American International University – Bangladesh
Department of Computer Science & Engineering

Project Title: Apply Data Pre-processing on a Dataset
Course: Introduction to Data Science

Submitted by-	**Submitted to-**
Name: Pritom Debnath ID: 20-42414-1 Section: B B.Sc. CSE	Name: Dr. Akinul Islam Jony

Project Overview:

Data pre-processing is a phase in the data analysis process that takes raw data and converts it into a clean format that computers and machine learning can understand and analyse. Raw data in the real world is a jumbled mess. It may include contradictions and inaccuracies. It must be cleaned before it may be used for the intended purpose. The information in this project offers statistics in arrests per 100,000 residents for assault and murder in each of the 50 United States in 1973. The percentage of the people residing in cities is also provided. The dataset, as we can see, is not in clean format. Before it can be used, the dataset must be pre-processed and cleaned.

Project Solution Design:

The dataset shows that there is a missing value (null) in the Assault column. As a result, we must deal with the missing value. Because the Assault column's data type is numeric, substituting in the missing value with mean (average) might be an acceptable choice.

In addition, the Urban population (%) column has corrupt data. Because the Urban population (%) column shows the fraction of the population that lives in cities, it cannot be greater than 100 or less than 0. Yet, there is data in Iowa state where the Urban population (%) score is 570, indicating that there may be a larger problem. This issue might be caused by malfunctioning data gathering devices, data input issues, or technological restrictions. To deal with this faulty data, we must smooth it by removing the final digit (s).

We must separate the percentage of the population living in urban areas into Population_level column in four groups during data pre-processing. Those are less than 50% (small), less than 60% (medium), less than 70% (large) and 70% and above (extra-large)

As Polulation_level is not an ordered factor variable, that’s why it should be a better choice to add an ordered factor variable in the dataset. So, ordered_factor_population column is added.

Polulation_level	Ordered_factor_population
Small	1
Medium	2
Large	3
Extra-large	4

So, at the end of data discretization stage, two new column named type will be integrated into the dataset based on above conditions.

Data pre-processing:

I. Importing the Dataset:

The data is saved in the working directory in the dataset.csv file. To begin pre-processing data in R, we must first import the dataset. Importing the dataset in R code –

```
dataset<-read.csv("dataset.csv")
print(dataset)
```

After importing the dataset, the dataset.csv converts into R dataframe and it is stored in dataset variable. After printing the dataset variable, it looks like this-

	States	Murder	Assault	Urban.population....
1	Alabama	13.2	236	58
2	Alaska	10	263	48
3	Arizona	8.1	294	80
4	Arkansas	8.8	190	50
5	California	9	276	91
6	Colorado	7.9	204	78
7	Connecticut	3.3	110	77
8	Delaware	5.9	238	72
9	Florida	15.4	335	80
10	Georgia	17.4	NA	60
11	Hawaii	5.3	46	83
12	Idaho	2.6	120	54
13	Illinois	10.4	249	83
14	Indiana	7.2	113	65
15	Iowa	2.2	56	570
16	Kansas	6	115	66
17	Kentucky	9.7	109	52
18	Louisiana	15.4	249	66
19	Maine	2.1	83	51
20	Maryland	11.3	300	67
21	Massachusetts	4.4	149	85
22	Michigan	12.1	255	74
23	Minnesota	2.7	72	66
24	Mississippi	16.1	259	44
25	Missouri	9	178	70
26	Montana	6	109	53
27	Nebraska	4.3	102	62
28	Nevada	12.2	252	81
29	New Hampshire	2.1	57	56
30	New Jersey	7.4	159	89
31	New Mexico	11.4	285	70
32	New York	1 1.1	254	6
33	North Carolina	13	337	45
34	North Dakota	0.8	45	44
35	Ohio	7.3	120	75
36	Oklahoma	6.6	151	68
37	Oregon	4.9	159	67
38	Pennsylvania	6.3	106	72
39	Rhode Island	3.4	174	87
40	South Carolina	14.4	879	48
41	South Dakota	3.8	86	45
42	Tennessee	13.2	188	59
43	Texas	12.7	201	80
44	Utah	3.2	120	80

Fig-1: Unprocessed dataset

II. Dealing with Missing Values:

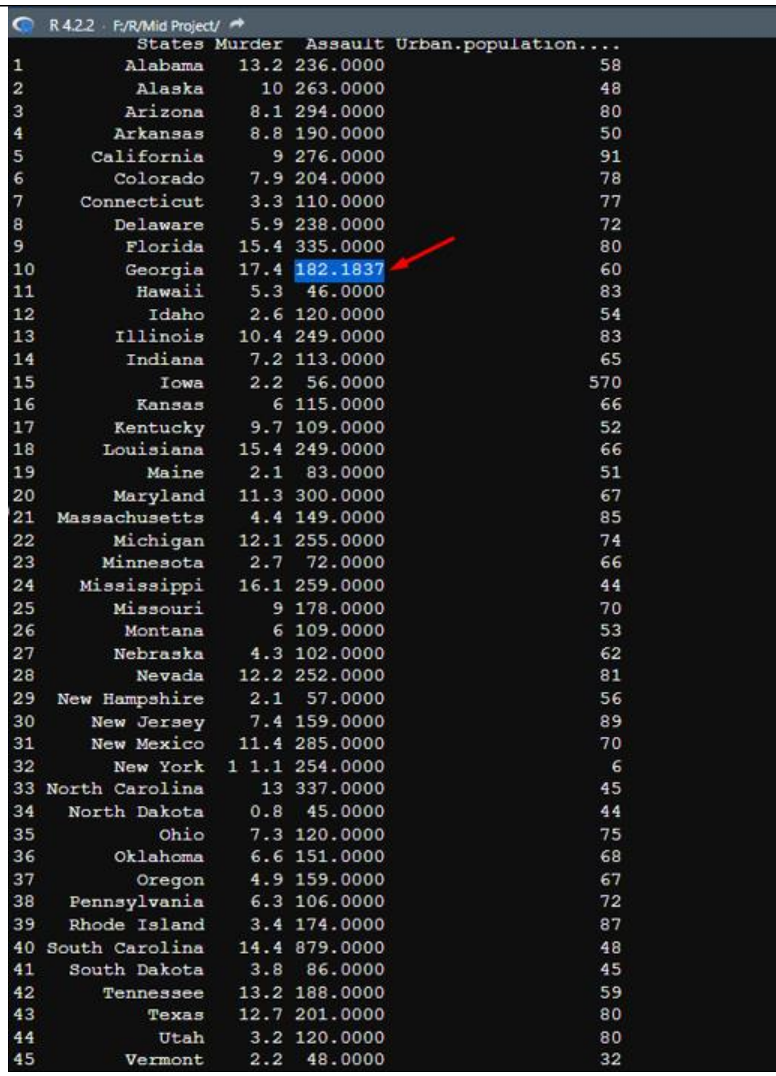
According to the dataset, there is a missing value (NA) in the Assault column. We can replace the missing value with the Assault column's mean value. R code for replacing missing value by the mean-

```
dataset$Assault <- ifelse(is.na(dataset$Assault),
                          mean(dataset$Assault,
                               na.rm = TRUE),dataset$Assault)
print(dataset)
```

Here in the code:

is.na(dataset$Assault)	Returns true for all the cells in the specified column with no values.
mean(dataset$Assault, na.rm = TRUE)	Returns the average of the column passed as argument.
na.rm= TRUE	Calculates the mean excluding the null value.

Now, the dataset looks like this-



	States	Murder	Assault	Urban.population....
1	Alabama	13.2	236.0000	58
2	Alaska	10	263.0000	48
3	Arizona	8.1	294.0000	80
4	Arkansas	8.8	190.0000	50
5	California	9	276.0000	91
6	Colorado	7.9	204.0000	78
7	Connecticut	3.3	110.0000	77
8	Delaware	5.9	238.0000	72
9	Florida	15.4	335.0000	80
10	Georgia	17.4	182.1837	60
11	Hawaii	5.3	46.0000	83
12	Idaho	2.6	120.0000	54
13	Illinois	10.4	249.0000	83
14	Indiana	7.2	113.0000	65
15	Iowa	2.2	56.0000	570
16	Kansas	6	115.0000	66
17	Kentucky	9.7	109.0000	52
18	Louisiana	15.4	249.0000	66
19	Maine	2.1	83.0000	51
20	Maryland	11.3	300.0000	67
21	Massachusetts	4.4	149.0000	85
22	Michigan	12.1	255.0000	74
23	Minnesota	2.7	72.0000	66
24	Mississippi	16.1	259.0000	44
25	Missouri	9	178.0000	70
26	Montana	6	109.0000	53
27	Nebraska	4.3	102.0000	62
28	Nevada	12.2	252.0000	81
29	New Hampshire	2.1	57.0000	56
30	New Jersey	7.4	159.0000	89
31	New Mexico	11.4	285.0000	70
32	New York	1 1.1	254.0000	6
33	North Carolina	13	337.0000	45
34	North Dakota	0.8	45.0000	44
35	Ohio	7.3	120.0000	75
36	Oklahoma	6.6	151.0000	68
37	Oregon	4.9	159.0000	67
38	Pennsylvania	6.3	106.0000	72
39	Rhode Island	3.4	174.0000	87
40	South Carolina	14.4	879.0000	48
41	South Dakota	3.8	86.0000	45
42	Tennessee	13.2	188.0000	59
43	Texas	12.7	201.0000	80
44	Utah	3.2	120.0000	80
45	Vermont	2.2	48.0000	32

Fig:2 – Removed NA value

III. <u>**Dealing with Data Formats:**</u>

After dealing with null values in the Assault column, we can see that the Assault variable has decimal places in the data. Because we don't want decimal places in the Assault column, we'll round it up. We can round Assault variable by the following R code-

```
dataset$Assault <- as.numeric(format(round(dataset$Assault, 0)))
print(dataset)
```

Here, the argument 0 in the round function means no decimal places. Now, the dataset looks like this-

	States	Murder	Assault	Urban.population....
1	Alabama	13.2	236	58
2	Alaska	10	263	48
3	Arizona	8.1	294	80
4	Arkansas	8.8	190	50
5	California	9	276	91
6	Colorado	7.9	204	78
7	Connecticut	3.3	110	77
8	Delaware	5.9	238	72
9	Florida	15.4	335	80
10	Georgia	17.4	182	60
11	Hawaii	5.3	46	83
12	Idaho	2.6	120	54
13	Illinois	10.4	249	83
14	Indiana	7.2	113	65
15	Iowa	2.2	56	57
16	Kansas	6	115	66
17	Kentucky	9.7	109	52
18	Louisiana	15.4	249	66
19	Maine	2.1	83	51
20	Maryland	11.3	300	67
21	Massachusetts	4.4	149	85
22	Michigan	12.1	255	74
23	Minnesota	2.7	72	66
24	Mississippi	16.1	259	44
25	Missouri	9	178	70
26	Montana	6	109	53
27	Nebraska	4.3	102	62
28	Nevada	12.2	252	81
29	New Hampshire	2.1	57	56
30	New Jersey	7.4	159	89
31	New Mexico	11.4	285	70
32	New York	1 1.1	254	6
33	North Carolina	13	337	45
34	North Dakota	0.8	45	44
35	Ohio	7.3	120	75
36	Oklahoma	6.6	151	68
37	Oregon	4.9	159	67
38	Pennsylvania	6.3	106	72
39	Rhode Island	3.4	174	87
40	South Carolina	14.4	879	48
41	South Dakota	3.8	86	45
42	Tennessee	13.2	188	59
43	Texas	12.7	201	80
44	Utah	3.2	120	80
45	Vermont	2.2	48	32
46	Virginia	8.5	156	63
47	Washington	4	145	73

Fig.3- Assault column is rounded up

IV. **Smooth Noisy Data:**

We can see that, there is noisy data present in Urban population (%) column which is 570. As this column represents percentage, so it must be between 0 to 100. We need to smooth the noisy data. R code for smoothing this noisy data-

```r
fix_UrbanPopulation <- function(df){
  i=1
  for(data in df){
    while(data>100){
      data <- data/10
    }
    df[i] <- data
    i <- i+1
  }
  return (df)
}
```

Here, the fix_UrbanPopulation(df) function fix the data range (0 to 100) we divide each data by 10 continuously, while it is greater than 100. Now the dataset looks like this-

	States	Murder	Assault	Urban.population....
1	Alabama	13.2	236	58
2	Alaska	10	263	48
3	Arizona	8.1	294	80
4	Arkansas	8.8	190	50
5	California	9	276	91
6	Colorado	7.9	204	78
7	Connecticut	3.3	110	77
8	Delaware	5.9	238	72
9	Florida	15.4	335	80
10	Georgia	17.4	182	60
11	Hawaii	5.3	46	83
12	Idaho	2.6	120	54
13	Illinois	10.4	249	83
14	Indiana	7.2	113	65
15	Iowa	2.2	56	57
16	Kansas	6	115	66
17	Kentucky	9.7	109	52
18	Louisiana	15.4	249	66
19	Maine	2.1	83	51
20	Maryland	11.3	300	67
21	Massachusetts	4.4	149	85
22	Michigan	12.1	255	74
23	Minnesota	2.7	72	66
24	Mississippi	16.1	259	44
25	Missouri	9	178	70
26	Montana	6	109	53
27	Nebraska	4.3	102	62
28	Nevada	12.2	252	81
29	New Hampshire	2.1	57	56
30	New Jersey	7.4	159	89
31	New Mexico	11.4	285	70
32	New York	1 1.1	254	6
33	North Carolina	13	337	45
34	North Dakota	0.8	45	44
35	Ohio	7.3	120	75
36	Oklahoma	6.6	151	68
37	Oregon	4.9	159	67
38	Pennsylvania	6.3	106	72
39	Rhode Island	3.4	174	87
40	South Carolina	14.4	879	48
41	South Dakota	3.8	86	45
42	Tennessee	13.2	188	59
43	Texas	12.7	201	80
44	Utah	3.2	120	80
45	Vermont	2.2	48	32
46	Virginia	8.5	156	63
47	Washington	4	145	73

Fig.4 – Noise free Urban Population column

V. Data Transformation:

Smoothing, noise removal from data, summarization, generalization, and normalization are all part of the data transformation process. Smoothing, which we studied in IV, will be used in this case (Smooth Noisy Data).

VI. Data Reduction:

This dataset does not involve any data reduction steps.

VII. Data Discretization and Data Integration:

We frequently work with data that is gathered through continuous procedures. But there are situations when it's necessary to split up these continuous numbers into smaller chunks. Discrete mapping is the term for this process. As you can see, every attribute in our dataset is of the continuous type. Using logic, we may discretize the data into category kinds and include the column into our dataset. R code for this step-

```
dataset$Urban.population.... <- fix_UrbanPopulation(dataset$Urban.population....)
print(dataset)

dataset$Polpulation_level<- with(dataset, ifelse(dataset$Urban.population.... < 50, 'small',
                                         ifelse(dataset$Urban.population.... < 60, 'medium',
                                         ifelse(dataset$Urban.population.... < 70, 'large','extra-large'))))

dataset$Ordered_factor_population <- with(dataset, ifelse(dataset$Polpulation_level == 'small', 1,
                                  ifelse(dataset$Polpulation_level == 'medium', 2,
                                  ifelse(Polpulation_level == 'large', 3,
                                  4))))
```

Here, the with () function take two parameters. One is dataframe, another one is expression. with () function integrates a new column in the dataframe based on the expression. As we have two columns to add for each column we used with () function.

After integrating new column, the dataset looks like this-

	States	Murder	Assault	Urban.population....	Polpulation_level	Ordered_factor_population
1	Alabama	13.2	236	58	medium	2
2	Alaska	10	263	48	small	1
3	Arizona	8.1	294	80	extra-large	4
4	Arkansas	8.8	190	50	medium	2
5	California	9	276	91	extra-large	4
6	Colorado	7.9	204	78	extra-large	4
7	Connecticut	3.3	110	77	extra-large	4
8	Delaware	5.9	238	72	extra-large	4
9	Florida	15.4	335	80	extra-large	4
10	Georgia	17.4	182	60	large	3
11	Hawaii	5.3	46	83	extra-large	4
12	Idaho	2.6	120	54	medium	2
13	Illinois	10.4	249	83	extra-large	4
14	Indiana	7.2	113	65	large	3
15	Iowa	2.2	56	57	medium	2
16	Kansas	6	115	66	large	3
17	Kentucky	9.7	109	52	medium	2
18	Louisiana	15.4	249	66	large	3
19	Maine	2.1	83	51	medium	2
20	Maryland	11.3	300	67	large	3
21	Massachusetts	4.4	149	85	extra-large	4
22	Michigan	12.1	255	74	extra-large	4
23	Minnesota	2.7	72	66	large	3
24	Mississippi	16.1	259	44	small	1
25	Missouri	9	178	70	extra-large	4
26	Montana	6	109	53	medium	2
27	Nebraska	4.3	102	62	large	3
28	Nevada	12.2	252	81	extra-large	4
29	New Hampshire	2.1	57	56	medium	2
30	New Jersey	7.4	159	89	extra-large	4
31	New Mexico	11.4	285	70	extra-large	4
32	New York	1 1.1	254	6	small	1
33	North Carolina	13	337	45	small	1
34	North Dakota	0.8	45	44	small	1
35	Ohio	7.3	120	75	extra-large	4
36	Oklahoma	6.6	151	68	large	3
37	Oregon	4.9	159	67	large	3
38	Pennsylvania	6.3	106	72	extra-large	4
39	Rhode Island	3.4	174	87	extra-large	4
40	South Carolina	14.4	879	48	small	1
41	South Dakota	3.8	86	45	small	1
42	Tennessee	13.2	188	59	medium	2
43	Texas	12.7	201	80	extra-large	4
44	Utah	3.2	120	80	extra-large	4
45	Vermont	2.2	48	32	small	1
46	Virginia	8.5	156	63	large	3
47	Washington	4	145	73	extra-large	4
48	West Virginia	5.7	81	39	small	1
49	Wisconsin	2.6	53	66	large	3
50	Wyoming	6.8	161	60	large	3

Fig.5 – Adding two rows

Discussion & Conclusion:

At the beginning of the project, we were given a dataset which was totally messy. Null value, noisy data was present in this dataset in Fig.1. After pre-processing the dataset and integrating new column in the dataset, we got totally a clean dataset. The dataset looks like this-

R 4.2.2 · F:/R/Mid Project/

	States	Murder	Assault	Urban.population....	Polpulation_level	Ordered_factor_population
1	Alabama	13.2	236	58	medium	2
2	Alaska	10	263	48	small	1
3	Arizona	8.1	294	80	extra-large	4
4	Arkansas	8.8	190	50	medium	2
5	California	9	276	91	extra-large	4
6	Colorado	7.9	204	78	extra-large	4
7	Connecticut	3.3	110	77	extra-large	4
8	Delaware	5.9	238	72	extra-large	4
9	Florida	15.4	335	80	extra-large	4
10	Georgia	17.4	182	60	large	3
11	Hawaii	5.3	46	83	extra-large	4
12	Idaho	2.6	120	54	medium	2
13	Illinois	10.4	249	83	extra-large	4
14	Indiana	7.2	113	65	large	3
15	Iowa	2.2	56	57	medium	2
16	Kansas	6	115	66	large	3
17	Kentucky	9.7	109	52	medium	2
18	Louisiana	15.4	249	66	large	3
19	Maine	2.1	83	51	medium	2
20	Maryland	11.3	300	67	large	3
21	Massachusetts	4.4	149	85	extra-large	4
22	Michigan	12.1	255	74	extra-large	4
23	Minnesota	2.7	72	66	large	3
24	Mississippi	16.1	259	44	small	1
25	Missouri	9	178	70	extra-large	4
26	Montana	6	109	53	medium	2
27	Nebraska	4.3	102	62	large	3
28	Nevada	12.2	252	81	extra-large	4
29	New Hampshire	2.1	57	56	medium	2
30	New Jersey	7.4	159	89	extra-large	4
31	New Mexico	11.4	285	70	extra-large	4
32	New York	1 1.1	254	6	small	1
33	North Carolina	13	337	45	small	1
34	North Dakota	0.8	45	44	small	1
35	Ohio	7.3	120	75	extra-large	4
36	Oklahoma	6.6	151	68	large	3
37	Oregon	4.9	159	67	large	3
38	Pennsylvania	6.3	106	72	extra-large	4
39	Rhode Island	3.4	174	87	extra-large	4
40	South Carolina	14.4	879	48	small	1
41	South Dakota	3.8	86	45	small	1
42	Tennessee	13.2	188	59	medium	2
43	Texas	12.7	201	80	extra-large	4
44	Utah	3.2	120	80	extra-large	4
45	Vermont	2.2	48	32	small	1
46	Virginia	8.5	156	63	large	3
47	Washington	4	145	73	extra-large	4
48	West Virginia	5.7	81	39	small	1
49	Wisconsin	2.6	53	66	large	3
50	Wyoming	6.8	161	60	large	3

Fig.6 – Dataset after pre-processing

Now, we can use this clean, pre-processed dataset for further use.