SMART BRIDGE

ARTIFICIAL INTELLIGENCE

ASSIGNMENT-2

NAME:KATARI VISHNU RUSHIKESH VARMA

REG-NO:20BCI0241

TASK-1

LOAD DATASET:

CODE:

import numpy as np

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import LabelEncoder, StandardScaler

data1=pd.read_csv("C:/Users/hp/Downloads/archive/drug200.csv")

data1.head()

Output:

Out[36]:

	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

DATA PREPROCESSING:

CODE:

#Task1:Data Preprocessing

#find missing values

data1.isnull().any()

```
Out[37]:  Age          False
          Sex          False
          BP           False
          Cholesterol  False
          Na_to_K      False
          Drug         False
          dtype: bool
```

Drop rows with missing values

data1.dropna(inplace=True)

data1.head()

```
Out[38]:
     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
```

#check for categorical columns

categorical_columns = data1.select_dtypes(include=["object"]).columns

print(categorical_columns)

```
Index(['Sex', 'BP', 'Cholesterol', 'Drug'], dtype='object')
```


Split features and labels

import matplotlib.pyplot as plt

X = data1.drop("Drug", axis=1)

y = data1["Drug"]

#one hot encoding for x

Perform one-hot encoding

encoded_df = pd.get_dummies(data1, columns=['Sex', 'BP', 'Cholesterol'])

Perform one-hot encoding

le = LabelEncoder()

```python
y = le.fit_transform(y)
```

Splitting the dataset into training and testing sets

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

TASK-2 Build The ANN MODEL

CODE:

```python
#Task-2-Build the ANN Model


from tensorflow import keras
from tensorflow.keras import layers


# Initialize the model
model = keras.Sequential()


# Add the input layer and the first hidden layer
model.add(layers.Dense(64, activation='relu', input_shape=(4,)))


# Add two more hidden layers
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(32, activation='relu'))


# Add the output layer
model.add(layers.Dense(3, activation='softmax'))


# Print the model summary
model.summary()
```

OUTPUT:

```
Model: "sequential_1"
_____
 Layer (type)                Output Shape              Param #
=================================================================
 dense_4 (Dense)             (None, 64)                320

 dense_5 (Dense)             (None, 64)                4160

 dense_6 (Dense)             (None, 32)                2080

 dense_7 (Dense)             (None, 3)                 99

=================================================================
Total params: 6,659
Trainable params: 6,659
Non-trainable params: 0
_____
```

TASK:Test the model with Random data

CODE:

```
#task3-To test the model with random data
import numpy as np


# Generate random input sample
random_input = np.random.rand(1, 4)


# Make predictions with the trained model
predictions = model.predict(random_input)


# Decode the predicted labels
predicted_labels = le.inverse_transform(np.argmax(predictions, axis=1))


# Print the predicted labels
print("Predicted Labels:", predicted_labels)
```

OUTPUT:

```
1/1 [==============================] - 0s 118ms/step
Predicted Labels: ['drugA']
```