CS5590/490 - Python and Deep Learning Programming
In Class Programming Report - 3
Class ID 24 - Anurag Thantharate
Date Submitted: 02/09/2020

Video Link: https://www.loom.com/share/88d66a59cab442898ab1ff616ef72c95

---

1. Create a class Employee and then do the following
    a. Create a data member to count the number of Employees
    b. Create a constructor to initialize name, family, salary, department
    c. Create a function to average salary
    d. Create a Fulltime Employee class and it should inherit the properties of Employee class
    e. Create the instances of Fulltime Employee class and Employee class and call their member functions.

Spring 2020 〉 Python with DL 〉 ICP3 〉 employee.py

employee.py

```python
# Defining class 'Employee'
class Employee:
    empCount = 0  # variable assigned (data member) to count the employees
    totalSalary = 0

    # constructor to initialize name, family, salary, department
    def __init__(self, name, family, salary, department):
        self.name = name
        self.family = family
        self.salary = salary
        self.department = department
        self.__class__.empCount += 1
        self.__class__.totalSalary = Employee.totalSalary + salary

    # Create function for average salary
    def avg_salary(self):
        avg_Salary = (Employee.totalSalary / Employee.empCount)
        return avg_Salary

# Create instances for passing employee data
emp1 = Employee('Andy', 'T', 3000, 'IT')
```

Employee 〉 __init__()

Run: employee

```
C:\ProgramData\Anaconda3\python.exe "C:/Users/1d630534/Documents/Anurag Projects/Spring 2020/Python with DL/ICP3/employee.py"
Andy
Brett
Jondy
Kristin
Dale
Average Salary of Consultant (Not Full-time) 4000.0
Average Salary of Full time Employee 6000.0
Total no. of Employees in the Company are 5
```

Spring 2020 〉 Python with DL 〉 ICP3 〉 employee.py

employee.py

```python
        avg_Salary = (Employee.totalSalary / Employee.empCount)
        return avg_Salary

# Create instances for passing employee data
emp1 = Employee('Andy', 'T', 3000, 'IT')
emp2 = Employee('Brett', 'W', 4000, 'Finance')
emp3 = Employee('Jondy', 'A', 5000, 'Development')

# Defining sub-class for full time employee inheriting Employee class functions
class FulltimeEmp(Employee):
    FTEcount = 0
    FTEtotalSalary = 0

    def __init__(self, name, family, salary, department):
        Employee.__init__(self, name, family, salary, department)
        self.__class__.FTEcount += 1
        self.__class__.FTEtotalSalary = FulltimeEmp.FTEtotalSalary + salary

    # Creating a member function associated with full time employees class
    def fulltimeCount(self):
        fulltimeCount = FulltimeEmp.FTEcount
        return fulltimeCount

    # Create function to call average salary
    def avgFTEsalary(self):
        avgFTEsalary = (FulltimeEmp.FTEtotalSalary / FulltimeEmp.FTEcount)
        return avgFTEsalary

# Create instances (passing arguments) of the full time employee and calling their member functions
emp4 = FulltimeEmp('Kristin', 'Y', 6000, 'Product')
emp5 = FulltimeEmp('Dale', 'W', 6000, 'PMO')
```

```python
# Create instances (passing arguments) of the full time employee and calling their member functions
emp4 = FulltimeEmp('Kristin', 'Y', 6000, 'Product')
emp5 = FulltimeEmp('Dale', 'W', 6000, 'PMO')

print(emp1.name)
print(emp2.name)
print(emp3.name)
print(emp4.name)
print(emp5.name)
(emp3.avg_salary())
print("Average Salary of Consultant (Not Full-time)", emp3.avg_salary())
(emp4.avgFTEsalary())
print("Average Salary of Full time Employee", emp4.avgFTEsalary())

# Calling full time employee function to identify the number of employees
totalEmp = emp5.fulltimeCount() + emp1.empCount
print("Total no. of Employees in the Company are ", totalEmp)
```

---

2. Web Scraping - Write a simple program that parse a Wiki page mentioned below and follow the instructions: https://en.wikipedia.org/wiki/Deep_learning
   a. Print out the title of the page
   b. Find all the links in the page ('a' tag)
   c. Iterate over each tag(above) then return the link using attribute "href" using get

Spring 2020 〉 Python with DL 〉 ICP3 〉 webscraping.py

webscraping.py

```python
# Importing requests library to send HTTP requests
# Parsing data using BeautifulSoup function
import requests
from bs4 import BeautifulSoup

#Parsing the webpage
webpage = "https://en.wikipedia.org/wiki/Deep_learning"
Parsedpage = requests.get(webpage).text
soup = BeautifulSoup(Parsedpage,"html.parser")

# Print the title of the web page
title = soup.title
print(title)

# Finding all the links within the page containing a tag
Tag = soup.find_all("a")
for link in Tag:
    print(link.get("href"))
```

for link in Tag

Run: webscraping

```
C:\ProgramData\Anaconda3\python.exe "C:/Users/1d630534/Documents/Anurag Projects/Spring 2020/Python with DL/ICP3/webscraping.py"
<title>Deep learning - Wikipedia</title>
None
#mw-head
#p-search
/wiki/Student_approaches_to_learning
/wiki/Artificial_neural_network
/wiki/Machine_learning
/wiki/Data_mining
/wiki/File:Kernel_Machine.svg
/wiki/Statistical_classification
/wiki/Cluster_analysis
```

---

3. Using NumPy create random vector of size 15 having only Integers in the range 1-20.
    a. Then reshape the array to 3 by 5.
    b. Then replace the max in each row by 0. You can NOT implement it via for loop. You need to use np.where, reshape)

numpie.py

```python
import numpy as np

#Generate random integers
x = np.random.randint(1, 20, 15)
print("List of randomly generated integers: ", x)

#Reshaping randomly generated integers into 3X5 Matrix
a = x.reshape(3, 5)
print("Reshape generated arrays into 3X5 Matrix")
print(a)
a[np.where(a==np.max(a))] = 0
print("Maximum value replaced by 0:")
print(a)
```

Run: numpie (1)

```
List of randomly generated integers:  [14  5  9 15  3 19  9  1  4  2  1  3 19 16 12]
Reshape generated arrays into 3X5 Matrix
[[14  5  9 15  3]
 [19  9  1  4  2]
 [ 1  3 19 16 12]]
Maximum value replaced by 0:
[[14  5  9 15  3]
 [ 0  9  1  4  2]
 [ 1  3  0 16 12]]

Process finished with exit code 0
```
