

Lab 1 (Part A)

[1] At first, please find the required dataset for this lab named as 'housing.data1.txt'. You need to upload the dataset to your Google Colab environment first. Then import the required packages like below.

```
import pandas as pd
import numpy as np
```

[2] Import your dataset using pandas and Create a dataframe

```
dataset = pd.read_csv('housing.data1.txt', header = None, sep = '\s+' )
```

[3] Convert your dataset to csv file

[4] Then use the `print()` function to generate output. It should look like below:

	0	1	2	3	4	...	9	10	11	12	13
0	0.00632	18.0	2.310	0.000	0.5380	...	296.00	15.3	396.90	4.98	24.0
1	0.00000	0.0	0.469	6.421	4.9671	...	9.14	21.6	NaN	NaN	NaN
2	0.02729	0.0	7.070	0.000	0.4690	...	242.00	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.180	0.000	0.4580	...	222.00	18.7	394.63	2.94	33.4
4	0.06905	0.0	0.000	0.458	7.1470	...	18.70	396.9	5.33	36.20	NaN
..
501	0.06263	0.0	11.930	0.000	0.5730	...	273.00	21.0	391.99	9.67	22.4
502	0.04527	0.0	20.600	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
503	0.06076	0.0	11.930	0.000	0.5730	...	273.00	21.0	396.90	5.64	23.9
504	0.10959	0.0	11.930	0.000	0.5730	...	273.00	21.0	393.45	6.48	22.0
505	0.04741	0.0	11.930	0.000	0.5730	...	273.00	21.0	396.90	7.88	11.9

[506 rows x 14 columns]

[5] Learn about `head()` function and the purpose of it. Use the `head()` function to produce the output below:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0.00632	18.0	2.310	0.000	0.5380	6.575	65.2000	4.0900	1.0	296.00	15.3	396.90	4.98	24.0
1	0.00000	0.0	0.469	6.421	4.9671	2.000	242.0000	17.8000	396.9	9.14	21.6	NaN	NaN	NaN
2	0.02729	0.0	7.070	0.000	0.4690	7.185	61.1000	4.9671	2.0	242.00	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.180	0.000	0.4580	6.998	45.8000	6.0622	3.0	222.00	18.7	394.63	2.94	33.4
4	0.06905	0.0	0.000	0.458	7.1470	54.200	6.0622	3.0000	222.0	18.70	396.9	5.33	36.20	NaN

[6] As this is a large dataset, therefore we want to find out about missing data by using code. Find out how you can find about if there is NaN in your dataset. I used a variable named check_nan to store my result. If you print the variable, then you should get a True as result. That means you have NaN in your dataset.


```
[ ] print(check_nan)
```

 True

[7] Next, replace the NaN values with mean value (by using SimpleImputer class, module 04).

[8] Check if you still have NaN values in your dataset or not. Recall that your dataset is a numpy array now as you used the SimpleImputer class. It should produce this output below:

```
print(array_has_nan)
```

 False

[9] Print your output and see if it's a numpy array. Your output should look like this below:

```
[[ 6.32000000e-03  1.80000000e+01  2.31000000e+00 ...  3.96900000e+02
   4.98000000e+00  2.40000000e+01]
 [ 0.00000000e+00  0.00000000e+00  4.69000000e-01 ...  3.49281783e+02
   1.29052049e+01  2.25598739e+01]
 [ 2.72900000e-02  0.00000000e+00  7.07000000e+00 ...  3.92830000e+02
   4.03000000e+00  3.47000000e+01]
 ...
 [ 6.07600000e-02  0.00000000e+00  1.19300000e+01 ...  3.96900000e+02
   5.64000000e+00  2.39000000e+01]
 [ 1.09590000e-01  0.00000000e+00  1.19300000e+01 ...  3.93450000e+02
   6.48000000e+00  2.20000000e+01]
 [ 4.74100000e-02  0.00000000e+00  1.19300000e+01 ...  3.96900000e+02
   7.88000000e+00  1.19000000e+01]]
```

[10] Once you are done, please save your .ipynb file as abc123_1_A.ipynb.

Lab 1- Part B

It's important to visualize the important characteristics of your dataset. It may help us to visually detect the presence of outliers, the distribution of the data, and the relationships between features. In this lab, you need to plot five columns from the dataset: **LSTAT**, **INDUS**, **NOX**, **RM**, and **MEDV**. However, you are encouraged to create a scatterplot matrix of the whole `DataFrame` to further explore the data. Please try to identify any linear relationship. You can assume the last column is our target variable/dependent variable that we are trying to predict based on the features (all other columns).

Note: For this part, you should be using `housing.data2.txt`, not the `housing.data1.txt`.

[1] import these packages

```
import pandas as pd
import matplotlib.pyplot as plt
from mlxtend.plotting import scatterplotmatrix
import numpy as np
```

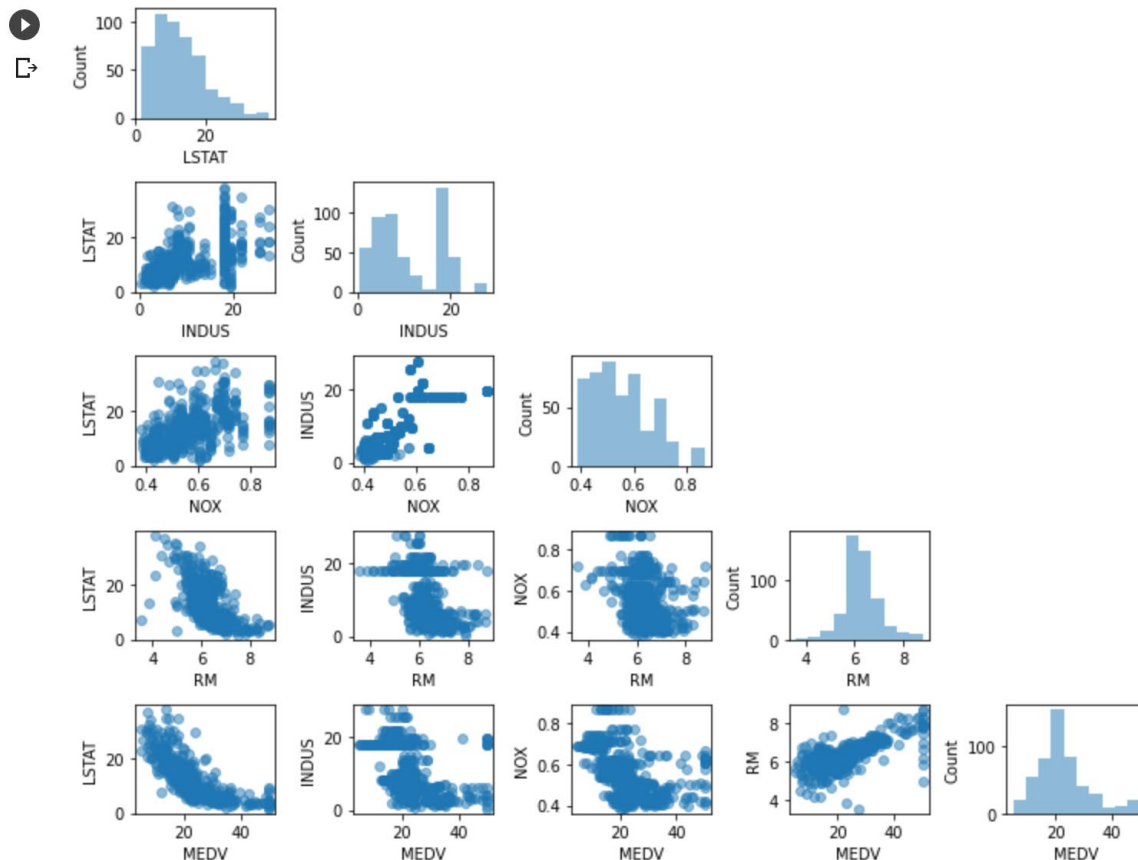
[2] Before you generate the plots, import the dataset using pandas and create a dataframe

```
dataset = pd.read_csv('housing.data2.txt', header=None, sep='\s+')
```

[3] As you notice, the dataset does not have any column names. Assign the column names (you can find out the column names from below description of dataset) to get a clear idea about the features. Use the `head()` function to produce the output below:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.33	36.2

[4] Use the `scatterplotmatrix()` function to generate the output like below. Also, write a comment (create a Text Cell in your ipynb file) if you notice any linear relationship between columns.



See the link below for scatterplotmatrix

http://rasbt.github.io/mlxtend/user_guide/plotting/scatterplotmatrix/

Moreover, if you want to use other packages/functions to generate your plots, you are welcome to do that (no penalty).

[5] Once you are done, please save your .ipynb file as abc123_1_B.ipynb.

What to Submit?

You need to submit a zipped file containing your both .ipynb files. The name of the zipped file should be abc123_lab1

Caution

Please don't change anything in the dataset files. The TA will use the same dataset files to grade your work.

Dataset Info:

the **Housing Dataset**, which contains information about houses in the suburbs of Boston collected by D. Harrison and D.L. Rubinfeld in 1978. We made few changes though.

The features of the samples may be summarized as shown in the excerpt of the dataset description:

CRIM: This is the per capita crime rate by town

ZN: This is the proportion of residential land zoned for lots larger than 25,000 sq.ft.

INDUS: This is the proportion of non-retail business acres per town

CHAS: This is the Charles River dummy variable (this is equal to 1 if tract bounds river; 0 otherwise)

NOX: This is the nitric oxides concentration (parts per 10 million)

RM: This is the average number of rooms per dwelling

AGE: This is the proportion of owner-occupied units built prior to 1940

DIS: This is the weighted distances to five Boston employment centers

RAD: This is the index of accessibility to radial highways

TAX: This is the full-value property-tax rate per \$10,000

PTRATIO: This is the pupil-teacher ratio by town

B: This is calculated as $1000(B_k - 0.63)^2$, where B_k is the proportion of people of African American descent by town

LSTAT: This is the percentage lower status of the population

MEDV: This is the median value of owner-occupied homes in \$1000s