# Data Mining - Lab - 2

# Numpy & Perform Data Exploration with Pandas

# 23010101407

---

## Numpy

1. NumPy (Numerical Python) is a powerful open-source library in Python used for numerical and scientific computing.
2. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on them efficiently.
3. NumPy is highly optimized and written in C, making it much faster than using regular Python lists for numerical operations.
4. It serves as the foundation for many other Python libraries in data science and machine learning, like pandas, TensorFlow, and scikit-learn.
5. With features like broadcasting, vectorization, and integration with C/C++ code, NumPy allows for cleaner and faster code in numerical computations.

### Step 1. Import the Numpy library

In [3]:
```
import numpy as np
```

### Step 2. Create a 1D array of numbers

In [5]:
```
# array ma badha element same data type na leva na
arr = np.array([1,4,2,5,6,3])
arr
```

Out[5]:
```
array([1, 4, 2, 5, 6, 3])
```

In [21]:
```
# jetli range hoy enathi ek ocho ave
# jyathi start krvu hoy te coma seprated dy devanu (2,10) to otp 2 to 9 no array
arr = np.arange(10)
type(arr) #type of arr
arr.dtype # datatype ape int, float etc...
arr.size # size ape
arr.ndim # ---> no. of dimantion
```

Out[21]:
```
1
```

## Step 3. Reshape 1D to 2D Array

In [11]:
```
arr2D = np.array([[1,2,3],[4,5,6],[7,8,9]])
arr2D
```

Out[11]:
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

In [13]:
```
# (row,col)
arr2D= np.arange(12).reshape(3,4)
arr2D
```

Out[13]:
```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11]])
```

## Step 4. Create a Linspace array

In [23]:
```
#auto data type float
#strting point,ending point,no. of output(apde ketla joi ch e)
#starting and ending point both are inclisive(bey array ma ave)
arr = np.linspace(0,5,10)
arr
```

Out[23]:
```
array([0.        , 0.55555556, 1.11111111, 1.66666667, 2.22222222,
       2.77777778, 3.33333333, 3.88888889, 4.44444444, 5.        ])
```

## Step 5. Create a Random Numbered Array

In [37]:
```
arr = np.random.rand(5)
arr
```

Out[37]:
```
array([0.92066394, 0.18001599, 0.27637333, 0.82266963, 0.04610274])
```

In [ ]:
```

```

## Step 6. Create a Random Integer Array

In [ ]:
```
# starting , ending , size
arr = np.random.randint(1,100,size=10)
arr
```

In [ ]:
```

```

## Step 7. Create a 1D Array and get Max,Min,ArgMax,ArgMin

In [39]:
```
arr = np.array([1,4,2,5,6,3])
arr
```

Out[39]:
```
array([1, 4, 2, 5, 6, 3])
```

In [41]:
```
arr = np.array([1,4,2,5,6,3])
arr.max()
```

Out[41]: 6

In [43]:
```
arr = np.array([1,4,2,5,6,3])
arr.min()
```

Out[43]: 1

In [47]:
```
#index ape and index 0 thi start thy
arr = np.array([1,4,2,5,6,3])
arr.argmax()
```

Out[47]: 4

In [49]:
```
arr = np.array([1,4,2,5,6,3])
arr.argmin()
```

Out[49]: 0

## Step 8. Indexing in 1D Array

In [55]:
```
arr = np.array([1,4,2,5,6,3])
arr[0]
```

Out[55]: 1

## Step 9. Indexing in 2D Array

In [57]:
```
arr2D= np.arange(12).reshape(3,4)
arr2D[1,0]
```

Out[57]: 4

## Step 10. Conditional Selection

In [63]:
```
arr= np.arange(12)
arr
arr[arr%2==0]
```

Out[63]: array([ 0,  2,  4,  6,  8, 10])

In [ ]:
```

```

## 🔥 You did it! 10 exercises down — you're on fire! 🔥

# Pandas

## Step 1. Import the necessary libraries

In [1]:
```
import pandas as pd
```

## Step 2. Import the dataset from this address.

## Step 3. Assign it to a variable called users and use the 'user_id' as index

In [3]:
```
df =  pd.read_csv("https://raw.githubusercontent.com/justmarkham/DAT8/master/dat
df
```

Out[3]:

| | user_id | age | gender | occupation | zip_code |
|---|---|---|---|---|---|
| **0** | 1 | 24 | M | technician | 85711 |
| **1** | 2 | 53 | F | other | 94043 |
| **2** | 3 | 23 | M | writer | 32067 |
| **3** | 4 | 24 | M | technician | 43537 |
| **4** | 5 | 33 | F | other | 15213 |
| **...** | ... | ... | ... | ... | ... |
| **938** | 939 | 26 | F | student | 33319 |
| **939** | 940 | 32 | M | administrator | 02215 |
| **940** | 941 | 20 | M | student | 97229 |
| **941** | 942 | 48 | F | librarian | 78209 |
| **942** | 943 | 22 | M | student | 77841 |

943 rows × 5 columns

## Step 4. See the first 25 entries

In [5]:
```
tp=df.head(25)
tp
```

Out[5]:

| | user_id | age | gender | occupation | zip_code |
|---|---|---|---|---|---|
| **0** | 1 | 24 | M | technician | 85711 |
| **1** | 2 | 53 | F | other | 94043 |
| **2** | 3 | 23 | M | writer | 32067 |
| **3** | 4 | 24 | M | technician | 43537 |
| **4** | 5 | 33 | F | other | 15213 |
| **5** | 6 | 42 | M | executive | 98101 |
| **6** | 7 | 57 | M | administrator | 91344 |
| **7** | 8 | 36 | M | administrator | 05201 |
| **8** | 9 | 29 | M | student | 01002 |
| **9** | 10 | 53 | M | lawyer | 90703 |
| **10** | 11 | 39 | F | other | 30329 |
| **11** | 12 | 28 | F | other | 06405 |
| **12** | 13 | 47 | M | educator | 29206 |
| **13** | 14 | 45 | M | scientist | 55106 |
| **14** | 15 | 49 | F | educator | 97301 |
| **15** | 16 | 21 | M | entertainment | 10309 |
| **16** | 17 | 30 | M | programmer | 06355 |
| **17** | 18 | 35 | F | other | 37212 |
| **18** | 19 | 40 | M | librarian | 02138 |
| **19** | 20 | 42 | F | homemaker | 95660 |
| **20** | 21 | 26 | M | writer | 30068 |
| **21** | 22 | 25 | M | writer | 40206 |
| **22** | 23 | 30 | F | artist | 48197 |
| **23** | 24 | 21 | F | artist | 94533 |
| **24** | 25 | 39 | M | engineer | 55107 |

## Step 5. See the last 10 entries

In [7]:
```
tp=df.tail(10)
tp
```

Out[7]:

| | user_id | age | gender | occupation | zip_code |
|---|---|---|---|---|---|
| **933** | 934 | 61 | M | engineer | 22902 |
| **934** | 935 | 42 | M | doctor | 66221 |
| **935** | 936 | 24 | M | other | 32789 |
| **936** | 937 | 48 | M | educator | 98072 |
| **937** | 938 | 38 | F | technician | 55038 |
| **938** | 939 | 26 | F | student | 33319 |
| **939** | 940 | 32 | M | administrator | 02215 |
| **940** | 941 | 20 | M | student | 97229 |
| **941** | 942 | 48 | F | librarian | 78209 |
| **942** | 943 | 22 | M | student | 77841 |

## Step 6. What is the number of observations in the dataset?

In [11]:
```
df["user_id"].count()
```

Out[11]:
```
943
```

## Step 7. What is the number of columns in the dataset?

In [17]:
```
# 1 is col
#0 is row
df.shape[1]
```

Out[17]:
```
5
```

## Step 8. Print the name of all the columns.

In [19]:
```
df.columns
```

Out[19]:
```
Index(['user_id', 'age', 'gender', 'occupation', 'zip_code'], dtype='object')
```

## Step 9. How is the dataset indexed?

In [21]:
```
df.index
```

Out[21]:
```
RangeIndex(start=0, stop=943, step=1)
```

## Step 10. What is the data type of each column?

In [29]:
```
df.dtypes
```

Out[29]:
```
user_id        int64
age            int64
gender        object
occupation    object
zip_code      object
dtype: object
```

## Step 11. Print only the occupation column

In [25]:
```
cl = df[["user_id","occupation"]]
cl
```

Out[25]:

| | user_id | occupation |
|---|---|---|
| **0** | 1 | technician |
| **1** | 2 | other |
| **2** | 3 | writer |
| **3** | 4 | technician |
| **4** | 5 | other |
| **...** | ... | ... |
| **938** | 939 | student |
| **939** | 940 | administrator |
| **940** | 941 | student |
| **941** | 942 | librarian |
| **942** | 943 | student |

943 rows × 2 columns

## Step 12. How many different occupations are in this dataset?

In [45]:
```
df["occupation"].nunique()
```

Out[45]:
```
21
```

## Step 13. What is the most frequent occupation?

In [51]:
```
#mode is give to ocuurences
#when i give enter a 0 so give me a first occurences
df["occupation"].mode()[0]
```

Out[51]:
```
'student'
```

## Step 14. Summarize the DataFrame.

In [39]:
```
tp=df.describe()
tp
```

Out[39]:

| | user_id | age |
|---|---|---|
| **count** | 943.000000 | 943.000000 |
| **mean** | 472.000000 | 34.051962 |
| **std** | 272.364951 | 12.192740 |
| **min** | 1.000000 | 7.000000 |
| **25%** | 236.500000 | 25.000000 |
| **50%** | 472.000000 | 31.000000 |
| **75%** | 707.500000 | 43.000000 |
| **max** | 943.000000 | 73.000000 |

## Step 15. Summarize all the columns

In [53]:
```
tp=df.info()
tp
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 943 entries, 0 to 942
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   user_id     943 non-null    int64 
 1   age         943 non-null    int64 
 2   gender      943 non-null    object
 3   occupation  943 non-null    object
 4   zip_code    943 non-null    object
dtypes: int64(2), object(3)
memory usage: 37.0+ KB
```

## Step 16. Summarize only the occupation column

In [55]:
```
df["occupation"].info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 943 entries, 0 to 942
Series name: occupation
Non-Null Count  Dtype 
--------------  ----- 
943 non-null    object
dtypes: object(1)
memory usage: 7.5+ KB
```

## Step 17. What is the mean age of users?

In [59]:
```
df["age"].mean()
```

Out[59]: 34.05196182396607

## Step 18. What is the age with least occurrence?

In [61]:
```
counts = df['age'].value_counts()

least_common = counts[counts == counts.min()]
least_common
```

Out[61]:
```
age
7     1
66    1
11    1
10    1
73    1
Name: count, dtype: int64
```

## You're not just learning, you're mastering it. Keep aiming higher! 🚀