Graded Questions (Unsupervised Learning - PCA)

1. Fundamentals of PCA – I
2. Fundamentals of PCA – II

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

1. The Algorithm of PCA
2. PCA In Python

**Graded Questions**

The following .csv file contains the same data you used while learning linear regression. All data has been thoroughly cleaned in the dataset provided. The dataset contains 16 columns. The column ‘Price’ is the dependent variable, and the rest of the columns are independent variables and are quite self-explanatory. All columns have numerical values. There is a total of 545 data points.

**Also, please round off all the values till 2 decimal places for answering the questions even if not explicitly mentioned.**

Please **use the following code file** to solve the graded questions. Fill in the code at relevant places to get the answers. There are two sections in the code, one for questions 1 and 2; other for question 4.

**[Code\_for\_Graded\_Questions\_PCA](https://cdn.upgrad.com/UpGrad/temp/98ae450c-a9ec-424e-83db-9b55e3bbde03/Graded_Questions_PCA.ipynb" \o "Graded_Questions_PCA.ipynb" \t "_blank)**

[file\_download](https://cdn.upgrad.com/UpGrad/temp/98ae450c-a9ec-424e-83db-9b55e3bbde03/Graded_Questions_PCA.ipynb" \o "Graded_Questions_PCA.ipynb" \t "_blank)**[Download](https://cdn.upgrad.com/UpGrad/temp/98ae450c-a9ec-424e-83db-9b55e3bbde03/Graded_Questions_PCA.ipynb" \o "Graded_Questions_PCA.ipynb" \t "_blank)**

Please download the .csv data file here.

**[newhousing](https://cdn.upgrad.com/UpGrad/temp/a25c1659-e922-4f4e-ad16-cb430ab01e61/newhousing.csv" \o "newhousing.csv" \t "_blank)**

[file\_download](https://cdn.upgrad.com/UpGrad/temp/a25c1659-e922-4f4e-ad16-cb430ab01e61/newhousing.csv" \o "newhousing.csv" \t "_blank)**[Download](https://cdn.upgrad.com/UpGrad/temp/a25c1659-e922-4f4e-ad16-cb430ab01e61/newhousing.csv" \o "newhousing.csv" \t "_blank)**

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Coding Graded:

### Rounded Sum

**Description**

You're given a list of non-negative integers. Your task is to round the given numbers to the nearest multiple of 10. For instance, 15 should be rounded to 20 whereas 14 should be rounded to 10. After rounding the numbers, find their sum.   
  
Hint: The Python pre-defined function round() rounds off to nearest even number - it round 0.25 to 0.2. You might want to write your own function to round as per your requirement.  
  
Sample input (a list):  
[2, 18, 10]  
  
Sample output (an integer):  
30

Solution:

import ast,sys

import math

input\_str = sys.stdin.read()

input\_list = ast.literal\_eval(input\_str)

# write code here

def roundup(n) :

a = (n//10) \*10

b = a+10

return (b if n - a >= b - n else a)

# store the sum in the following variable

result = 0

for i in input\_list:

val = roundup(i)

result = result +val

# do not change the following code

print(result)

### Alarm Clock

**Description**

You're trying to automate your alarm clock by writing a function for it. You're given a day of the week encoded as 1=Mon, 2=Tue, ... 6=Sat, 7=Sun, and a boolean value (a boolean object is either True or False. Google "booleans python" to get a better understanding) indicating if you're are on vacation. Based on the day and whether you're on vacation, write a function that returns a time in form of a string indicating when the alarm clock should ring.   
  
When not on a vacation, on weekdays, the alarm should ring at "7:00" and on the weekends (Saturday and Sunday) it should ring at "10:00".   
  
While on a vacation, it should ring at "10:00" on weekdays. On vacation, it should not ring on weekends, that is, it should return "off".  
  
----------------------------------------------------------------------  
Sample input (a list):  
[7,True]  
  
Sample output (a string):  
off  
----------------------------------------------------------------------  
Sample input (a list):  
[3,True]  
  
Sample output (a string):  
10:00  
----------------------------------------------------------------------

Solution:

import ast,sys

input\_str = sys.stdin.read()

input\_list = ast.literal\_eval(input\_str)

day\_of\_the\_week = input\_list[0]

is\_on\_vacation = input\_list[1]

# write your code here

def alarm\_clock(day, vacation):

weekend = int(day) in (6,7)

if weekend and vacation:

return 'off'

elif weekend or vacation:

return '10:00'

return '7:00'

# do not change the following code

time = alarm\_clock(day\_of\_the\_week, is\_on\_vacation)

print(time.lower())

### Sum and Squares

**Description**

You're given a natural number 'n'. First, calculate the sum of squares of all the natural numbers up to 'n'. Then calculate the square of the sum of all natural numbers up to 'n'. Return the absolute difference of these two quantities.  
   
For instance, if n=3, then natural numbers up to 3 are: 1, 2 and 3. The sum of squares up to 3 will be 1^2 + 2^2 + 3^2 = 14. The square of the sum of natural numbers up to 3 is (1+2+3)^2=36. The result, which is their absolute difference is 22.  
  
  
Sample input (an integer):  
3  
  
Sample output (an integer):  
22

Solution:

import ast,sys

input\_str = sys.stdin.read()

n = ast.literal\_eval(input\_str)

# write your code here

x = [z+1 for z in range(n)]

sumofsquares = sum(list(map(lambda y: y\*\*2, x)))

squareofsum = sum(x)\*\*2

# store the result in the following variable

abs\_difference = abs(sumofsquares - squareofsum)

# print result --- do not change the following code

print(abs\_difference)