

## ML\_Assignment 3

Name: Divya Kudipudi

Id: 700765919

1. Read the provided CSV file 'data.csv'. <https://drive.google.com/drive/folders/1h8C3mLsso-R-sIOLsvoYwPLzy2fJ4IOF?usp=sharing>
2. Show the basic statistical description about the data.
3. Check if the data has null values. a. Replace the null values with the mean
4. Select at least two columns and aggregate the data using: min, max, count, mean.
5. Filter the dataframe to select the rows with calories values between 500 and 1000.
6. Filter the dataframe to select the rows with calories values > 500 and pulse < 100.
7. Create a new "df\_modified" dataframe that contains all the columns from df except for "Maxpulse".
8. Delete the "Maxpulse" column from the main df dataframe
9. Convert the datatype of Calories column to int datatype.

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

```
df = pd.read_csv("data.csv")  
df.head()
```

	Duration	Pulse	Maxpulse	Calories
0	60	110	130	409.1
1	60	117	145	479.0
2	60	103	135	340.0
3	45	109	175	282.4
4	45	117	148	406.0

The code imports required libraries for working with tabular data and implementing algorithms, including numpy, pandas, seaborn, matplotlib, scipy, and scikit-learn. It also suppresses warnings. The provided CSV file 'data.csv' is read using pandas into a dataframe called 'df', and the first few rows of the dataframe are displayed using the 'head()' method

```
#2. Show the basic statistical description about the data.
```

```
df.describe()
```

	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

The code calls the 'describe()' method on the dataframe 'df', which returns basic statistical information about the data such as count, mean, standard deviation, minimum value, maximum value, and quartile values for each numerical column in the dataframe

```
#3. Check if the data has null values.
```

```
df.isnull().any()
```

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

[+ Code](#)[+ Markdown](#)

```
#Replace the null values with the mean
```

```
df.fillna(df.mean(), inplace=True)
df.isnull().any()
```

```
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.
```

```
df.agg({'Maxpulse':['min','max','count','mean'],'Calories':['min','max','count','mean']})
```

	Maxpulse	Calories
min	100.000000	50.300000
max	184.000000	1860.400000
count	169.000000	169.000000
mean	134.047337	375.790244

The code first calls the 'isnull().any()' method on the dataframe 'df', which checks if there are any null values in the dataframe and returns a boolean value for each column indicating if it contains null values. Then, it fills the null values with the mean using the 'fillna()' method and the 'mean()' function, and checks again for null values using 'isnull().any()'.

Finally, the code selects two columns ('Maxpulse' and 'Calories') and aggregates the data using the 'agg()' method. The aggregation operations used are 'min', 'max', 'count', and 'mean'. The resulting dataframe shows the minimum, maximum, count, and mean values for each selected column.

jupyter Machine\_learning\_Assignment-4 (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel) C

In [13]: #5. Filter the dataframe to select the rows with calories values between 500 and 1000.

```
df.loc[(df['Calories']>500)&(df['Calories']<1000)]
```

Out[13]:

	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

The code filters the dataframe 'df' using the 'loc[]' method to select rows where the 'Calories' column has values between 500 and 1000 (exclusive). The resulting dataframe contains only the rows that satisfy the condition.

In [14]: #6. Filter the dataframe to select the rows with calories values > 500 and pulse < 100.

```
df.loc[(df['Calories']>500)&(df['Pulse']<100)]
```

Out[14]:

	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

In [15]: #7. Create a new "df\_modified" dataframe that contains all the columns from df except for "Maxpulse".

```
df_modified = df[['Duration', 'Pulse', 'Calories']]
df_modified.head()
```

Out[15]:

	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

The code filters the dataframe 'df' to select rows where the 'Calories' column has values greater than 500 and the 'Pulse' column has values less than 100, using the 'loc[]' method. The resulting dataframe contains only the rows that satisfy the condition.

Then, the code creates a new dataframe 'df\_modified' by selecting all columns from the original dataframe 'df' except for 'Maxpulse'. This is done by indexing the dataframe with a list of the desired column names. The resulting dataframe contains only the columns 'Duration', 'Pulse', and 'Calories',

```
In [16]: #8. Delete the "Maxpulse" column from the main df dataframe
```

```
del df['Maxpulse']
```

```
In [17]: df.head()
```

```
Out[17]:
```

	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

```
In [18]: df.dtypes
```

```
Out[18]: Duration      int64
Pulse        int64
Calories     float64
dtype: object
```

```
In [19]: #9. Convert the datatype of Calories column to int datatype.
```

```
df['Calories'] = df['Calories'].astype(np.int64)
df.dtypes
```

```
Out[19]: Duration      int64
Pulse        int64
Calories     int64
dtype: object
```

and is assigned to the variable 'df\_modified'. The code deletes the 'Maxpulse' column from the dataframe 'df' using the 'del' statement.

Then, the code converts the datatype of the 'Calories' column from float to integer using the 'astype()' method and the numpy 'int64' datatype. The resulting dataframe 'df' contains the modified 'Calories' column with integer datatype.

Finally, the 'dtypes' attribute is used to display the datatypes of all columns in the dataframe 'df'.

Github: <https://github.com/Divyakudipudi7/Assignment-3>

Video Link:

[https://drive.google.com/file/d/1kDrPmkkIDk2ydvmPMvYkS0mbTS5bVALC/view?usp=share\\_link](https://drive.google.com/file/d/1kDrPmkkIDk2ydvmPMvYkS0mbTS5bVALC/view?usp=share_link)