Machine Learning - CS5710
Assignment 1
Rishma Reddy Nalla
700752916

GitHub Link: https://github.com/RishmaReddy-Nalla/CS-5710/tree/main/Assignment

1. Read the provided CSV file 'data.csv'. https://drive.google.com/drive/folders/1h8C3mLsso-R-sIOLsvoYwPLzy2fJ4IOF?usp=sharing

```
import pandas as pd
# Load the dataset from a CSV file
df = pd.read_csv("/workspaces/CS-5710/Assignment/data.csv")
```
✓ 0.0s

2. Show the basic statistical description about the data.

```
# Generate descriptive statistics of the DataFrame
df.describe()
```
✓ 0.0s

	Duration	Pulse	Maxpulse	Calories
count	169.000000	169.000000	169.000000	164.000000
mean	63.846154	107.461538	134.047337	375.790244
std	42.299949	14.510259	16.450434	266.379919
min	15.000000	80.000000	100.000000	50.300000
25%	45.000000	100.000000	124.000000	250.925000
50%	60.000000	105.000000	131.000000	318.600000
75%	60.000000	111.000000	141.000000	387.600000
max	300.000000	159.000000	184.000000	1860.400000

3. Check if the data has null values.
a. Replace the null values with the mean

```
# Check for missing values in the DataFrame
df.isnull()
```

✓ 0.0s

	Duration	Pulse	Maxpulse	Calories
0	False	False	False	False
1	False	False	False	False
2	False	False	False	False
3	False	False	False	False
4	False	False	False	False
...	...	...	...	...
164	False	False	False	False
165	False	False	False	False
166	False	False	False	False
167	False	False	False	False
168	False	False	False	False

169 rows × 4 columns

```
# Fill missing values with the mean of each column
df = df.fillna(df.mean())
```

✓ 0.0s

```
# Check if there are any missing values left in the DataFrame
df.isna().any()
```

✓ 0.0s

```
Duration    False
Pulse       False
Maxpulse    False
Calories    False
dtype: bool
```

4. Select at least two columns and aggregate the data using: min, max, count, mean.

```
# Aggregate the DataFrame to find the min, max, count, and mean for numeric columns
df.agg(["min","max","count","mean"])
```

9] ✓ 0.0s

	Duration	Pulse	Maxpulse	Calories
min	15.000000	80.000000	100.000000	50.300000
max	300.000000	159.000000	184.000000	1860.400000
count	169.000000	169.000000	169.000000	169.000000
mean	63.846154	107.461538	134.047337	375.790244

5. Filter the dataframe to select the rows with calories values between 500 and 1000

```
# Filter the DataFrame for rows where Calories are between 500 and 1000
df[(df["Calories"] >  500) & (df["Calories"] < 1000)]
```

] ✓ 0.0s

	Duration	Pulse	Maxpulse	Calories
51	80	123	146	643.1
62	160	109	135	853.0
65	180	90	130	800.4
66	150	105	135	873.4
67	150	107	130	816.0
72	90	100	127	700.0
73	150	97	127	953.2
75	90	98	125	563.2
78	120	100	130	500.4
90	180	101	127	600.1
99	90	93	124	604.1
103	90	90	100	500.4
106	180	90	120	800.3
108	90	90	120	500.3

6. Filter the dataframe to select the rows with calories values > 500 and pulse < 100.

```
# Filter the DataFrame for rows where Calories are greater than 500 and Pulse is less than 100
df[(df["Calories"] >  500) & (df["Pulse"] < 100)]
```

✓ 0.0s

	Duration	Pulse	Maxpulse	Calories
65	180	90	130	800.4
70	150	97	129	1115.0
73	150	97	127	953.2
75	90	98	125	563.2
99	90	93	124	604.1
103	90	90	100	500.4
106	180	90	120	800.3
108	90	90	120	500.3

7. Create a new “df_modified” dataframe that contains all the columns from df except for “Maxpulse”.

```
# Create a new DataFrame with only specific columns
df_modified = df[["Duration", "Pulse", "Calories"]]
```

✓ 0.0s

```
# Display the modified DataFrame
df_modified
```

✓ 0.0s

	Duration	Pulse	Calories
0	60	110	409.1
1	60	117	479.0
2	60	103	340.0
3	45	109	282.4
4	45	117	406.0
...	...	...	...
164	60	105	290.8
165	60	110	300.0
166	60	115	310.2
167	75	120	320.4
168	75	125	330.4

169 rows × 3 columns

8. Delete the “Maxpulse” column from the main df dataframe

```
# Drop the 'Maxpulse' column from the original DataFrame
df.drop(columns=["Maxpulse"], inplace=True)
```

[44] ✓ 0.0s

```
df
```

[45] ✓ 0.0s

...

	Duration	Pulse	Calories
0	60	110	409.1
1	60	117	479.0
2	60	103	340.0
3	45	109	282.4
4	45	117	406.0
...	...	...	...
164	60	105	290.8
165	60	110	300.0
166	60	115	310.2
167	75	120	320.4
168	75	125	330.4

169 rows × 3 columns

9. Convert the datatype of Calories column to int datatype.

```
# Print data types before conversion
print("Data Types before conversion")
print(df.dtypes)

# Convert the 'Calories' column to integer type
df['Calories'] = df["Calories"].astype(int)

# Print data types after conversion
print("\n","Data Types after Conversion")
print(df.dtypes)
```

8] ✓ 0.0s

```
Data Types before conversion
Duration    int64
Pulse       int64
Calories    int64
dtype: object

 Data Types after Conversion
Duration    int64
Pulse       int64
Calories    int64
dtype: object
```