CS-5710 Machine Learning

Assignment – II

Laxma Reddy Nalla

700732071

GitHub Link: CS-5710/Assignment2 at dev · LaxmaReddy-Nalla/CS-5710 (github.com)

YouTube Link: Youtube Video for assignment 2 https://youtu.be/E3wRChkE62E

Question 1:

Generate an array of 15 random numbers in between 1,20

```
# creating 15 Random numbers using random Module in between 1,20
import random
arr = []
for i in range(1,16):
    arr.append(random.randint(1,20))
arr
```

✓ 0.8s

```
[13, 2, 13, 6, 16, 16, 19, 9, 3, 9, 13, 15, 15, 3, 2]
```

```
# converting Normal random Array into numpy array
import numpy as np
arr1 = np.array(arr)
arr1
```

✓ 0.1s

```
array([13,  2, 13,  6, 16, 16, 19,  9,  3,  9, 13, 15, 15,  3,  2])
```

1. Reshape the array to 3 by 5

```
# Reshaping array into (3,5) rows, column
arr2 = arr1.reshape(3,5)
arr2
```

✓ 0.9s

```
array([[13,  2, 13,  6, 16],
       [16, 19,  9,  3,  9],
       [13, 15, 15,  3,  2]])
```

2. Print array shape.

```
    # printing shape of the array
    print(arr2.shape)
[7] ✓ 0.9s
...  (3, 5)
```

3. Replace the max in each row by 0

```
    # Replacing max values row wise to zero 0
    maxNum = np.amax(arr2, axis=1)
    arr2 = np.where(np.isin(arr2,maxNum), 0, arr2)
    print(arr2)
] ✓ 0.6s
[[13  2 13  6  0]
 [ 0  0  9  3  9]
 [13  0  0  3  2]]
```

1. Read the provided CSV file ‘data.csv’.

Url= https://drive.google.com/drive/folders/1h8C3mLsso-R-sIOLsvoYwPLzy2fJ4IOF?usp=sharing

Downloaded the file from drive and imported using pandas

```
    # importing dataset using pandas
    import pandas as pd
    df = pd.read_csv("data.csv")
[11] ✓ 0.1s
```

2. Show the basic statistical description about the data.

```
# printing Statical analysis of given dataset
df.describe()
```

✓ 0.3s

	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

```
# Printing weather Columns null or not using boolean as True or False
print(df.isnull())
```
[14] ✓ 0.1s

```
     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 x 4 columns]
```

```
# filling null values using mean of the dataset
df = df.fillna(df.mean())
```
[15] ✓ 0.8s

```
print(df.isna().any())
```
[16] ✓ 0.1s

```
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.

```
#  getting  aggregate operation on dataset columns Duration and Pulse
df.iloc[:,0:2].agg(["min","max","count","mean"])
```
[21] ✓ 0.1s

	Duration	Pulse
min	15.000000	80.000000
max	300.000000	159.000000
count	169.000000	169.000000
mean	63.846154	107.461538

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

```
# Filtered data with Calories value in between 500 - 1000
df[(df["Calories"] >  500) & (df["Calories"]<1000)]
```

	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 data for Calories greater than 500 and Pulse Less than 100
df[(df["Calories"] >  500) & (df["Pulse"]<100)]
```

22] ✓ 0.1s

	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”.

```
# Created another dataframe my selecting only Duration, Pulse and Calories
df_modified = df[["Duration", "Pulse","Calories"]]
```

+ Code

```
# Printing modified data
df_modified
```

	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

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

```
# Dropped Maxpulse column from original data using drop() method
df.drop(columns=["Maxpulse"])
```

	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.

```
df.dtypes # printing Data types of columns
```

```
Duration      int64
Pulse         int64
Maxpulse      int64
Calories    float64
dtype: object
```

```
df['Calories'] = df["Calories"].astype(int) # Changing Datatype of Calories column to int datatype
df.dtypes
```

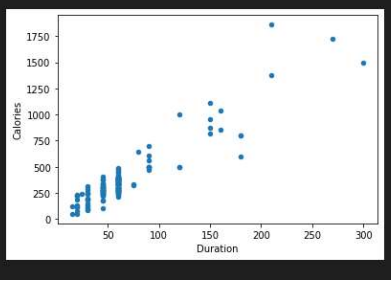
✓ 0.2s

```
Duration    int64
Pulse       int64
Maxpulse    int64
Calories    int32
dtype: object
```

10. Using pandas create a scatter plot for the two columns (Duration and Calories).

```
df.plot.scatter(x="Duration", y="Calories") # Drawing Scatter plot using Pandas plot.scatter() method
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1cdf9b2610>
```



Question 3:

1. Write a Python programming to create a below chart of the popularity of programming Languages.

2. Sample data: Programming languages: Java, Python, PHP, JavaScript, C#, C++ Popularity: 22.2, 17.6, 8.8, 8, 7.7, 6.7

```python
# Plotting a Pie Chart using Matplotlib module for Top programming lamguages.
import matplotlib.pyplot as plt
# Data to plot
languages = 'Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++'
popurarity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]
colors = ["#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#9467bd", "#8c564b"]
explode = (0.1, 0.1, 0.05, 0,0.07,0)
plt.pie(popurarity, explode=explode, labels=languages, colors=colors, autopct='%1.1f%%', shadow=True, startangle=140)
plt.axis('equal')
plt.show()
```

✓ 0.3s