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/datasets/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

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
df = pd.read_csv('/content/drug200.csv')
df.head()
```

	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

```
features = df.drop('Drug', axis=1)
labels = df['Drug']
```

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


features = pd.get_dummies(features, columns=['Sex', 'BP', 'Cholesterol'])
df.head()
```

	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

```
X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=0.2, random_state=42)
```

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
print("Size of X_train:", X_train.shape)
print("Size of X_test:", X_test.shape)
```

```
Size of X_train: (160, 9)
Size of X_test: (40, 9)
```

Task-2

```
from tensorflow import keras
from tensorflow.keras import layers
model = keras.Sequential()
```

```
# Input layer
model.add(layers.Dense(64, activation='relu', input_shape=(X_train.shape[1],)))

# three hidden layers
model.add(layers.Dense(32, activation='relu'))
model.add(layers.Dense(16, activation='relu'))
model.add(layers.Dense(8, activation='relu'))

# output layer
model.add(layers.Dense(5, activation='softmax'))
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 64)                640

 dense_1 (Dense)             (None, 32)                2080

 dense_2 (Dense)             (None, 16)                528

 dense_3 (Dense)             (None, 8)                 136

 dense_4 (Dense)             (None, 5)                 45

=================================================================
Total params: 3,429
Trainable params: 3,429
Non-trainable params: 0
_________________________________________________________________
```

Task-3

```
import numpy as np
random_input = np.random.randn(1, X_train.shape[1])

predicted_classes = model.predict(random_input)
predicted_class = np.argmax(predicted_classes)

print("Predicted Class:", predicted_class)
```

```
1/1 [==============================] - 0s 199ms/step
Predicted Class: 2
```