

# CS 5720 Neural Network Deep Learning

## ICP-4

CRN	23441
NAME	MANOJ BALA
EMAIL	mx40210@ucmo.edu
STUDENT_ID	700754021

### GitHub Repository:

<https://github.com/mxb40210/700754021-NeuralNetworkDeepLearning>

### Assignment 4:

<https://github.com/mxb40210/700754021-NeuralNetworkDeepLearning/tree/main/assignments/assignment4>

### 1. Question 1

The screenshot displays a Jupyter Notebook titled "700754021-NeuralNetworkDeepLearning - solutions.ipynb". The notebook is open to the "1. Data Manipulation" section. The code cells and their outputs are as follows:

**Cell 45:** Imports pandas as pd.

```
In 45 1 # Imports
      2 import pandas as pd
```

**Cell 46:** Reads the CSV file 'data.csv' and prints the shape of the resulting DataFrame.

```
In 46 1 # a. Read the provided CSV file 'data.csv'
      2
      3 # Path of the csv
      4 data_path = 'data/data.csv'
      5
      6 # Read csv using pandas
      7 data_df = pd.read_csv(data_path)
      8
      9 # Shape (rows, columns)
     10 data_df.shape
```

**Output 46:** (169, 4)

**Cell 47:** Prints the basic statistical description of the data using data\_df.describe().

```
In 47 1 # c. Show the basic statistical description about the data.
      2 data_df.describe()
```

**Output 47:** A summary statistics table for the DataFrame.

	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	307.600000
max	300.000000	159.000000	184.000000	1840.400000

**Cell 48:** Checks if the data has null values using data\_df.isnull().sum().

```
In 48 1 # d. Check if the data has null values.
      2 data_df.isnull().sum()
```

**Output 48:** A series showing the count of null values for each column.

	<unnamed>
Duration	0
Pulse	0
Maxpulse	0
Calories	5

700754021-NeuralNetworkDeepLearning - solutions.ipynb

700754021-NeuralNetworkDeepLearning | assignments | assignment4 | solutions.ipynb

question\_2 (2) | Git: | Python 3 (ipykernel) | Managed: http://localhost:8888

```
In 49 1 # i. Replace null values with the mean
2 data_df.fillna(data_df.mean(), inplace=True)
3
4 data_df.isnull().sum()
```

Out 49 |< 4 rows |> Length: 4, dtype: int64 pd.Series

	<unnamed>
Duration	0
Pulse	0
Maxpulse	0
Calories	0

```
In 50 1 # e. Select at least two columns and aggregate the data using: min, max, count, mean.
2
3 # Select two columns in a List - Duration & Calories
4 aggregate_columns = ['Duration', 'Calories']
5
6 # Aggregate the data by min, max, count, mean for these two columns
7 aggregated_data = data_df[aggregate_columns].agg(['min', 'max', 'count', 'mean'])
8
9 # Print the aggregated
10 print(aggregated_data)
```

	Duration	Calories
min	15.000000	50.300000
max	300.000000	1860.400000
count	169.000000	169.000000
mean	63.846154	375.790244

700754021-NeuralNetworkDeepLearning - solutions.ipynb

700754021-NeuralNetworkDeepLearning | assignments | assignment4 | solutions.ipynb

question\_2 (2) | Git: | Python 3 (ipykernel) | Managed: http://localhost:8888

```
In 51 1 # f. Filter the dataframe to select the rows with calories values between 500 and 1000.
2 filtered_data_by_calories = data_df[(data_df['Calories'] >= 500) & (data_df['Calories'] <= 1000)]
3 print(filtered_data_by_calories)
```

	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
83	120	100	130	500.0
99	180	101	177	800.1

```
In 52 1 # g. Filter the dataframe to select the rows with calories values > 500 and pulse < 100.
2 filtered_data_df = data_df[(data_df['Calories'] > 500) & (data_df['Pulse'] < 100)]
3 print(filtered_data_df)
```

	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

```
700754021-NeuralNetworkDeepLearning - solutions.ipynb
700754021-NeuralNetworkDeepLearning assignments assignment4 solutions.ipynb
question_2 (2) question_3.py
Managed: http://localhost:8888 Python 3 (ipykernel) Trusted

In 53 1 # h. Create a new "df_modified" dataframe that contains all the columns from df except for "Maxpulse".
2 df_modified = data_df.drop(columns=['Maxpulse'])
3 print('data_df columns: {}'.format(data_df.columns))
4 print('df_modified columns: {}'.format(df_modified.columns))

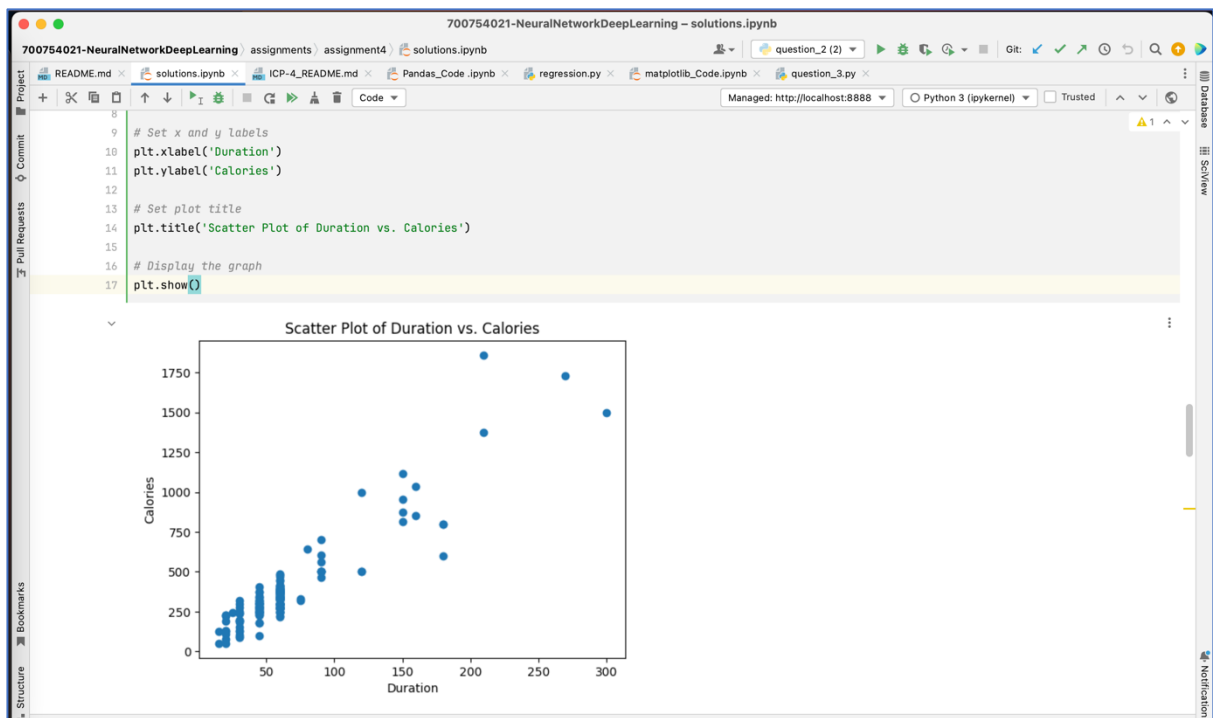
data_df columns: Index(['Duration', 'Pulse', 'Maxpulse', 'Calories'], dtype='object')
df_modified columns: Index(['Duration', 'Pulse', 'Calories'], dtype='object')

In 54 1 # i. Delete the "Maxpulse" column from the main df dataframe
2 print('Before deleting maxpulse, data_df columns: {}'.format(data_df.columns))
3 data_df.drop(columns=['Maxpulse'], inplace=True)
4 print('After deleting maxpulse, data_df columns: {}'.format(data_df.columns))

Before deleting maxpulse, data_df columns: Index(['Duration', 'Pulse', 'Maxpulse', 'Calories'], dtype='object')
After deleting maxpulse, data_df columns: Index(['Duration', 'Pulse', 'Calories'], dtype='object')

In 55 1 # j. Convert the datatype of Calories column to int datatype.
2 print('Before converting to int, Calorie datatype: {}'.format(data_df['Calories'].dtype))
3 data_df['Calories'] = data_df['Calories'].astype(int)
4 print('After converting to int, Calorie datatype: {}'.format(data_df['Calories'].dtype))

Before converting to int, Calorie datatype: float64
After converting to int, Calorie datatype: int64
```



## 2. Question 2

