(https://www.darshan.ac.in/)

Data Mining

Lab - 2

Name : Gohel Krish Vishalbhai

ENR NO . : 22010101060

Roll No. : 216

Batch : A6

Step 1. Import the necessary libraries

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

Step 2. Import the dataset from this address (https://raw.githubusercontent.com/justmarkham/DAT8/master/data/u.user).

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

In [6]:
```
users = pd.read_csv('https://raw.githubusercontent.com/justmarkham/DAT8/master/data/u.user',sep='|',index_col='user_id
```

Step 4. See the first 25 entries

In [7]:
```
users.head(25)
```

```
13  47  M       educator  29206
14  45  M      scientist  55106
15  49  F       educator  97301
16  21  M  entertainment  10309
17  30  M     programmer  06355
18  35  F          other  37212
19  40  M      librarian  02138
20  42  F      homemaker  95660
21  26  M         writer  30068
22  25  M         writer  40206
23  30  F         artist  48197
24  21  F         artist  94533
25  39  M       engineer  55107
```

Step 5. See the last 10 entries

In [8]:
```
users.tail(10)
```

Out[8]:

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

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

In [13]:
```
# len(users)

users.shape[0] # return the num. of obs. and length of row
```

Out[13]:
```
943
```

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

In [15]:
```
users.shape[1] # return the num. of columns in dataset
```

Out[15]:
```
4
```

Step 8. Print the name of all the columns.

In [18]:
```
users.columns
```

Out[18]:
```
Index(['age', 'gender', 'occupation', 'zip_code'], dtype='object')
```

Step 9. How is the dataset indexed?

In [19]:
```
# "the index" (aka "the labels")
users.index
```

Out[19]:
```
Index([  1,   2,   3,   4,   5,   6,   7,   8,   9,  10,
       ...
       934, 935, 936, 937, 938, 939, 940, 941, 942, 943],
      dtype='int64', name='user_id', length=943)
```

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

In [25]:
```
users.dtypes
```

Out[25]:
```
age            int64
gender        object
occupation    object
zip_code      object
dtype: object
```

Step 11. Print only the occupation column

In [26]:
```
users['occupation']
```

Out[26]:
```
user_id
1         technician
2              other
3             writer
4         technician
5              other
           ...
939          student
940    administrator
941          student
942        librarian
943          student
Name: occupation, Length: 943, dtype: object
```

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

In [28]:
```
# len(users['occupation'].unique())
users['occupation'].nunique()
```

Out[28]:
```
21
```

Step 13. What is the most frequent occupation?

In [31]:
```
users['occupation'].value_counts().keys()[0]
```

Out[31]:
```
'student'
```

Step 14. Summarize the DataFrame.

In [35]:
```
users.describe()
```

Out[35]:

	age
count	943.000000
mean	34.051962
std	12.192740
min	7.000000
25%	25.000000
50%	31.000000
75%	43.000000
max	73.000000

Step 15. Summarize all the columns

In [38]:
```
users.describe(include='all')
```

Out[38]:

	age	gender	occupation	zip_code
count	943.000000	943	943	943
unique	NaN	2	21	795
top	NaN	M	student	55414
freq	NaN	670	196	9
mean	34.051962	NaN	NaN	NaN
std	12.192740	NaN	NaN	NaN
min	7.000000	NaN	NaN	NaN
25%	25.000000	NaN	NaN	NaN
50%	31.000000	NaN	NaN	NaN
75%	43.000000	NaN	NaN	NaN
max	73.000000	NaN	NaN	NaN

Step 16. Summarize only the occupation column

In [41]:
```
users['occupation'].describe()
```

Out[41]:
```
count         943
unique         21
top       student
freq          196
Name: occupation, dtype: object
```

Step 17. What is the mean age of users?

In [42]:
```
users['age'].mean()
```

Out[42]:
```
34.05196182396607
```

Step 18. What is the age with least occurrence?

In [44]:
```
users['age'].value_counts().tail()
```

Out[44]:
```
age
7     1
66    1
11    1
10    1
73    1
Name: count, dtype: int64
```