Batch: D2	**Roll No.:** 16010122323
Experiment 01	
Grade: AA / AB / BB / BC / CC / CD /DD	

Title: Dataset preparing/ pre-processing

Objective:

1. To learn how to prepare the dataset

2. To learn various steps in Data -Preprocessing

Course Outcome:

CO1: Learn how to locate and download datasets, extract insights from that data and present their findings in a variety of different formats.

Books/ Journals/ Websites referred:

1. Data Visualization made simple New York: Routledge - Kristen Sosulski, First edition, 2019
2. Sosulski, K. Data Visualization Made Simple: Insights into Becoming Visual, First edition, 2018
3. https://www.kaggle.com/uciml/adult-census-income
4. https://archive.ics.uci.edu/ml/datasets/adult
5. https://ori.hhs.gov/education/products/n_illinois_u/datamanagement/dctopic.html
6. A review of research process, data collection and analysis - Surya Raj Niraula
7. https://www.jigsawacademy.com/blogs/data-science/what-is-data-processing/

Resources used:

1. https://www.kaggle.com/uciml/adult-census-income
2. https://archive.ics.uci.edu/ml/datasets/adult

Theory (About Data Preprocessing):

Concept of Data processing is collecting and manipulating data into a usable and appropriate form. The automatic processing of data in a predetermined sequence of operations is the manipulation of data.

Data Preprocessing is a Data Mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends and is likely to contain many errors. Data Preprocessing is a proven method of resolving such issues. we need to transform or organize it to make it into a proper format by using Data Preprocessing.

Preprocessing of data is mainly to check the data quality. The quality can be checked by the following:

• Accuracy: To check whether the data entered is correct or not.
• Completeness: To check whether the data is available or not recorded.
• Consistency: To check whether the same data is kept in all the places that do or do not match.
• Timeliness: The data should be updated correctly.
• Believability: The data should be trustable.
• Interpretability: The understandability of the data.

Major Tasks in Data Preprocessing:

1. Data cleaning: process to remove incorrect data, incomplete data and inaccurate data from the datasets, including removing missing values.
2. Data integration: process of combining multiple sources into a single dataset.
3. Data reduction: process helps in the reduction of the volume of the data which makes the analysis easier.
4. Data transformation: process in which change is made in the format or the structure of the data.

WHY WE SHOULD USE DATA PROCESSING

In the modern era, most of the work relies on data, therefore collection of large amounts of data for different purposes. The processing of this data collected is essential so that the data goes through all the above-stated steps and gets sorted, stored, filtered, presented in the required format and analyzed.

IMPLEMENTATION:
Working (Put the code and Output for each Data Preprocessing task):

Different steps in Data Preprocessing:

- **Finding missing, null values etc.**
- **Replacing missing, null values with statistical parameters.**
- **Encoding categorical data**
- **Normalization**

1. PYTHON

DATA PREPROCESSING: In this experiment, we clean the data set according to our needs.

TASK 1: importing the necessary libraries.

In [8]:
```
import pandas as pd
import numpy as np
```

Task 2: Reading the data set and displaying the information about the same.

In [9]:
```
income_dataset = pd.read_csv('adult.csv')
income_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   age             32561 non-null  int64 
 1   workclass       32561 non-null  object
 2   fnlwgt          32561 non-null  int64 
 3   education       32561 non-null  object
 4   education.num   32561 non-null  int64 
 5   marital.status  32561 non-null  object
 6   occupation      32561 non-null  object
 7   relationship    32561 non-null  object
 8   race            32561 non-null  object
 9   sex             32561 non-null  object
 10  capital.gain    32561 non-null  int64 
 11  capital.loss    32561 non-null  int64 
 12  hours.per.week  32561 non-null  int64 
 13  native.country  32561 non-null  object
 14  income          32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

TASK 3: Removing the redundant columns not required for analysis with respect to the problem set.

In [10]:
```
redundant_columns = ['fnlwgt', 'education.num', 'capital.gain', 'capital.loss', 'hours.per.week']
final_income_dataset = income_dataset.drop(redundant_columns, axis = 1)
final_income_dataset
```

Out[10]:

	age	workclass	education	marital.status	occupation	relationship	race	sex	native.country	income
0	90	?	HS-grad	Widowed	?	Not-in-family	White	Female	United-States	<=50K
1	82	Private	HS-grad	Widowed	Exec-managerial	Not-in-family	White	Female	United-States	<=50K
2	66	?	Some-college	Widowed	?	Unmarried	Black	Female	United-States	<=50K
3	54	Private	7th-8th	Divorced	Machine-op-inspct	Unmarried	White	Female	United-States	<=50K
4	41	Private	Some-college	Separated	Prof-specialty	Own-child	White	Female	United-States	<=50K
...	...	...	...	...	...	...	...	...	...	...
32556	22	Private	Some-college	Never-married	Protective-serv	Not-in-family	White	Male	United-States	<=50K
32557	27	Private	Assoc-acdm	Married-civ-spouse	Tech-support	Wife	White	Female	United-States	<=50K
32558	40	Private	HS-grad	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	United-States	>50K
32559	58	Private	HS-grad	Widowed	Adm-clerical	Unmarried	White	Female	United-States	<=50K
32560	22	Private	HS-grad	Never-married	Adm-clerical	Own-child	White	Male	United-States	<=50K

32561 rows × 10 columns

TASK 4: Extracting the independent variable.

```
In [11]: independent_variable = final_income_dataset.iloc[:, :-1].values
         independent_variable
```

```
Out[11]: array([[90, '?', 'HS-grad', ..., 'White', 'Female', 'United-States'],
                [82, 'Private', 'HS-grad', ..., 'White', 'Female',
                 'United-States'],
                [66, '?', 'Some-college', ..., 'Black', 'Female', 'United-States'],
                ...,
                [40, 'Private', 'HS-grad', ..., 'White', 'Male', 'United-States'],
                [58, 'Private', 'HS-grad', ..., 'White', 'Female',
                 'United-States'],
                [22, 'Private', 'HS-grad', ..., 'White', 'Male', 'United-States']],
               dtype=object)
```

TASK 5: Exctracting dependent variable.

```
In [12]: dependent_variable = final_income_dataset.iloc[:, 9].values
         dependent_variable
```

```
Out[12]: array(['<=50K', '<=50K', '<=50K', ..., '>50K', '<=50K', '<=50K'],
               dtype=object)
```


TASK 5: Exctracting dependent variable.

```
In [12]: dependent_variable = final_income_dataset.iloc[:, 9].values
         dependent_variable
```

```
Out[12]: array(['<=50K', '<=50K', '<=50K', ..., '>50K', '<=50K', '<=50K'],
               dtype=object)
```


TASK 6: Taking care of missing data and replacing it with 'NA'.

```
In [13]: from sklearn.impute import SimpleImputer
         imputer = SimpleImputer(missing_values = '?', strategy = 'constant', fill_value='NA')
         transformed_values = imputer.fit_transform(independent_variable)
         transformed_values
```

```
Out[13]: array([[90, 'NA', 'HS-grad', ..., 'White', 'Female', 'United-States'],
                [82, 'Private', 'HS-grad', ..., 'White', 'Female',
                 'United-States'],
                [66, 'NA', 'Some-college', ..., 'Black', 'Female',
                 'United-States'],
                ...,
                [40, 'Private', 'HS-grad', ..., 'White', 'Male', 'United-States'],
                [58, 'Private', 'HS-grad', ..., 'White', 'Female',
                 'United-States'],
                [22, 'Private', 'HS-grad', ..., 'White', 'Male', 'United-States']],
               dtype=object)
```

TASK 7: Encoding the Categorical data.

```
In [14]: from sklearn.preprocessing import LabelEncoder
         for i in range(1, 9):
             transformed_values[:, i] = LabelEncoder().fit_transform(transformed_values[:, i])
         transformed_values
```

```
Out[14]: array([[90, 2, 11, ..., 4, 0, 39],
                [82, 4, 11, ..., 4, 0, 39],
                [66, 2, 15, ..., 2, 0, 39],
                ...,
                [40, 4, 11, ..., 4, 1, 39],
                [58, 4, 11, ..., 4, 0, 39],
                [22, 4, 11, ..., 4, 1, 39]], dtype=object)
```

2. RStudio

Task 1: Reading the dataset and displaying it.

```
Console  Terminal ×  Jobs ×
R 4.1.1 · ~/
es" ...
 $ income        : chr  "<=50K" "<=50K" "<=50K" "<=50K" ...
> dataset=read.csv("C:/Users/Tanvi/Desktop/adultFinal.csv")
> str(dataset)
'data.frame':   100 obs. of  15 variables:
 $ age           : int  90 82 66 54 41 34 38 74 68 41 ...
 $ workclass     : chr  "?" "Private" "?" "Private" ...
 $ education     : chr  "HS-grad" "HS-grad" "Some-college" "7th-8th" ...
 $ marital.status: chr  "Widowed" "Widowed" "Widowed" "Divorced" ...
 $ occupation    : chr  "?" "Exec-managerial" "?" "Machine-op-inspct" ...
 $ relationship  : chr  "Not-in-family" "Not-in-family" "Unmarried" "Unmarried" ...
 $ race          : chr  "White" "White" "Black" "White" ...
 $ sex           : chr  "Female" "Female" "Female" "Female" ...
 $ capital.gain  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ capital.loss  : int  4356 4356 4356 3900 3900 3770 3770 3683 3683 3004 ...
 $ hours.per.week: int  40 18 40 40 40 45 40 20 40 60 ...
 $ native.country: chr  "United-States" "United-States" "United-States" "United-Stat
es" ...
 $ income        : chr  "<=50K" "<=50K" "<=50K" "<=50K" ...
 $ education.num : int  9 9 10 4 10 9 6 16 9 10 ...
 $ fnlwgt        : int  77053 132870 186061 140359 264663 216864 150601 88638 422013
 70037 ...
```

Task 2: Removing the columns that are not required
(Creating a new dataset and storing the required columns)

```
> dataset.new<-dataset[c(1:8,12:13)]
> names(dataset.new)
 [1] "age"            "workclass"      "education"      "marital.status" "occupatio
n"
 [6] "relationship"   "race"           "sex"            "native.country" "income"

> nrows(dataset.new)
Error in nrows(dataset.new) : could not find function "nrows"
> nrow(dataset.new)
[1] 100
>
```

Task 3: Taking care of missing data be replacing “?” with NA

```
Console  Terminal ×  Jobs ×
R 4.1.1 · ~/
Error in nrows(dataset.new) : could not find function "nrows"
> nrow(dataset.new)
[1] 100
> dataset.new$workclass <- as.character(dataset.new$workclass)
> dataset.new$education <- as.character(dataset.new$education)
> dataset.new$marital.status <- as.character(dataset.new$marital.status)
> dataset.new$occupation <- as.character(dataset.new$occupation)
> dataset.new$relationship <- as.character(dataset.new$relationship)
> dataset.new$race <- as.character(dataset.new$race)
> dataset.new$sex <- as.character(dataset.new$sex)
> dataset.new$native.country <- as.character(dataset.new$native.country)
> library("dplyr")
```

```
Console  Terminal ×  Jobs ×
R 4.1.1 · ~/
The downloaded binary packages are in
        C:\Users\Tanvi\AppData\Local\Temp\RtmpOaOSGT\downloaded_packages
> library("dplyr")
> library(stringr)
> dataset.new %>%
+ mutate_if(is.character, str_trim) -> dataset.new
> dataset.new$workclass <- as.factor(dataset.new$workclass)
> dataset.new$education <- as.factor(dataset.new$education)
> dataset.new$marital.status <- as.factor(dataset.new$marital.status)
> dataset.new$occupation <- as.factor(dataset.new$occupation)
> dataset.new$relationship <- as.factor(dataset.new$relationship)
> dataset.new$race <- as.factor(dataset.new$race)
> dataset.new$sex <- as.factor(dataset.new$sex)
> dataset.new$native.country <- as.factor(dataset.new$native.country)
> dataset.new[dataset.new == "?"] <- NA
> sum(is.na(dataset.new))
[1] 21
```

Task 4: Encoding the Categorical Data

```
> dataset.new$workclass=factor(dataset.new$workclass,levels=c('Private','State-gov','Federal-gov','Self-emp-not-inc','Self-emp-inc','Local-gov','NA'),labels=c(1,2,3,4,5,6,7))
> dataset.new$education=factor(dataset.new$education,levels=c('1st-4th','5th-6th','7th-8th','10th','11th','Some-college','Bachelors','Masters','Doctorate','HS-grad','Doctorate','Prof-school','Assoc-acdm','Assoc-voc','NA'),labels=c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15))
> dataset.new$marital.status=factor(dataset.new$marital.status,levels=c('Widowed','Divorced','Seperated','Never-married','Married-civ-spouse','Married-spouse-absent','NA'),labels=c(1,2,3,4,5,6,7))
> dataset.new$occupation=factor(dataset.new$occupation,levels=c('Exec-managerial','Machine-op-inspct','Prof-specialty','Other-service','Adm-clerical','Craft-repair','Transport-moving','Handlers-cleaners','Sales','Farming-fishing','Tech-support','Protective-serv','NA'),labels=c(1,2,3,4,5,6,7,8,9,10,11,12,13))
> dataset.new$relationship=factor(dataset.new$relationship,levels=c('Not-in-family','Unmarried','Own-child','Other-relative','Husband','Wife','NA'),labels=c(1,2,3,4,5,6,7))
> dataset.new$race=factor(dataset.new$race,levels=c('White','Black','Asian-Pac-Islander','NA'),labels=c(1,2,3,4))
> dataset.new$race=factor(dataset.new$race,levels=c('White','Black','Asian-Pac-Islander','NA'),labels=c(1,2,3,4))
> dataset.new$race=factor(dataset.new$race,levels=c('White','Black','Asian-Pac-Islander','NA'),labels=c(1,2,3,4))
> dataset.new$sex=factor(dataset.new$sex,levels=c('Female','Male','NA'),labels=c(1,2,3))
> dataset.new$native.country=factor(dataset.new$native.country,levels=c('United-States','Mexic
```

```
> dataset.new$native.country=factor(dataset.new$native.country,levels=c('United-States','Mexico','Greece','Vietnam','China','Taiwan','India','Philippines','NA'),labels=c(1,2,3,4,5,6,7,8,9))
>
```

Task 5: NA to 0

```
> dataset[which(is.na(dataset))]<-0
```

Final: (Result from 1 to 20 observations)

	i..age	workclass	education	marital.status	occupation	relationship	race	sex	native.country	income
1	90	0	10	1	0	1	1	1	1	1
2	82	1	10	1	0	1	1	1	1	1
3	66	0	6	1	0	2	2	1	1	1
4	54	1	3	2	2	2	1	1	1	1
5	41	1	6	0	3	3	1	1	1	1
6	34	1	10	2	4	2	1	1	1	1
7	38	1	4	0	5	2	1	2	1	1
8	74	2	9	4	3	4	1	1	1	2
9	68	3	10	2	3	1	1	1	1	1
10	41	1	6	4	6	2	1	2	0	2
11	45	1	9	2	3	2	2	1	1	2
12	38	4	11	4	3	1	1	2	1	2
13	52	1	7	1	4	1	1	1	1	2
14	32	1	8	0	0	1	1	2	1	2
15	51	0	9	4	0	1	1	2	1	2
16	46	1	11	2	3	1	1	2	1	2
17	45	1	5	2	7	1	1	2	1	2
18	57	1	8	2	0	1	1	2	1	2
19	22	1	12	4	8	1	2	2	0	2
20	34	1	7	0	9	1	1	2	1	2
21	37	1	7	4	0	1	1	2	1	2
22	29	1	5	0	9	1	1	1	1	1

Conclusion (Students should write in their own words):

Through this experiment, we were working on a chosen dataset. We learnt about data processing methods i.e., cleaning, integrations, reduction, transformations; and steps and implemented the same on our “Adult Income” dataset.

Post Lab Question:

1. **Write the importance of Data Preprocessing in Software System Designing**

When exploring this wealth of information data pre-processing cleans and prepares the data before predictive models are developed. Predictions from incorrect data can be difficult to debug, or worse, can lead to inaccurate or misleading results that impact system performance and reliability. The goal here is to find the most predictive features

of the data and filter it so it will enhance the predictive power of the analytics model. Some common techniques include feature selection to reduce high-dimension data, feature extraction and transformation for dimensionality reduction, and domain analysis such as signal, image, and video processing.
The information gathered from data pre-processing is then taken and implemented across a number of analytics-driven embedded systems. An example of this is the innovation in using Big Data and analytics to make cars smarter. Automotive OEMs are collecting enormous amounts of data from real-world driving situations (think millions of miles of driving), recording data such as engine performance, video, radar, and other signals. This data is used to generate important metrics such as fuel economy and performance at the fleet level. Engineering teams are also using this real-world data to design, develop, and test new types of automotive systems, such as advanced driver assistance systems (ADAS).

.