UNIVERSITY OF SOUTH DAKOTA

Assignment 03

CSC585- Data Mining Methods

September 27, 2021

Md Minhazur Rahman

ID: 101085958

MS in Physics-Analytics for Large Data Sets

Department of Physics,

University of South Dakota.

1. What is the role of a data analyst, a data engineer, and a data scientist. Describe each position. Describe how they compliment each other?

   Role of Data Analyst:
   - ⇒ Using advanced computerised models to extract the data needed.
   - ⇒ Removing corrupted data.
   - ⇒ Performing initial analysis to assess the quality of the data.
   - ⇒ Performing further analysis to determine the meaning of the data.
   - ⇒ Performing final analysis to provide additional data screening.
   - ⇒ Preparing reports based on analysis and presenting to management.

   Role of Data Scientist:
   - ⇒ Needs to determine how to use business data for important business decision.
   - ⇒ Tries to find out new data source and determine their accuracy.
   - ⇒ Analyze databases to simplify and improve product development, marketing techniques, and business processes.
   - ⇒ Create new data models and algorithms.
   - ⇒ Use predictive models to improve customer experience, making promotional offers, revenue generation, and more.
   - ⇒ Coordinate with various technical/functional teams to implement models and monitor results.

   Role of Data Engineer:
   - ⇒ Make data accessible so that organizations can use it to better their performance.
   - ⇒ Collection and management of data, converting it into useful information.
   - ⇒ Build and maintain data pipelines and maintain databases.
   - ⇒ Collaboration with management to perceive organizational goals.
   - ⇒ Creation of new data validation processes and analytical tools.
   - ⇒ Design, build, test, and maintain data management systems.

In [1]:
```
import numpy as np
import numpy.random as rn
import math
import functools
import pandas as pd
```

2. Use the Python random library (not the numpy one) to create a list of 20 random numbers. Provide the code snippet and output.

In [13]:
```
rand_nums = rn.rand(20)
rand_nums
```

Out[13]:
```
array([0.91776255, 0.47702682, 0.38853017, 0.77706385, 0.97167234,
       0.74281808, 0.81091842, 0.66708782, 0.71603537, 0.32609375,
       0.4447418 , 0.52699454, 0.15439155, 0.5731983 , 0.37102435,
       0.24136848, 0.43508799, 0.83190474, 0.10161592, 0.76184039])
```

3. Sort the list from question two and return into a new list leaving the original list unchanged. Provide the code snippet and output.

In [16]:
```
sorted_rand_nums = rand_nums.copy()
sorted_rand_nums.sort()
sorted_rand_nums
```

Out[16]:
```
array([0.10161592, 0.15439155, 0.24136848, 0.32609375, 0.37102435,
       0.38853017, 0.43508799, 0.4447418 , 0.47702682, 0.52699454,
       0.5731983 , 0.66708782, 0.71603537, 0.74281808, 0.76184039,
       0.77706385, 0.81091842, 0.83190474, 0.91776255, 0.97167234])
```

4. Use a list comprehension to compute a new list from the output of question two which contains the square of each element. Provide the code snippet and output.

In [17]:
```
sq_rand_nums = [x**2 for x in rand_nums]
sq_rand_nums
```

Out[17]:
```
[0.8422880954537756,
 0.22755458820801036,
 0.1509556917042688,
 0.6038282328883188,
 0.9441471438308892,
 0.5517786978236987,
 0.6575886839510597,
 0.4450061643640957,
 0.5127066452946052,
 0.10633713123927047,
 0.19779527099990024,
 0.2777232452483508,
 0.02383674931412893,
 0.32855628916168433,
 0.1376590691467966,
 0.05825874082710794,
 0.18930156189916336,
 0.6920654898814224,
```

```
 0.01032579466800494,
 0.5804007748039282]
```

5. Write a function which computes the square root of it's input and returns the value. Use a list comprehension to compute a new list containing the square root of each element of the list created in question four. Provide the code snippet and output.

In [48]:
```
def find_sqrt(x):
    return math.sqrt(x)
sqrt_rand_nums = [find_sqrt(x) for x in sq_rand_nums]
sqrt_rand_nums
```

Out[48]:
```
[0.9177625485133808,
 0.47702682126690776,
 0.3885301683322272,
 0.7770638538037391,
 0.9716723438643755,
 0.7428180785520091,
 0.8109184200343829,
 0.6670878235765481,
 0.7160353659524125,
 0.3260937460904003,
 0.4447418026224882,
 0.5269945400555406,
 0.15439154547490264,
 0.5731982982892433,
 0.3710243511506982,
 0.24136847521395155,
 0.4350879932831557,
 0.8319047360614209,
 0.10161591739488918,
 0.761840386697849]
```

6. Use the reduce function (see the Python documentation) to compute the sum of the elements of the list created in question five. Provide the code snippet and output.

In [27]:
```
functools.reduce(lambda a,b: a+b, sqrt_rand_nums)
```

Out[27]:
```
11.237177216230524
```

7. Create a pandas DataFrame (DF) from a 40 x 3 (row by column) numpy array generated using the random module of numpy. Once in a DF, multiply each element by 100. After the multiplication, make sure the data has been changed in the DF! Provide the code snippet and output.

In [38]:
```
rn.rand(40,3)
df = pd.DataFrame(rn.rand(40,3))
df = df*100
df.head()
```

Out[38]:

	0	1	2
0	17.722665	93.994028	2.055120
1	69.852817	96.296823	71.869591

	0	1	2
2	75.715306	34.705907	75.408260
3	2.512994	33.748453	74.533816
4	4.096581	63.599617	60.565771

8. Compute the sum of the three columns for each of the forty rows. Provide the code snippet and output. Store the sum as a new column in the DF from question seven. Provide the code snippet and output.

In [40]:
```
df[4] = df[0]+df[1]+df[2]
df.head()
```

Out[40]:

	0	1	2	4
0	17.722665	93.994028	2.055120	113.771813
1	69.852817	96.296823	71.869591	238.019231
2	75.715306	34.705907	75.408260	185.829473
3	2.512994	33.748453	74.533816	110.795263
4	4.096581	63.599617	60.565771	128.261969

9. Create a new DF from the DF created in question eight which contains all rows (include all four columns) where the sum column is greater than 125. Provide the code snippet and output.

In [45]:
```
df2 = df[df[4]>125]
df2.head()
```

Out[45]:

	0	1	2	4
1	69.852817	96.296823	71.869591	238.019231
2	75.715306	34.705907	75.408260	185.829473
4	4.096581	63.599617	60.565771	128.261969
6	83.142592	85.209122	4.375873	172.727586
8	68.882328	61.708167	43.126675	173.717170