-------------------------------------- Workshop #2 ------------------------------------

- This workshop includes marked tasks that comprise 16% of your final mark in this module.
- To complete the tasks, you need to apply data preprocessing methods discussed in Lecture 2. However, you may need to research to find solutions to some tasks.
- You can duplicate the code and report cells if you need more than one code/report cell for your solution

Tasks

TASK 2.1: Download the adult dataset from Canvas, import it into Jupyter Notebook, and complete the following steps (Completing the report cell is required):

a. Write a code to show how many Null values each column contains. Report the columns that contain Null values in the report cell (Hint: use the isna().sum() method to show the number of the Null values) (NOTE: Nan values are also considered as Null values) (1%).

b. Change the display setting and print the first 100 rows. Find columns that contain wrong data and report them in the report cell (1%).

c. We cannot inspect the entire data by eyes for columns containing wrong data. Explain in the report cell what method we could use to find all columns which contain wrong data (2%)

In [37]:
```
############# WRITE YOUR CODE IN THIS CELL (IF APPLICABLE)###################
#importing the relevant libraries
import pandas as pd
import numpy as np

#Reading csv file into my dataframe (data)
data = pd.read_csv('C:/Users/HP/Downloads/adult.csv')

#Printing
data
```

Out[37]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black
...	...	...	...	...	...	...	...	...	...
48837	39	Private	215419	Bachelors	13	Divorced	Prof-specialty	Not-in-family	White
48838	64	NaN	321403	HS-grad	9	Widowed	NaN	Other-relative	Black
48839	38	Private	374983	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	White
48840	44	Private	83891	Bachelors	13	Divorced	Adm-clerical	Own-child	Asian-Pac-Islander
48841	35	Self-emp-inc	182148	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White

48842 rows × 15 columns

In [38]:
```
#printing data info
print(data.info())

#checking data shape/make up
data.shape
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48842 entries, 0 to 48841
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   age             48842 non-null  int64 
 1   workclass       47879 non-null  object
 2   fnlwgt          48842 non-null  int64 
 3   education       48842 non-null  object
 4   education-num   48842 non-null  int64 
 5   marital-status  48842 non-null  object
 6   occupation      47876 non-null  object
 7   relationship    48842 non-null  object
 8   race            48842 non-null  object
 9   sex             48842 non-null  object
 10  capital-gain    48842 non-null  int64 
 11  capital-loss    48842 non-null  int64 
 12  hours-per-week  48842 non-null  int64 
 13  native-country  48568 non-null  object
 14  income          48842 non-null  object
dtypes: int64(6), object(9)
memory usage: 5.6+ MB
None
```

Out[38]:
```
(48842, 15)
```

In [39]:
```
# Counting the number of null values in each column
null_count_per_column = data.isna().sum()

#printing outcome
print("Number of null values in each column:")
print(null_count_per_column)
```

```
Number of null values in each column:
age                 0
workclass         963
fnlwgt              0
education           0
education-num       0
marital-status      0
occupation        966
relationship        0
race                0
sex                 0
capital-gain        0
capital-loss        0
hours-per-week      0
native-country    274
income              0
dtype: int64
```

```
############# WRITE YOUR REPORT IN THIS CELL (IF
APPLICABLE)####################

using the isna().sum() method to show the number of the Null values in the
dataset by columns,only three 3 columns out of the 14 columns showed a record
of null values as follows; workclass column has 963 null values, occupation
column has 966 null values, while native-country column has 274 null values
(53 WORDS)
```

In [40]:
```python
# Check for null values in each column
columns_with_null = data.columns[data.isna().any()].tolist()

#printing outcome
print("Columns containing null values:")
print(columns_with_null)
```

```
Columns containing null values:
['workclass', 'occupation', 'native-country']
```

In [41]:
```python
#changing the display setting to print the first 100 rows
pd.set_option('display.max_rows', 120)
pd.set_option('display.max_column', 80)
pd.set_option('display.width', 100)

#printing outcome
print(data.head(100))
```

```
    age         workclass  fnlwgt     education  education-num          mari
tal-status  \
0    39         State-gov   77516     Bachelors             13           Nev
er-married
1    50  Self-emp-not-inc   83311     Bachelors             13      Married-
civ-spouse
2    38           Private  215646       HS-grad              9
Divorced
3    53           Private  234721          11th              7      Married-
civ-spouse
4    28           Private  338409     Bachelors             13      Married-
civ-spouse
5    37           Private  284582       Masters             14      Married-
civ-spouse
6    49           Private  160187           9th              5  Married-spo
use-absent
7    52  Self-emp-not-inc  209642       HS-grad              9      Married-
civ-spouse
8    31           Private   45781       Masters             14           Nev
```

In [42]:
```
#2B
#finding columns that contain wrong data
categorical_columns = data.select_dtypes(include=['object', 'category']).columr
for column in categorical_columns:
    print(f"\nValue counts for categorical column {column}:")
    print(data[column].value_counts(dropna=False))
```

```
Value counts for categorical column workclass:
workclass
Private             33906
Self-emp-not-inc     3862
Local-gov            3136
State-gov            1981
?                    1836
Self-emp-inc         1695
Federal-gov          1432
NaN                   963
Without-pay            21
Never-worked           10
Name: count, dtype: int64

Value counts for categorical column education:
education
HS-grad         15784
Some-college    10878
```

##################################### WRITE YOUR REPORT IN THIS CELL (IF APPLICABLE)### 2.b. Using the select data types for categorical columns function which is data.select_dtypes(include=['object', 'category']).columns, the columns with wrong data were identified as Workclass, Occupation and Native-Country (25 WORDS)

##################################### WRITE YOUR REPORT IN THIS CELL (IF APPLICABLE)###

2.c. METHODS WE COULD USE TO FIND COLUMNS THAT CONTAINS WRONG DATA

1. Data Type Checks Ensuring that each column contains data of the correct type (e.g., numerical, string, datetime) is crucial. Sometimes, numeric values may be incorrectly entered or imported as strings, or vice versa, leading to incorrect data processing.
2. Null or Missing Values Analysis Identifying and analyzing null or missing values in each column can indicate issues with data collection or processing. While not all missing values represent incorrect data, their presence can impact data analysis and may require imputation or other handling strategies.
3. Use of Automated Data Cleaning Tools Several software tools and libraries (e.g., Pandas in Python, dplyr in R) offer functions and methods for data cleaning, including identifying outliers, handling missing values, and detecting inconsistent data types.
4. Data Visualization Visualizing data through histograms, box plots, scatter plots, and other graphical representations can quickly reveal outliers, data clusters, and trends that do not align with expectations. For instance, a box plot might show values that fall far outside the typical range for a column, indicating potential errors or outliers.

5. Descriptive Statistics Generating descriptive statistics (such as mean, median, mode, min, max, variance, standard deviation) for numerical columns can reveal outliers or values that don't make sense (e.g., negative ages, percentages over 100). For categorical data, examining the frequency of each category can help identify misspellings, inconsistent labeling, or unexpected categories. (238 WORDS)

TASK 2.2: Complete the following tasks (Completing the report cell is required):

a. Determine whether the columns which contain Null values are numerical, nominal, or ordinal variables and report it in the report cell (1%).

b. Delete all rows which contain Null values (1%). Then print the number of Null values of all columns using the isna().sum() method.

c. Import the dataset again. Find the mode of the categorical (i.e. nominal or ordinal) columns which contain Null values and use it to fill their Null values (2%)

In [43]:
```
############# WRITE YOUR CODE IN THIS CELL (IF APPLICABLE)####################
# Determining the columns by count with null values
null_count_per_column = data.isna().sum()

#printing outcome
print("Number of null values in each column:")
print(null_count_per_column)
```

```
Number of null values in each column:
age                 0
workclass         963
fnlwgt              0
education           0
education-num       0
marital-status      0
occupation        966
relationship        0
race                0
sex                 0
capital-gain        0
capital-loss        0
hours-per-week      0
native-country    274
income              0
dtype: int64
```

```
############# WRITE YOUR REPORT IN THIS CELL (IF
APPLICABLE)####################
2.2A
```

In the above adult dataset, the two columns that contain null values 'workclass' and 'native-country' are nominal variables columns because the variables they contain are not ranked or ordered. As the name implies, nominal variable refers to data in categories or labels that cannot be ordered or ranked. However, the third column occupation is ordinal variable because the data or variables are ordered, it has variables like Prof-specialty, Executive Managerial which are ranks.

In [44]:
```
############# WRITE YOUR CODE IN THIS CELL (IF APPLICABLE)###################

#2.2B:deleting all rows which contain Null values
data.dropna(inplace=True)

# Printing the number of null values in each column
null_count_per_column = data.isna().sum()
print("Number of null values in each column:")
print(null_count_per_column)
```

```
Number of null values in each column:
age               0
workclass         0
fnlwgt            0
education         0
education-num     0
marital-status    0
occupation        0
relationship      0
race              0
sex               0
capital-gain      0
capital-loss      0
hours-per-week    0
native-country    0
income            0
dtype: int64
```

In [45]:
```
#2.2C
#importing the dataset again
data = pd.read_csv('C:/Users/HP/Downloads/adult.csv')

#printing
data
```

Out[45]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black
...	...	...	...	...	...	...	...	...	...
48837	39	Private	215419	Bachelors	13	Divorced	Prof-specialty	Not-in-family	White
48838	64	NaN	321403	HS-grad	9	Widowed	NaN	Other-relative	Black
48839	38	Private	374983	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	White
48840	44	Private	83891	Bachelors	13	Divorced	Adm-clerical	Own-child	Asian-Pac-Islander
48841	35	Self-emp-inc	182148	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White

48842 rows × 15 columns

In [46]:
```
#finding the mode of 'workclass' column which contains null values
data['workclass'].mode()[0]
mode_workclass = data['workclass'].mode()[0]

print ("The mode of 'workclass' categorical variable column is")
print (mode_workclass)

#finding the mode of 'occupation' column which contains null values
data['occupation'].mode()[0]
mode_occupation = data['occupation'].mode()[0]

print ("The mode of 'occupation' categorical variable column is")
print (mode_occupation)

#finding the mode of 'native-country' column which contains null values
data['native-country'].mode()[0]
mode_native_country = data['native-country'].mode()[0]

print ("The mode of 'native-country' categorical variable column is")
print (mode_native_country)
```

```
The mode of 'workclass' categorical variable column is
Private
The mode of 'occupation' categorical variable column is
Prof-specialty
The mode of 'native-country' categorical variable column is
United-States
```

In [47]:
```
#Printing the number of null values in each column before filling the nulls wit
null_count_per_column = data.isna().sum()
print("Number of null values in each column:")
print(null_count_per_column)
```

```
Number of null values in each column:
age                 0
workclass         963
fnlwgt              0
education           0
education-num       0
marital-status      0
occupation        966
relationship        0
race                0
sex                 0
capital-gain        0
capital-loss        0
hours-per-week      0
native-country    274
income              0
dtype: int64
```

In [48]:
```
#filling null values in 'workclass' column with the mode
data['workclass'].fillna(mode_workclass, inplace=True)

#filling null values in 'occupation' column with the mode
data['occupation'].fillna(mode_occupation, inplace=True)

#filling null values in 'native-country' column with the mode
data['native-country'].fillna(mode_native_country, inplace=True)

#Printing the number of null values in each column after filling the nulls with
null_count_per_column = data.isna().sum()
print("Number of null values in each column:")
print(null_count_per_column)

#confirming further that the null values have been filled with the column mode
data
```

```
Number of null values in each column:
age               0
workclass         0
fnlwgt            0
education         0
education-num     0
marital-status    0
occupation        0
relationship      0
race              0
sex               0
capital-gain      0
capital-loss      0
hours-per-week    0
native-country    0
income            0
dtype: int64
```

Out[48]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black
...	...	...	...	...	...	...	...	...	...
48837	39	Private	215419	Bachelors	13	Divorced	Prof-specialty	Not-in-family	White
48838	64	Private	321403	HS-grad	9	Widowed	Prof-specialty	Other-relative	Black
48839	38	Private	374983	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	White
48840	44	Private	83891	Bachelors	13	Divorced	Adm-clerical	Own-child	Asian-Pac-Islander
48841	35	Self-emp-inc	182148	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White

48842 rows × 15 columns

In [49]:
```
#printing the first 100 rows
print(data.head(100))
```

```
    age         workclass  fnlwgt     education  education-num         marita
l-status  \
0    39         State-gov   77516     Bachelors             13          Never
-married
1    50  Self-emp-not-inc   83311     Bachelors             13     Married-ci
v-spouse
2    38           Private  215646       HS-grad              9
Divorced
3    53           Private  234721          11th              7     Married-ci
v-spouse
4    28           Private  338409     Bachelors             13     Married-ci
v-spouse
5    37           Private  284582       Masters             14     Married-ci
v-spouse
6    49           Private  160187           9th              5  Married-spous
e-absent
7    52  Self-emp-not-inc  209642       HS-grad              9     Married-ci
v-spouse
8    31           Private   45781       Masters             14          Never
-married
9    42           Private  159449     Bachelors             13     Married-ci
v-spouse
10   37           Private  280464  Some-college             10     Married-ci
v-spouse
11   30         State-gov  141297     Bachelors             13     Married-ci
v-spouse
12   23           Private  122272     Bachelors             13          Never
-married
13   32           Private  205019    Assoc-acdm             12          Never
-married
14   40           Private  121772     Assoc-voc             11     Married-ci
v-spouse
15   34           Private  245487       7th-8th              4     Married-ci
v-spouse
16   25  Self-emp-not-inc  176756       HS-grad              9          Never
-married
17   32           Private  186824       HS-grad              9          Never
-married
18   38           Private   28887          11th              7     Married-ci
v-spouse
19   43  Self-emp-not-inc  292175       Masters             14
Divorced
20   40           Private  193524     Doctorate             16     Married-ci
v-spouse
21   54           Private  302146       HS-grad              9              S
eparated
22   35       Federal-gov   76845           9th              5     Married-ci
v-spouse
23   43           Private  117037          11th              7     Married-ci
v-spouse
24   59           Private  109015       HS-grad              9
Divorced
25   56         Local-gov  216851     Bachelors             13     Married-ci
v-spouse
26   19           Private  168294       HS-grad              9          Never
-married
27   54                 ?  180211  Some-college             10     Married-ci
```

```
v-spouse
28   39            Private  367260        HS-grad              9              
Divorced
29   49            Private  193366        HS-grad              9    Married-ci
v-spouse
30   23          Local-gov  190709     Assoc-acdm             12         Never
-married
31   20            Private  266015   Some-college             10         Never
-married
32   45            Private  386940      Bachelors             13              
Divorced
33   30        Federal-gov   59951   Some-college             10    Married-ci
v-spouse
34   22          State-gov  311512   Some-college             10    Married-ci
v-spouse
35   48            Private  242406           11th              7         Never
-married
36   21            Private  197200   Some-college             10         Never
-married
37   19            Private  544091        HS-grad              9     Married-A
F-spouse
38   31            Private   84154   Some-college             10    Married-ci
v-spouse
39   48   Self-emp-not-inc  265477     Assoc-acdm             12    Married-ci
v-spouse
40   31            Private  507875            9th              5    Married-ci
v-spouse
41   53   Self-emp-not-inc   88506      Bachelors             13    Married-ci
v-spouse
42   24            Private  172987      Bachelors             13    Married-ci
v-spouse
43   49            Private   94638        HS-grad              9             S
eparated
44   25            Private  289980        HS-grad              9         Never
-married
45   57        Federal-gov  337895      Bachelors             13    Married-ci
v-spouse
46   53            Private  144361        HS-grad              9    Married-ci
v-spouse
47   44            Private  128354        Masters             14              
Divorced
48   41          State-gov  101603      Assoc-voc             11    Married-ci
v-spouse
49   29            Private  271466      Assoc-voc             11         Never
-married
50   25            Private   32275   Some-college             10    Married-ci
v-spouse
51   18            Private  226956        HS-grad              9         Never
-married
52   47            Private   51835    Prof-school             15    Married-ci
v-spouse
53   50        Federal-gov  251585      Bachelors             13              
Divorced
54   47       Self-emp-inc  109832        HS-grad              9              
Divorced
55   43            Private  237993   Some-college             10    Married-ci
v-spouse
```

```
56   46           Private  216666        5th-6th              3     Married-ci
v-spouse
57   35           Private   56352      Assoc-voc             11     Married-ci
v-spouse
58   41           Private  147372        HS-grad              9     Married-ci
v-spouse
59   30           Private  188146        HS-grad              9     Married-ci
v-spouse
60   30           Private   59496      Bachelors             13     Married-ci
v-spouse
61   32                 ?  293936        7th-8th              4  Married-spous
e-absent
62   48           Private  149640        HS-grad              9     Married-ci
v-spouse
63   42           Private  116632      Doctorate             16     Married-ci
v-spouse
64   29           Private  105598   Some-college             10
Divorced
65   36           Private  155537        HS-grad              9     Married-ci
v-spouse
66   28           Private  183175   Some-college             10
Divorced
67   53           Private  169846        HS-grad              9     Married-ci
v-spouse
68   49      Self-emp-inc  191681   Some-college             10     Married-ci
v-spouse
69   25                 ?  200681   Some-college             10          Never
-married
70   19           Private  101509   Some-college             10          Never
-married
71   31           Private  309974      Bachelors             13              S
eparated
72   29  Self-emp-not-inc  162298      Bachelors             13     Married-ci
v-spouse
73   23           Private  211678   Some-college             10          Never
-married
74   79           Private  124744   Some-college             10     Married-ci
v-spouse
75   27           Private  213921        HS-grad              9          Never
-married
76   40           Private   32214     Assoc-acdm             12     Married-ci
v-spouse
77   67                 ?  212759           10th              6     Married-ci
v-spouse
78   18           Private  309634           11th              7          Never
-married
79   31         Local-gov  125927        7th-8th              4     Married-ci
v-spouse
80   18           Private  446839        HS-grad              9          Never
-married
81   52           Private  276515      Bachelors             13     Married-ci
v-spouse
82   46           Private   51618        HS-grad              9     Married-ci
v-spouse
83   59           Private  159937        HS-grad              9     Married-ci
v-spouse
84   44           Private  343591        HS-grad              9
```

```
Divorced
85   53           Private  346253       HS-grad             9
Divorced
86   49         Local-gov  268234       HS-grad             9     Married-ci
v-spouse
87   33           Private  202051       Masters            14     Married-ci
v-spouse
88   30           Private   54334           9th             5          Never
-married
89   43       Federal-gov  410867     Doctorate            16          Never
-married
90   57           Private  249977     Assoc-voc            11     Married-ci
v-spouse
91   37           Private  286730  Some-college            10
Divorced
92   28           Private  212563  Some-college            10
Divorced
93   30           Private  117747       HS-grad             9     Married-ci
v-spouse
94   34         Local-gov  226296     Bachelors            13     Married-ci
v-spouse
95   29         Local-gov  115585  Some-college            10          Never
-married
96   48  Self-emp-not-inc  191277     Doctorate            16     Married-ci
v-spouse
97   37           Private  202683  Some-college            10     Married-ci
v-spouse
98   48           Private  171095    Assoc-acdm            12
Divorced
99   32       Federal-gov  249409       HS-grad             9          Never
-married

           occupation    relationship                race     sex  capital-ga
in  capital-loss  \
0        Adm-clerical   Not-in-family               White    Male          21
74            0
1     Exec-managerial         Husband               White    Male
0            0
2   Handlers-cleaners   Not-in-family               White    Male
0            0
3   Handlers-cleaners         Husband               Black    Male
0            0
4      Prof-specialty            Wife               Black  Female
0            0
5     Exec-managerial            Wife               White  Female
0            0
6       Other-service   Not-in-family               Black  Female
0            0
7     Exec-managerial         Husband               White    Male
0            0
8      Prof-specialty   Not-in-family               White  Female         140
84            0
9     Exec-managerial         Husband               White    Male          51
78            0
10    Exec-managerial         Husband               Black    Male
0            0
11     Prof-specialty         Husband  Asian-Pac-Islander    Male
```

```
0            0
12       Adm-clerical       Own-child               White  Female
0            0
13              Sales   Not-in-family               Black    Male
0            0
14       Craft-repair         Husband  Asian-Pac-Islander    Male
0            0
15   Transport-moving         Husband  Amer-Indian-Eskimo    Male
0            0
16    Farming-fishing       Own-child               White    Male
0            0
17  Machine-op-inspct       Unmarried               White    Male
0            0
18              Sales         Husband               White    Male
0            0
19    Exec-managerial       Unmarried               White  Female
0            0
20     Prof-specialty         Husband               White    Male
0            0
21      Other-service       Unmarried               Black  Female
0            0
22    Farming-fishing         Husband               Black    Male
0            0
23   Transport-moving         Husband               White    Male
0         2042
24       Tech-support       Unmarried               White  Female
0            0
25       Tech-support         Husband               White    Male
0            0
26       Craft-repair       Own-child               White    Male
0            0
27                  ?         Husband  Asian-Pac-Islander    Male
0            0
28    Exec-managerial   Not-in-family               White    Male
0            0
29       Craft-repair         Husband               White    Male
0            0
30    Protective-serv   Not-in-family               White    Male
0            0
31              Sales       Own-child               Black    Male
0            0
32    Exec-managerial       Own-child               White    Male
0         1408
33       Adm-clerical       Own-child               White    Male
0            0
34      Other-service         Husband               Black    Male
0            0
35  Machine-op-inspct       Unmarried               White    Male
0            0
36  Machine-op-inspct       Own-child               White    Male
0            0
37       Adm-clerical            Wife               White  Female
0            0
38              Sales         Husband               White    Male
0            0
39     Prof-specialty         Husband               White    Male
0            0
```

```
40  Machine-op-inspct         Husband              White    Male
0            0
41     Prof-specialty         Husband              White    Male
0            0
42       Tech-support         Husband              White    Male
0            0
43       Adm-clerical       Unmarried              White  Female
0            0
44  Handlers-cleaners   Not-in-family              White    Male
0            0
45     Prof-specialty         Husband              Black    Male
0            0
46  Machine-op-inspct         Husband              White    Male
0            0
47    Exec-managerial       Unmarried              White  Female
0            0
48       Craft-repair         Husband              White    Male
0            0
49     Prof-specialty   Not-in-family              White    Male
0            0
50    Exec-managerial            Wife              Other  Female
0            0
51      Other-service       Own-child              White  Female
0            0
52     Prof-specialty            Wife              White  Female
0         1902
53    Exec-managerial   Not-in-family              White    Male
0            0
54    Exec-managerial   Not-in-family              White    Male
0            0
55       Tech-support         Husband              White    Male
0            0
56  Machine-op-inspct         Husband              White    Male
0            0
57      Other-service         Husband              White    Male
0            0
58       Adm-clerical         Husband              White    Male
0            0
59  Machine-op-inspct         Husband              White    Male             50
13            0
60              Sales         Husband              White    Male             24
07            0
61                  ?   Not-in-family              White    Male
0            0
62    Transport-moving         Husband              White    Male
0            0
63     Prof-specialty         Husband              White    Male
0            0
64       Tech-support   Not-in-family              White    Male
0            0
65       Craft-repair         Husband              White    Male
0            0
66       Adm-clerical   Not-in-family              White  Female
0            0
67       Adm-clerical            Wife              White  Female
0            0
68    Exec-managerial         Husband              White    Male
```

```
0            0  
69                   ?       Own-child                White    Male  
0            0  
70     Prof-specialty       Own-child                White    Male  
0            0  
71               Sales       Own-child                Black  Female  
0            0  
72               Sales         Husband                White    Male  
0            0  
73  Machine-op-inspct   Not-in-family                White    Male  
0            0  
74     Prof-specialty  Other-relative                White    Male  
0            0  
75      Other-service       Own-child                White    Male  
0            0  
76       Adm-clerical         Husband                White    Male  
0            0  
77                   ?         Husband                White    Male  
0            0  
78      Other-service       Own-child                White  Female  
0            0  
79    Farming-fishing         Husband                White    Male  
0            0  
80               Sales   Not-in-family                White    Male  
0            0  
81      Other-service         Husband                White    Male  
0            0  
82      Other-service            Wife                White  Female  
0            0  
83               Sales         Husband                White    Male  
0            0  
84       Craft-repair   Not-in-family                White  Female         143
44            0  
85               Sales       Own-child                White  Female  
0            0  
86    Protective-serv         Husband                White    Male  
0            0  
87     Prof-specialty         Husband                White    Male  
0            0  
88               Sales   Not-in-family                White    Male  
0            0  
89     Prof-specialty   Not-in-family                White  Female  
0            0  
90     Prof-specialty         Husband                White    Male  
0            0  
91       Craft-repair       Unmarried                White  Female  
0            0  
92  Machine-op-inspct       Unmarried                Black  Female  
0            0  
93               Sales            Wife  Asian-Pac-Islander  Female  
0         1573  
94    Protective-serv         Husband                White    Male  
0            0  
95  Handlers-cleaners   Not-in-family                White    Male  
0            0  
96     Prof-specialty         Husband                White    Male  
0         1902  
```

```
97              Sales         Husband              White    Male
0           0
98    Exec-managerial       Unmarried              White  Female
0           0
99      Other-service       Own-child              Black    Male
0           0

    hours-per-week native-country income
0               40  United-States  <=50K
1               13  United-States  <=50K
2               40  United-States  <=50K
3               40  United-States  <=50K
4               40           Cuba  <=50K
5               40  United-States  <=50K
6               16        Jamaica  <=50K
7               45  United-States   >50K
8               50  United-States   >50K
9               40  United-States   >50K
10              80  United-States   >50K
11              40          India   >50K
12              30  United-States  <=50K
13              50  United-States  <=50K
14              40              ?   >50K
15              45         Mexico  <=50K
16              35  United-States  <=50K
17              40  United-States  <=50K
18              50  United-States  <=50K
19              45  United-States   >50K
20              60  United-States   >50K
21              20  United-States  <=50K
22              40  United-States  <=50K
23              40  United-States  <=50K
24              40  United-States  <=50K
25              40  United-States   >50K
26              40  United-States  <=50K
27              60          South   >50K
28              80  United-States  <=50K
29              40  United-States  <=50K
30              52  United-States  <=50K
31              44  United-States  <=50K
32              40  United-States  <=50K
33              40  United-States  <=50K
34              15  United-States  <=50K
35              40    Puerto-Rico  <=50K
36              40  United-States  <=50K
37              25  United-States  <=50K
38              38              ?   >50K
39              40  United-States  <=50K
40              43  United-States  <=50K
41              40  United-States  <=50K
42              50  United-States  <=50K
43              40  United-States  <=50K
44              35  United-States  <=50K
45              40  United-States   >50K
46              38  United-States  <=50K
47              40  United-States  <=50K
48              40  United-States  <=50K
```

```
49                43  United-States  <=50K
50                40  United-States  <=50K
51                30              ?  <=50K
52                60       Honduras   >50K
53                55  United-States   >50K
54                60  United-States  <=50K
55                40  United-States   >50K
56                40         Mexico  <=50K
57                40    Puerto-Rico  <=50K
58                48  United-States  <=50K
59                40  United-States  <=50K
60                40  United-States  <=50K
61                40              ?  <=50K
62                40  United-States  <=50K
63                45  United-States   >50K
64                58  United-States  <=50K
65                40  United-States  <=50K
66                40  United-States  <=50K
67                40  United-States   >50K
68                50  United-States   >50K
69                40  United-States  <=50K
70                32  United-States  <=50K
71                40  United-States  <=50K
72                70  United-States   >50K
73                40  United-States  <=50K
74                20  United-States  <=50K
75                40         Mexico  <=50K
76                40  United-States  <=50K
77                 2  United-States  <=50K
78                22  United-States  <=50K
79                40  United-States  <=50K
80                30  United-States  <=50K
81                40           Cuba  <=50K
82                40  United-States  <=50K
83                48  United-States  <=50K
84                40  United-States   >50K
85                35  United-States  <=50K
86                40  United-States   >50K
87                50  United-States  <=50K
88                40  United-States  <=50K
89                50  United-States   >50K
90                40  United-States  <=50K
91                40  United-States  <=50K
92                25  United-States  <=50K
93                35              ?  <=50K
94                40  United-States   >50K
95                50  United-States  <=50K
96                60  United-States   >50K
97                48  United-States   >50K
98                40        England  <=50K
99                40  United-States  <=50K
```

TASK 2.3. Select one of the columns which contains wrong data. Write a code to fill the wrong cells with an appropriate value. You have the freedom to determine

what value you want to use to fill the wrong cells (2%).

In [50]:
```
############# WRITE YOUR CODE IN THIS CELL (IF APPLICABLE)###################

# Find the mode of 'occupation' column excluding '?' values
mode_occupation = data[data['occupation'] != '?']['occupation'].mode()[0]

# Replace '?' with the mode
data['occupation'] = data['occupation'].replace('?', mode_occupation)

# Confirm that wrong cells have been corrected
print("Number of '?' values in 'occupation' column after correction is", (data[

print (data['occupation'].head (100))
```

```
Number of '?' values in 'occupation' column after correction is 0
0           Adm-clerical
1         Exec-managerial
2       Handlers-cleaners
3       Handlers-cleaners
4          Prof-specialty
5         Exec-managerial
6           Other-service
7         Exec-managerial
8          Prof-specialty
9         Exec-managerial
10        Exec-managerial
11         Prof-specialty
12          Adm-clerical
13                 Sales
14          Craft-repair
15       Transport-moving
16        Farming-fishing
17      Machine-op-inspct
18                 Sales
19        Exec-managerial
20         Prof-specialty
21          Other-service
22        Farming-fishing
23       Transport-moving
24          Tech-support
25          Tech-support
26          Craft-repair
27         Prof-specialty
28        Exec-managerial
29          Craft-repair
30        Protective-serv
31                 Sales
32        Exec-managerial
33          Adm-clerical
34          Other-service
35      Machine-op-inspct
36      Machine-op-inspct
37          Adm-clerical
38                 Sales
39         Prof-specialty
40      Machine-op-inspct
41         Prof-specialty
42          Tech-support
43          Adm-clerical
44      Handlers-cleaners
45         Prof-specialty
46      Machine-op-inspct
47        Exec-managerial
48          Craft-repair
49         Prof-specialty
50        Exec-managerial
51          Other-service
52         Prof-specialty
53        Exec-managerial
54        Exec-managerial
55          Tech-support
```

```
56    Machine-op-inspct
57        Other-service
58         Adm-clerical
59    Machine-op-inspct
60                Sales
61       Prof-specialty
62     Transport-moving
63       Prof-specialty
64         Tech-support
65         Craft-repair
66         Adm-clerical
67         Adm-clerical
68      Exec-managerial
69       Prof-specialty
70       Prof-specialty
71                Sales
72                Sales
73    Machine-op-inspct
74       Prof-specialty
75        Other-service
76         Adm-clerical
77       Prof-specialty
78        Other-service
79      Farming-fishing
80                Sales
81        Other-service
82        Other-service
83                Sales
84         Craft-repair
85                Sales
86      Protective-serv
87       Prof-specialty
88                Sales
89       Prof-specialty
90       Prof-specialty
91         Craft-repair
92    Machine-op-inspct
93                Sales
94      Protective-serv
95    Handlers-cleaners
96       Prof-specialty
97                Sales
98      Exec-managerial
99        Other-service
Name: occupation, dtype: object
```

```
############# WRITE YOUR REPORT IN THIS CELL (IF
APPLICABLE)####################
```

TASK 2.4: Complete the following tasks:

a. Assume we want to predict the income column as the output. Use the correct method to encode this column (1%).

b. Select one ordinal column and encode it using the appropriate method (1%) .

c. Select one nominal column and encode it using the appropriate method (1%).

d. After encoding the nominal column in the previous step, find a method to append the encoded data to the dataset (2%).

e. Normalise the numerical columns of the dataset (1%)

In [51]:

```
############# WRITE YOUR CODE IN THIS CELL (IF APPLICABLE)###############
#2.4a
#importing the label encoder library for encoding
from sklearn preprocessing import LabelEncoder
le = preprocessing.LabelEncoder()

#encoding the income column and printing the outcome
data['income'] = le.fit_transform(data['income'])

#printing encoded outcome
print(data['income'])
```

```
0        0
1        0
2        0
3        0
4        0
        ..
48837    1
48838    1
48839    1
48840    1
48841    3
Name: income, Length: 48842, dtype: int32
```

In [52]:
```
#2.4b
#importing ordinal encoder
from sklearn.preprocessing import OrdinalEncoder

# define ordinal encoding
encoder = OrdinalEncoder()

#selecting ordinal column 'Education' and transforming data
data['education'] = encoder.fit_transform(data[['education']])

#printing outcome
print(data['education'])
```

```
0         9.0
1         9.0
2        11.0
3         1.0
4         9.0
         ...
48837     9.0
48838    11.0
48839     9.0
48840     9.0
48841     9.0
Name: education, Length: 48842, dtype: float64
```

In [53]:
```
#2.4c
#importing label encoder
from sklearn.preprocessing import LabelEncoder
le = preprocessing.LabelEncoder()

#selecting nominal column
nominal_column = 'race'


# encoding the race column
encoded_race = le.fit_transform(data[nominal_column])

#printing outcome
encoded_race
```

Out[53]:
```
array([4, 4, 4, ..., 4, 1, 4])
```

In [54]:
```
#2.4d
#convert array into a list
rX = encoded_race.tolist()

#adding a new column and appending the encoded values
data['encoded_race'] = rX

#printing
data
```

Out[54]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race
0	39	State-gov	77516	9.0	13	Never-married	Adm-clerical	Not-in-family	White
1	50	Self-emp-not-inc	83311	9.0	13	Married-civ-spouse	Exec-managerial	Husband	White
2	38	Private	215646	11.0	9	Divorced	Handlers-cleaners	Not-in-family	White
3	53	Private	234721	1.0	7	Married-civ-spouse	Handlers-cleaners	Husband	Black
4	28	Private	338409	9.0	13	Married-civ-spouse	Prof-specialty	Wife	Black
...	...	...	...	...	...	...	...	...	...
48837	39	Private	215419	9.0	13	Divorced	Prof-specialty	Not-in-family	White
48838	64	Private	321403	11.0	9	Widowed	Prof-specialty	Other-relative	Black
48839	38	Private	374983	9.0	13	Married-civ-spouse	Prof-specialty	Husband	White
48840	44	Private	83891	9.0	13	Divorced	Adm-clerical	Own-child	Asian-Pac-Islander
48841	35	Self-emp-inc	182148	9.0	13	Married-civ-spouse	Exec-managerial	Husband	White

48842 rows × 16 columns

In [55]:
```
#2.4e
#importing the MinMaxScaler library
from sklearn.preprocessing import MinMaxScaler

#scaling the numerical columns
scaler = MinMaxScaler()
df_normalised=scaler.fit_transform(data.iloc[:, [0,2,3,4,10,11,12,14, 15]])
df_normalised=pd.DataFrame(df_normalised,columns=['age','fnlwgt','education','e
print(df_normalised)
```

```
            age    fnlwgt  education  education-num  capital-gain  capital-lo
ss  hours-per-week  \
0      0.301370  0.044131   0.600000       0.800000      0.021740
0.0        0.397959
1      0.452055  0.048052   0.600000       0.800000      0.000000
0.0        0.122449
2      0.287671  0.137581   0.733333       0.533333      0.000000
0.0        0.397959
3      0.493151  0.150486   0.066667       0.400000      0.000000
0.0        0.397959
4      0.150685  0.220635   0.600000       0.800000      0.000000
0.0        0.397959
...         ...       ...        ...            ...           ...
...         ...
48837  0.301370  0.137428   0.600000       0.800000      0.000000
0.0        0.357143
48838  0.643836  0.209130   0.733333       0.533333      0.000000
0.0        0.397959
48839  0.287671  0.245379   0.600000       0.800000      0.000000
0.0        0.500000
48840  0.369863  0.048444   0.600000       0.800000      0.054551
0.0        0.397959
48841  0.246575  0.114919   0.600000       0.800000      0.000000
0.0        0.602041

         income
0      0.000000
1      0.000000
2      0.000000
3      0.000000
4      0.000000
...         ...
48837  0.333333
48838  0.333333
48839  0.333333
48840  0.333333
48841  1.000000

[48842 rows x 8 columns]
```

us############ WRITE YOUR REPORT IN THIS CELL (IF APPLICABLE)###################

2.4a The code in 2.4a above performs an encoding of the income column of the task_dataset. Here the income column is a nominal column, and for nominal data, label encoding is the most suitable for encoding. So i imported the label Encoder from sklearn preprocessing, then encoded and tranformed the income column with le.fit_transform(data['income']) and assigned it into data['income'] as an input.

2.4b. The code in 2.4b performs an ordinal encoding of the education column. i first imported OrdinalEncoder from sklearn.preprocessing, selected education as the column for ordinal encoding being that the Education data is ordered. encoding and transforming the education data was then performed using encoder.fit_transform(data[['education']]) and assigned into data['education'] as an input

2.4c The code in 2.4c performs a label encoding of the race column which was selected and assigned into nominal_column. i first imported LabelEncoder from sklearn.preprocessing, encoded and transformed the race data using le.fit_transform(data[nominal_column]) and then assigned it into encoded_race

2.4d The code here rX = encoded_race.tolist() converts the encoded race arrays and into a list stored in rX and then adds it back to the dataset and appends it using data['encoded_race'] = rX

2.4e. The code here performs the task_dataset DataFrame normalisation using the MinMaxScaler which was imported from sklearn.preprocessing. Data.iloc, scaler.fit_transform(data.iloc[:, [0,2,3,4,10,11,12,14, 15]]) was used to select all numerical columns for the normalisation.