

```

# Task-1
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler

data = pd.read_csv("drug_classification_dataset.csv")

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

categorical_features = ['Sex', 'BP', 'Cholesterol']
encoder = LabelEncoder()
for feature in categorical_features:
    features[feature] = encoder.fit_transform(features[feature])

scaler = StandardScaler()
numerical_features = ['Age', 'Na_to_K']
features[numerical_features] =
scaler.fit_transform(features[numerical_features])

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

```

```

# Task-2
import keras
from keras.models import Sequential
from keras.layers import Dense

model = Sequential()

model.add(Dense(16, activation='relu', input_shape=(5,)))

model.add(Dense(16, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(4, activation='relu'))

model.add(Dense(1, activation='sigmoid'))

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

model.summary()

```

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 16)	96
dense_1 (Dense)	(None, 16)	272
dense_2 (Dense)	(None, 8)	136

dense_3 (Dense)	(None, 4)	36
dense_4 (Dense)	(None, 1)	5

```
=====
Total params: 545
Trainable params: 545
Non-trainable params: 0
=====
```

Task-3

```
import numpy as np
```

```
random_data = np.array([[50, 0, 2, 2, 20]])
```

```
random_data[:, [0, 4]] = scaler.transform(random_data[:, [0, 4]])
```

```
C:\Users\Ashmit Raghuvanshi\anaconda3\lib\site-packages\sklearn\
base.py:450: UserWarning: X does not have valid feature names, but
StandardScaler was fitted with feature names
  warnings.warn(
```

```
predictions = model.predict(random_data)
```

```
1/1 [=====] - 0s 311ms/step
```

```
predicted_class = np.where(predictions > 0.5, 'DrugY', 'DrugX')
```

```
print(predicted_class)
```

```
[['DrugX']]
```