:
```
# Advait Deochakke
# SmartBridge AI course
# Assignment 2
```

In [ ]:
```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split

df = pd.read_csv("drug200.csv")
```

In [ ]:
```
df.columns
```

Out[ ]:
```
Index(['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug'], dtype='object')
```

In [ ]:
```
df.head(5)
```

Out[ ]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	DrugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	DrugY

In [ ]:
```
# Handle missing values if any
df = df.dropna()

# Convert categorical variables to numerical format
label_encoder = LabelEncoder()
df['Sex'] = label_encoder.fit_transform(df['Sex'])
df['BP'] = label_encoder.fit_transform(df['BP'])
df['Cholesterol'] = label_encoder.fit_transform(df['Cholesterol'])
df['Drug'] = label_encoder.fit_transform(df['Drug'])

# Normalize numerical features
scaler = StandardScaler()
df[['Age', 'Na_to_K']] = scaler.fit_transform(df[['Age', 'Na_to_K']])

# Split the dataset into input features and labels
X = df.drop('Drug', axis=1)
y = df['Drug']
```

In [ ]:
```
from tensorflow import keras
from tensorflow.keras import layers

# Build the model
model = keras.Sequential([
    layers.Dense(64, activation='relu', input_shape=(5,)),  # Input layer
    layers.Dense(128, activation='relu'),  # Hidden layer 1
    layers.Dense(64, activation='relu'),  # Hidden layer 2
    layers.Dense(32, activation='relu'),  # Hidden layer 3
    layers.Dense(5, activation='softmax')  # Output layer with 5 classes
])

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

# Train the model
model.fit(X, y, epochs=50, batch_size=32)
```

```
Epoch 1/50
7/7 [==============================] - 1s 2ms/step - loss: 1.5850 - accuracy: 0.2650
Epoch 2/50
7/7 [==============================] - 0s 1ms/step - loss: 1.4287 - accuracy: 0.4950
Epoch 3/50
7/7 [==============================] - 0s 2ms/step - loss: 1.2718 - accuracy: 0.5950
Epoch 4/50
7/7 [==============================] - 0s 1ms/step - loss: 1.0919 - accuracy: 0.6500
Epoch 5/50
7/7 [==============================] - 0s 998us/step - loss: 0.8997 - accuracy: 0.7200
Epoch 6/50
7/7 [==============================] - 0s 1ms/step - loss: 0.7279 - accuracy: 0.7600
Epoch 7/50
7/7 [==============================] - 0s 2ms/step - loss: 0.6033 - accuracy: 0.7750
Epoch 8/50
7/7 [==============================] - 0s 2ms/step - loss: 0.4976 - accuracy: 0.8300
Epoch 9/50
7/7 [==============================] - 0s 2ms/step - loss: 0.3980 - accuracy: 0.8850
Epoch 10/50
7/7 [==============================] - 0s 2ms/step - loss: 0.3306 - accuracy: 0.9050
Epoch 11/50
7/7 [==============================] - 0s 1ms/step - loss: 0.2731 - accuracy: 0.9200
Epoch 12/50
7/7 [==============================] - 0s 2ms/step - loss: 0.2256 - accuracy: 0.9400
Epoch 13/50
7/7 [==============================] - 0s 2ms/step - loss: 0.1899 - accuracy: 0.9500
Epoch 14/50
7/7 [==============================] - 0s 2ms/step - loss: 0.1749 - accuracy: 0.9600
Epoch 15/50
7/7 [==============================] - 0s 1ms/step - loss: 0.1706 - accuracy: 0.9450
Epoch 16/50
7/7 [==============================] - 0s 2ms/step - loss: 0.1249 - accuracy: 0.9700
Epoch 17/50
7/7 [==============================] - 0s 2ms/step - loss: 0.1136 - accuracy: 0.9750
Epoch 18/50
7/7 [==============================] - 0s 2ms/step - loss: 0.0990 - accuracy: 0.9800
Epoch 19/50
7/7 [==============================] - 0s 1000us/step - loss: 0.0911 - accuracy: 0.9700
Epoch 20/50
7/7 [==============================] - 0s 2ms/step - loss: 0.0821 - accuracy: 0.9850
Epoch 21/50
7/7 [==============================] - 0s 2ms/step - loss: 0.0686 - accuracy: 0.9800
Epoch 22/50
7/7 [==============================] - 0s 2ms/step - loss: 0.0641 - accuracy: 0.9950
Epoch 23/50
7/7 [==============================] - 0s 2ms/step - loss: 0.0678 - accuracy: 0.9900
Epoch 24/50
7/7 [==============================] - 0s 2ms/step - loss: 0.0665 - accuracy: 0.9750
Epoch 25/50
7/7 [==============================] - 0s 1ms/step - loss: 0.0584 - accuracy: 0.9950
Epoch 26/50
7/7 [==============================] - 0s 1ms/step - loss: 0.0429 - accuracy: 1.0000
Epoch 27/50
7/7 [==============================] - 0s 2ms/step - loss: 0.0493 - accuracy: 0.9850
Epoch 28/50
7/7 [==============================] - 0s 1ms/step - loss: 0.0467 - accuracy: 0.9900
Epoch 29/50
7/7 [==============================] - 0s 1ms/step - loss: 0.0475 - accuracy: 0.9900
Epoch 30/50
7/7 [==============================] - 0s 1ms/step - loss: 0.0318 - accuracy: 1.0000
Epoch 31/50
7/7 [==============================] - 0s 2ms/step - loss: 0.0378 - accuracy: 0.9950
Epoch 32/50
7/7 [==============================] - 0s 2ms/step - loss: 0.0310 - accuracy: 0.9950
Epoch 33/50
7/7 [==============================] - 0s 1ms/step - loss: 0.0245 - accuracy: 1.0000
Epoch 34/50
7/7 [==============================] - 0s 1ms/step - loss: 0.0236 - accuracy: 1.0000
Epoch 35/50
7/7 [==============================] - 0s 1ms/step - loss: 0.0214 - accuracy: 1.0000
Epoch 36/50
7/7 [==============================] - 0s 2ms/step - loss: 0.0187 - accuracy: 1.0000
Epoch 37/50
7/7 [==============================] - 0s 1ms/step - loss: 0.0177 - accuracy: 1.0000
Epoch 38/50
7/7 [==============================] - 0s 2ms/step - loss: 0.0171 - accuracy: 1.0000
Epoch 39/50
7/7 [==============================] - 0s 2ms/step - loss: 0.0160 - accuracy: 1.0000
Epoch 40/50
7/7 [==============================] - 0s 1ms/step - loss: 0.0146 - accuracy: 1.0000
Epoch 41/50
7/7 [==============================] - 0s 1ms/step - loss: 0.0150 - accuracy: 1.0000
Epoch 42/50
7/7 [==============================] - 0s 2ms/step - loss: 0.0130 - accuracy: 1.0000
Epoch 43/50
7/7 [==============================] - 0s 3ms/step - loss: 0.0123 - accuracy: 1.0000
Epoch 44/50
7/7 [==============================] - 0s 2ms/step - loss: 0.0113 - accuracy: 1.0000
Epoch 45/50
7/7 [==============================] - 0s 2ms/step - loss: 0.0123 - accuracy: 1.0000
Epoch 46/50
7/7 [==============================] - 0s 2ms/step - loss: 0.0100 - accuracy: 1.0000
Epoch 47/50
7/7 [==============================] - 0s 1ms/step - loss: 0.0105 - accuracy: 1.0000
Epoch 48/50
7/7 [==============================] - 0s 1ms/step - loss: 0.0092 - accuracy: 1.0000
Epoch 49/50
7/7 [==============================] - 0s 1ms/step - loss: 0.0085 - accuracy: 1.0000
Epoch 50/50
7/7 [==============================] - 0s 999us/step - loss: 0.0153 - accuracy: 0.9950
```

Out[ ]:
```
<keras.callbacks.History at 0x2103b183a90>
```

In [ ]:
```
import numpy as np

# Generate random data for testing
random_data = np.array([[35, 0, 1, 0, 15.0]])

# Preprocess the random data (normalize numerical features)
random_data[:, [0, 4]] = scaler.transform(random_data[:, [0, 4]])

# Make predictions
predictions = model.predict(random_data)
predicted_class = np.argmax(predictions[0])

# Convert the predicted class back to the original label
predicted_drug = label_encoder.inverse_transform([predicted_class])[0]

print("Predicted Drug:", predicted_drug)
```

```
1/1 [==============================] - 0s 94ms/step
Predicted Drug: DrugY
c:\Users\advai\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning: X does not have valid feature names, but StandardScaler was fitted with feature names
  warnings.warn(
```