# PreProcessingPart2

April 24, 2025

[1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from functions_preprocessing import printing_column, show_invalid_entries,␣
 ↪replacing_invalid
```

[2]:

```
## Reading csv file
df = pd.read_csv("student_depression_dataset.csv")
```

Since **id** column doesn't has any relevance has only unique values we can drop it.

[3]:

```
## Dropping id column and printing head to verify

df = df.drop('id', axis=1)
df.head()
```

[3]:

```
   Gender   Age           City Profession  Academic Pressure  Work Pressure  \
0    Male  33.0  Visakhapatnam    Student                5.0            0.0
1  Female  24.0      Bangalore    Student                2.0            0.0
2    Male  31.0       Srinagar    Student                3.0            0.0
3  Female  28.0       Varanasi    Student                3.0            0.0
4  Female  25.0         Jaipur    Student                4.0            0.0

   CGPA  Study Satisfaction  Job Satisfaction       Sleep Duration  \
0  8.97                 2.0               0.0          '5-6 hours'
1  5.90                 5.0               0.0          '5-6 hours'
2  7.03                 5.0               0.0  'Less than 5 hours'
3  5.59                 2.0               0.0          '7-8 hours'
4  8.13                 3.0               0.0          '5-6 hours'

  Dietary Habits   Degree Have you ever had suicidal thoughts ?  \
0        Healthy  B.Pharm                                   Yes
1       Moderate      BSc                                    No
2        Healthy       BA                                    No
3       Moderate      BCA                                   Yes
4       Moderate   M.Tech                                   Yes
```

```
   Work/Study Hours Financial Stress Family History of Mental Illness  \
0               3.0              1.0                               No
1               3.0              2.0                              Yes
2               9.0              1.0                              Yes
3               4.0              5.0                              Yes
4               1.0              1.0                               No

   Depression
0           1
1           0
2           0
3           1
4           0
```

[4]:

```
## Printing all the unique values in all the columns

printing_column(df)
```

```
Gender unique values:
['Male' 'Female']

Age unique values:
[33. 24. 31. 28. 25. 29. 30. 27. 19. 20. 23. 18. 21. 22. 34. 32. 26. 39.
 35. 42. 36. 58. 49. 38. 51. 44. 43. 46. 59. 54. 48. 56. 37. 41.]

City unique values:
['Visakhapatnam' 'Bangalore' 'Srinagar' 'Varanasi' 'Jaipur' 'Pune' 'Thane'
 'Chennai' 'Nagpur' 'Nashik' 'Vadodara' 'Kalyan' 'Rajkot' 'Ahmedabad'
 'Kolkata' 'Mumbai' 'Lucknow' 'Indore' 'Surat' 'Ludhiana' 'Bhopal'
 'Meerut' 'Agra' 'Ghaziabad' 'Hyderabad' 'Vasai-Virar' 'Kanpur' 'Patna'
 'Faridabad' 'Delhi' 'Saanvi' 'M.Tech' 'Bhavna' "'Less Delhi'" 'City'
 '3.0' "'Less than 5 Kalyan'" 'Mira' 'Harsha' 'Vaanya' 'Gaurav' 'Harsh'
 'Reyansh' 'Kibara' 'Rashi' 'ME' 'M.Com' 'Nalyan' 'Mihir' 'Nalini'
 'Nandini' 'Khaziabad']

Profession unique values:
['Student' "'Civil Engineer'" 'Architect' "'UX/UI Designer'"
 "'Digital Marketer'" "'Content Writer'" "'Educational Consultant'"
 'Teacher' 'Manager' 'Chef' 'Doctor' 'Lawyer' 'Entrepreneur' 'Pharmacist']

Academic Pressure unique values:
[5. 2. 3. 4. 1. 0.]

Work Pressure unique values:
[0. 5. 2.]

CGPA unique values:
```

```
[ 8.97    5.9     7.03    5.59    8.13    5.7     9.54    8.04    9.79
  8.38    6.1     7.04    8.52    5.64    8.58    6.51    7.25    7.83
  9.93    8.74    6.73    5.57    8.59    7.1     6.08    5.74    9.86
  6.7     6.21    5.87    6.37    9.72    5.88    9.56    6.99    5.24
  9.21    7.85    6.95    5.86    7.92    9.66    8.94    9.71    7.87
  5.6     7.9     5.46    6.79    8.7     7.38    8.5     7.09    9.82
  8.89    7.94    9.11    6.75    7.53    9.49    9.01    7.64    5.27
  6.      9.44    5.75    7.51    9.05    6.38    8.95    9.88    5.32
  6.27    7.7     8.1     9.59    8.96    5.51    7.43    8.79    9.95
  5.37    6.86    8.32    9.74    5.66    7.48    8.23    8.81    6.03
  5.56    5.68    5.14    7.61    6.17    8.17    9.87    8.75    6.16
  9.5     7.99    5.67    8.92    6.19    5.76    6.25    5.11    5.58
  5.65    9.89    8.03    6.61    9.41    8.64    7.21    8.28    6.04
  9.13    8.08    9.96    5.12    8.35    7.07    9.6     9.24    8.54
  8.78    8.93    8.91    9.04    6.83    5.85    7.74    6.41    8.9
  7.75    7.88    5.42    7.52    7.68    8.4     9.39    6.84    5.99
  8.62    8.53    7.47    6.78    6.42    9.92    8.39    5.89    7.22
  6.81    9.02    9.97    9.63    9.67    5.41    7.27    6.05    6.85
  9.33    5.81    6.53    5.98    6.02    6.74    5.26    7.72    7.39
  8.43    9.34    5.44    5.82    5.72    8.19    8.44    8.98    9.37
  5.8     7.28    7.6     7.91    9.17    7.46    9.43    9.91    9.36
  5.16    7.08    9.26    8.83   10.      7.8     9.46    6.63    7.24
  6.47    7.77    5.06    7.17    8.24    6.88    9.03    5.08    5.45
  8.46    9.19    6.36    8.73    7.11    9.12    9.4     8.11    9.98
  5.55    8.61    8.14    6.89    9.84    5.48    8.21    7.82    8.55
  5.79    8.77    8.29    6.92    7.37    9.7     6.26    7.26    7.5
  6.82    7.15    5.77    5.91    5.1     7.71    9.06    5.71    5.84
  9.42    6.23    6.29    5.25    9.69    9.9     6.39    8.09    5.83
  5.47    6.56    8.71    9.94    6.69    5.52    7.3     7.02    6.33
  8.07    8.37    8.      7.79    8.65    6.28    7.35    8.69    7.12
  7.32    7.13    5.97    5.09    6.91    6.76    6.52    7.45    8.56
  6.5     8.63    8.27    8.49    6.59    9.29    5.3     7.06    5.38
  6.65    9.16    8.01    8.25    8.02    8.47    7.34    8.88    7.14
  8.42    5.17    9.1     7.49    9.85    7.42    9.31    6.35    7.
  5.39    5.61    9.78    9.25    5.69    9.47    8.16    7.23    6.46
  0.      8.26    6.32    6.77    8.85    5.03    7.65    5.78    6.24
  5.35    6.06    7.78    6.64    7.0625  6.98    6.44    6.09  ]

Study Satisfaction unique values:
[2. 5. 3. 4. 1. 0.]

Job Satisfaction unique values:
[0. 3. 4. 2. 1.]

Sleep Duration unique values:
["'5-6 hours'" "'Less than 5 hours'" "'7-8 hours'" "'More than 8 hours'"
 'Others']
```

```
Dietary Habits unique values:
['Healthy' 'Moderate' 'Unhealthy' 'Others']

Degree unique values:
['B.Pharm' 'BSc' 'BA' 'BCA' 'M.Tech' 'PhD' "'Class 12'" 'B.Ed' 'LLB' 'BE'
 'M.Ed' 'MSc' 'BHM' 'M.Pharm' 'MCA' 'MA' 'B.Com' 'MD' 'MBA' 'MBBS' 'M.Com'
 'B.Arch' 'LLM' 'B.Tech' 'BBA' 'ME' 'MHM' 'Others']

Have you ever had suicidal thoughts ? unique values:
['Yes' 'No']

Work/Study Hours unique values:
[ 3.  9.  4.  1.  0. 12.  2. 11. 10.  6.  8.  5.  7.]

Financial Stress unique values:
['1.0' '2.0' '5.0' '3.0' '4.0' '?']

Family History of Mental Illness unique values:
['No' 'Yes']

Depression unique values:
[1 0]
```

After printing all the unique values in each column we can visualize that a lot of columns are having incorrect or missing entries. - City: ‘M.Tech’, “ ‘Less Delhi’ ”, ‘City’, ‘3.0’, “ ‘Less than 5 Kalyan’ ”, ‘ME’, ‘M.Com’ - Sleep Duration: ‘Others’ - Dietary Habits: ‘Others’ - Degree: ‘Others’ - Financial Stress: ‘Others’

### 0.0.1 City Column

Dealing with City column first, looking closely we can observe that the city column does have incorrect values which weren’t suppose to be in there. Here we have two values which represent cities “Less than 5 Kalyan” and “Less Delhi”. We can replace them with appropriate values i.e. “Less than 5 Kalyan” with “Kalyan” and “Less Delhi” with “Delhi”.

[5]:

```
invalid_cities = ['M.Tech', "'Less Delhi'", 'City', '3.0', "'Less than 5␣
 ↪Kalyan'", 'ME', 'M.Com']
show_invalid_entries(df, 'City', invalid_cities)
```

```
Number of invalid City entries: 8
City
City                    2
M.Tech                  1
'Less Delhi'            1
3.0                     1
'Less than 5 Kalyan'    1
ME                      1
M.Com                   1
```

```
Name: count, dtype: int64
```

Replacing all values with mode value of City column except “Less Delhi” and “Less than 5 Kalyan” since they will be replaced by the city names.

[6]:
```
## Replaced "Less Delhi" with "Delhi" and "Less than 5 Kalyan"
df['City'] = df['City'].replace("'Less Delhi'", 'Delhi')
df['City'] = df['City'].replace("'Less than 5 Kalyan'", 'Kalyan')
```

[7]:
```
show_invalid_entries(df, 'City', invalid_cities)
```

```
Number of invalid City entries: 6
City
City       2
M.Tech     1
3.0        1
ME         1
M.Com      1
Name: count, dtype: int64
```

[8]:
```
## Replacing rest of the invalid cities with the mode values
df = replacing_invalid(df, 'City', invalid_cities)
```

[9]:
```
show_invalid_entries(df, 'City', invalid_cities)
```

```
Number of invalid City entries: 0
Series([], Name: count, dtype: int64)
```

All invalid values in **City** column are replaced.

---

### 0.0.2 Sleep Duration

Discussed earlier we observe there is an invalid value **“Other”** in the column **Sleep Duration**.

[10]:
```
invalid_sleep = ['Others']
show_invalid_entries(df, "Sleep Duration", invalid_sleep)
```

```
Number of invalid Sleep Duration entries: 18
Sleep Duration
Others    18
Name: count, dtype: int64
```

Removing the **18 other** values with the value which appeared most (mode).

[11]:
```
df = replacing_invalid(df, 'Sleep Duration', invalid_sleep)
```

[12]:
```
show_invalid_entries(df, "Sleep Duration", invalid_sleep)
```

```
Number of invalid Sleep Duration entries: 0
Series([], Name: count, dtype: int64)
```

---

### 0.0.3 Dietary Habits

Discussed earlier we observe there is an invalid value **“Other”** in the column **Dietary Habits**.

[13]:

```
invalid_habit = ['Others']
show_invalid_entries(df, "Dietary Habits", invalid_habit)
```

```
Number of invalid Dietary Habits entries: 12
Dietary Habits
Others    12
Name: count, dtype: int64
```

[14]:

```
df = replacing_invalid(df, 'Dietary Habits', invalid_habit)
```

[15]:

```
show_invalid_entries(df, "Dietary Habits", invalid_habit)
```

```
Number of invalid Dietary Habits entries: 0
Series([], Name: count, dtype: int64)
```

---

### 0.0.4 Degree

Discussed earlier we observe there is an invalid value **“Other”** in the column **Degree**.

[16]:

```
invalid_degree = ['Others']
show_invalid_entries(df, "Degree", invalid_degree)
```

```
Number of invalid Degree entries: 35
Degree
Others    35
Name: count, dtype: int64
```

[17]:

```
df = replacing_invalid(df, 'Degree', invalid_degree)
```

[18]:

```
show_invalid_entries(df, "Degree", invalid_degree)
```

```
Number of invalid Degree entries: 0
Series([], Name: count, dtype: int64)
```

---

### 0.0.5 Financial Stress

Discussed earlier we observe there is an invalid value **“?”** in the column **Financial Stress**.

[19]:

```
invalid_stress = ['?']
show_invalid_entries(df, "Financial Stress", invalid_stress)
```

```
Number of invalid Financial Stress entries: 3
Financial Stress
?    3
Name: count, dtype: int64
```

[20]:

```python
df = replacing_invalid(df, "Financial Stress", invalid_stress)
```

[21]:

```python
show_invalid_entries(df, "Financial Stress", invalid_stress)
```

```
Number of invalid Financial Stress entries: 0
Series([], Name: count, dtype: int64)
```

[22]:

```python
df['Financial Stress'] = df['Financial Stress'].astype(float).astype('int64')
```

---

[ ]:

```python
## Printing all the unique values of the column again; just to make sure that we have not missed anything.
printing_column(df)
```

```
Gender unique values:
['Male' 'Female']

Age unique values:
[33. 24. 31. 28. 25. 29. 30. 27. 19. 20. 23. 18. 21. 22. 34. 32. 26. 39.
 35. 42. 36. 58. 49. 38. 51. 44. 43. 46. 59. 54. 48. 56. 37. 41.]

City unique values:
['Visakhapatnam' 'Bangalore' 'Srinagar' 'Varanasi' 'Jaipur' 'Pune' 'Thane'
 'Chennai' 'Nagpur' 'Nashik' 'Vadodara' 'Kalyan' 'Rajkot' 'Ahmedabad'
 'Kolkata' 'Mumbai' 'Lucknow' 'Indore' 'Surat' 'Ludhiana' 'Bhopal'
 'Meerut' 'Agra' 'Ghaziabad' 'Hyderabad' 'Vasai-Virar' 'Kanpur' 'Patna'
 'Faridabad' 'Delhi' 'Saanvi' 'Bhavna' 'Mira' 'Harsha' 'Vaanya' 'Gaurav'
 'Harsh' 'Reyansh' 'Kibara' 'Rashi' 'Nalyan' 'Mihir' 'Nalini' 'Nandini'
 'Khaziabad']

Profession unique values:
['Student' "'Civil Engineer'" 'Architect' "'UX/UI Designer'"
 "'Digital Marketer'" "'Content Writer'" "'Educational Consultant'"
 'Teacher' 'Manager' 'Chef' 'Doctor' 'Lawyer' 'Entrepreneur' 'Pharmacist']

Academic Pressure unique values:
[5. 2. 3. 4. 1. 0.]

Work Pressure unique values:
[0. 5. 2.]

CGPA unique values:
[ 8.97     5.9      7.03     5.59     8.13     5.7      9.54     8.04     9.79
```

```
  8.38    6.1     7.04    8.52    5.64    8.58    6.51    7.25    7.83
  9.93    8.74    6.73    5.57    8.59    7.1     6.08    5.74    9.86
  6.7     6.21    5.87    6.37    9.72    5.88    9.56    6.99    5.24
  9.21    7.85    6.95    5.86    7.92    9.66    8.94    9.71    7.87
  5.6     7.9     5.46    6.79    8.7     7.38    8.5     7.09    9.82
  8.89    7.94    9.11    6.75    7.53    9.49    9.01    7.64    5.27
  6.      9.44    5.75    7.51    9.05    6.38    8.95    9.88    5.32
  6.27    7.7     8.1     9.59    8.96    5.51    7.43    8.79    9.95
  5.37    6.86    8.32    9.74    5.66    7.48    8.23    8.81    6.03
  5.56    5.68    5.14    7.61    6.17    8.17    9.87    8.75    6.16
  9.5     7.99    5.67    8.92    6.19    5.76    6.25    5.11    5.58
  5.65    9.89    8.03    6.61    9.41    8.64    7.21    8.28    6.04
  9.13    8.08    9.96    5.12    8.35    7.07    9.6     9.24    8.54
  8.78    8.93    8.91    9.04    6.83    5.85    7.74    6.41    8.9
  7.75    7.88    5.42    7.52    7.68    8.4     9.39    6.84    5.99
  8.62    8.53    7.47    6.78    6.42    9.92    8.39    5.89    7.22
  6.81    9.02    9.97    9.63    9.67    5.41    7.27    6.05    6.85
  9.33    5.81    6.53    5.98    6.02    6.74    5.26    7.72    7.39
  8.43    9.34    5.44    5.82    5.72    8.19    8.44    8.98    9.37
  5.8     7.28    7.6     7.91    9.17    7.46    9.43    9.91    9.36
  5.16    7.08    9.26    8.83   10.      7.8     9.46    6.63    7.24
  6.47    7.77    5.06    7.17    8.24    6.88    9.03    5.08    5.45
  8.46    9.19    6.36    8.73    7.11    9.12    9.4     8.11    9.98
  5.55    8.61    8.14    6.89    9.84    5.48    8.21    7.82    8.55
  5.79    8.77    8.29    6.92    7.37    9.7     6.26    7.26    7.5
  6.82    7.15    5.77    5.91    5.1     7.71    9.06    5.71    5.84
  9.42    6.23    6.29    5.25    9.69    9.9     6.39    8.09    5.83
  5.47    6.56    8.71    9.94    6.69    5.52    7.3     7.02    6.33
  8.07    8.37    8.      7.79    8.65    6.28    7.35    8.69    7.12
  7.32    7.13    5.97    5.09    6.91    6.76    6.52    7.45    8.56
  6.5     8.63    8.27    8.49    6.59    9.29    5.3     7.06    5.38
  6.65    9.16    8.01    8.25    8.02    8.47    7.34    8.88    7.14
  8.42    5.17    9.1     7.49    9.85    7.42    9.31    6.35    7.
  5.39    5.61    9.78    9.25    5.69    9.47    8.16    7.23    6.46
  0.      8.26    6.32    6.77    8.85    5.03    7.65    5.78    6.24
  5.35    6.06    7.78    6.64    7.0625  6.98    6.44    6.09  ]

Study Satisfaction unique values:
[2. 5. 3. 4. 1. 0.]

Job Satisfaction unique values:
[0. 3. 4. 2. 1.]

Sleep Duration unique values:
["'5-6 hours'" "'Less than 5 hours'" "'7-8 hours'" "'More than 8 hours'"]

Dietary Habits unique values:
['Healthy' 'Moderate' 'Unhealthy']
```

```
Degree unique values:
['B.Pharm' 'BSc' 'BA' 'BCA' 'M.Tech' 'PhD' "'Class 12'" 'B.Ed' 'LLB' 'BE'
 'M.Ed' 'MSc' 'BHM' 'M.Pharm' 'MCA' 'MA' 'B.Com' 'MD' 'MBA' 'MBBS' 'M.Com'
 'B.Arch' 'LLM' 'B.Tech' 'BBA' 'ME' 'MHM']

Have you ever had suicidal thoughts ? unique values:
['Yes' 'No']

Work/Study Hours unique values:
[ 3.  9.  4.  1.  0. 12.  2. 11. 10.  6.  8.  5.  7.]

Financial Stress unique values:
[1 2 5 3 4]

Family History of Mental Illness unique values:
['No' 'Yes']

Depression unique values:
[1 0]
```

[ ]:

```
## Printing the types again to make sure we have the correct types
print(df.dtypes)
```

```
Gender                                    object
Age                                      float64
City                                      object
Profession                                object
Academic Pressure                        float64
Work Pressure                            float64
CGPA                                     float64
Study Satisfaction                       float64
Job Satisfaction                         float64
Sleep Duration                            object
Dietary Habits                            object
Degree                                    object
Have you ever had suicidal thoughts ?     object
Work/Study Hours                         float64
Financial Stress                           int64
Family History of Mental Illness          object
Depression                                 int64
dtype: object
```

[26]:

```
## Age, Academic Pressure, Work Pressure, Study Satisfaction, Job Satisfaction,␣
 ↪Work/Study Hours
## Converting these columns which from float to int64 since there are no␣
 ↪decimal values
```

```
cols_to_convert = ['Age', 'Academic Pressure', 'Work Pressure', 'Study␣
 ↪Satisfaction', 'Job Satisfaction', 'Work/Study Hours']
df[cols_to_convert] = df[cols_to_convert].astype(float).astype('int64')
```

[27]:
```
print(df.dtypes)
```

```
Gender                                    object
Age                                        int64
City                                      object
Profession                                object
Academic Pressure                          int64
Work Pressure                              int64
CGPA                                     float64
Study Satisfaction                         int64
Job Satisfaction                           int64
Sleep Duration                            object
Dietary Habits                            object
Degree                                    object
Have you ever had suicidal thoughts ?     object
Work/Study Hours                           int64
Financial Stress                           int64
Family History of Mental Illness          object
Depression                                 int64
dtype: object
```

[28]:
```
df.to_csv("Final.csv")
```