ASSIGNMENT 1

NAME: DANDA SAI PAVANI

REG NO: 20BCR7042

AI Assignment 1	
Task - 1	Create a pandas dataframe (DataFrame name as 'df') with numpy random values (4 features and 4 observation)
Task - 2	Rename the task - 1 'df' dataframe column names to 'Random value 1', 'Random value 2', 'Random value 3' & 'Random value 4'
Task - 3	Find the descriptive statistics of the 'df' dataframe.
Task - 4	Check for the null values in 'df' and find the data type of the columns.
Task - 5	Display the 'Random value 2' & 'Random value 3' columns with location method and index location method.

TASK-1

```
import pandas as pd
import numpy as np
np.random.seed(42)
data = np.random.rand(4, 4)
df = pd.DataFrame(data, columns=['Feature 1', 'Feature 2', 'Feature 3', 'Feature
    4'])
print(df)
```

TASK-2

```
df = df.rename(columns={'Feature 1': 'Random value 1',
                        'Feature 2': 'Random value 2',
                        'Feature 3': 'Random value 3',
                        'Feature 4': 'Random value 4'})
```

TASK-3

```
statistics = df.describe()
print(statistics)
```

OUTPUT:

```
Shell                                                    Clear
       Random value 1  Random value 2  Random value 3  Random value 4
count        4.000000        4.000000        4.000000        4.000000
mean         0.491029        0.506780        0.248122        0.654537
std          0.291252        0.386153        0.329856        0.350875
min          0.156019        0.155995        0.020584        0.183405
25%          0.319910        0.198253        0.048709        0.494845
50%          0.487828        0.460206        0.119954        0.732417
75%          0.658947        0.768733        0.319367        0.892110
max          0.832443        0.950714        0.731994        0.969910
>
```

TASK-4

```
null_values = df.isnull().sum()
data_types = df.dtypes
print("Null Values:")
print(null_values)
print("\nData Types:")
print(data_types)
```

OUTPUT:

```
Null Values:
Random value 1    0
Random value 2    0
Random value 3    0
Random value 4    0
dtype: int64

Data Types:
Random value 1    float64
Random value 2    float64
Random value 3    float64
Random value 4    float64
dtype: object
>
```

TASK-5

```
random_value_2_3_loc = df.loc[:, ['Random value 2', 'Random value 3']]
print("Using label-based location (loc):")
print(random_value_2_3_loc)

random_value_2_3_iloc = df.iloc[:, [1, 2]]
print("\nUsing index-based location (iloc):")
print(random_value_2_3_iloc)
```

OUTPUT:

```
Using label-based location (loc):
Random value 2  Random value 3
0        0.950714        0.731994
1        0.155995        0.058084
2        0.708073        0.020584
3        0.212339        0.181825

Using index-based location (iloc):
Random value 2  Random value 3
0        0.950714        0.731994
1        0.155995        0.058084
2        0.708073        0.020584
3        0.212339        0.181825
>
```

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/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

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: