ICP-3 Neural Networks & Deep Learning

V V S Murthy KOLLA

700729142

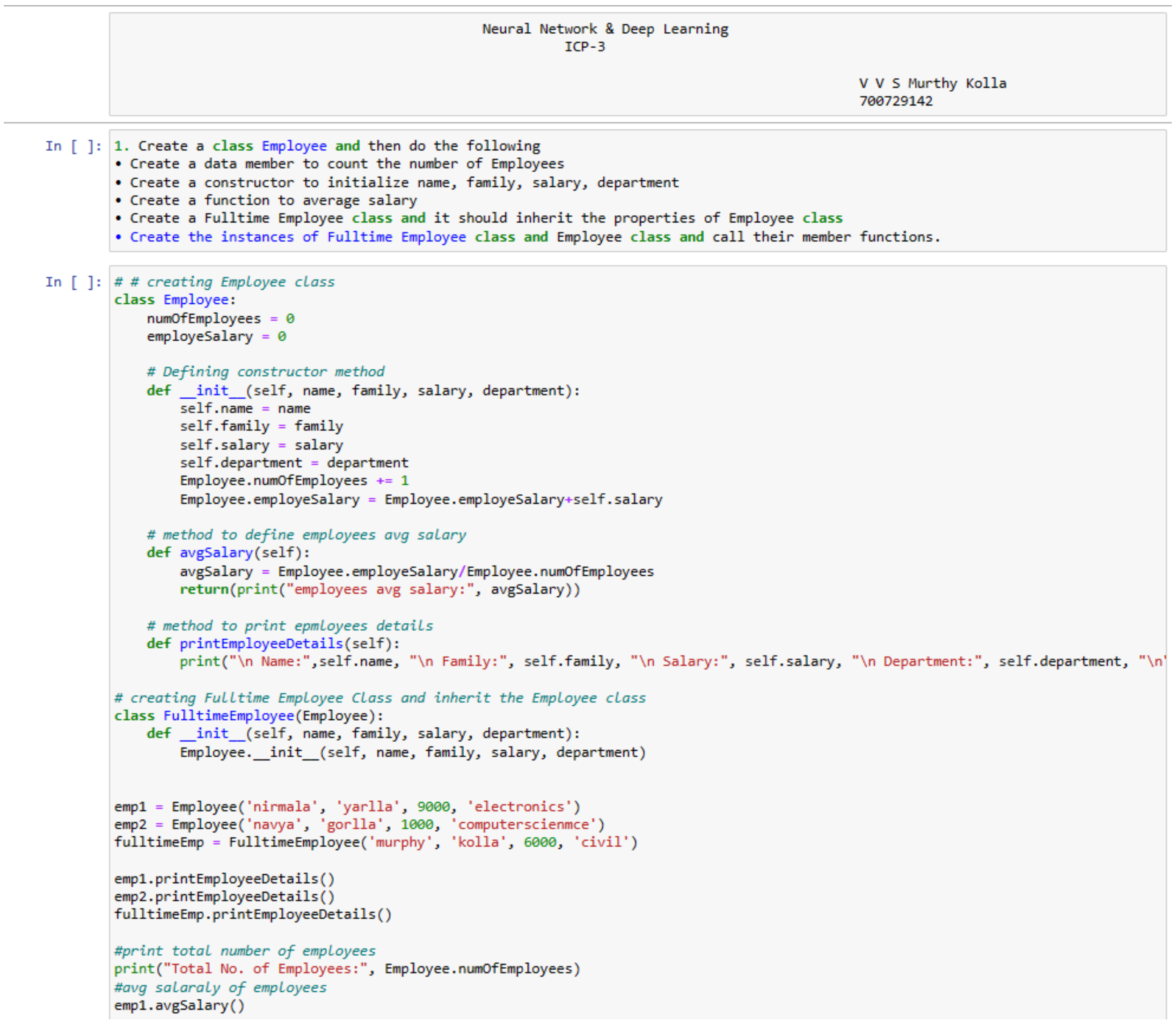
1.Create a class Employee and then do the following • Create a data member to count the number of Employees

• Create a constructor to initialize name, family, salary, department

• Create a function to average salary

• Create a Fulltime Employee class and it should inherit the properties of Employee class

• Create the instances of Fulltime Employee class and Employee class and call their member functions

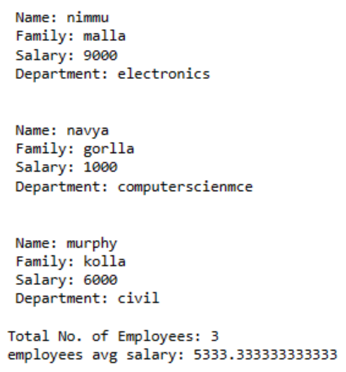


```
                    Neural Network & Deep Learning
                                ICP-3

                                                    V V S Murthy Kolla
                                                    700729142
```

In [ ]:
```
1. Create a class Employee and then do the following
• Create a data member to count the number of Employees
• Create a constructor to initialize name, family, salary, department
• Create a function to average salary
• Create a Fulltime Employee class and it should inherit the properties of Employee class
• Create the instances of Fulltime Employee class and Employee class and call their member functions.
```

In [ ]:
```python
# # creating Employee class
class Employee:
    numOfEmployees = 0
    employeSalary = 0

    # Defining constructor method
    def __init__(self, name, family, salary, department):
        self.name = name
        self.family = family
        self.salary = salary
        self.department = department
        Employee.numOfEmployees += 1
        Employee.employeSalary = Employee.employeSalary+self.salary

    # method to define employees avg salary
    def avgSalary(self):
        avgSalary = Employee.employeSalary/Employee.numOfEmployees
        return(print("employees avg salary:", avgSalary))

    # method to print epmloyees details
    def printEmployeeDetails(self):
        print("\n Name:",self.name, "\n Family:", self.family, "\n Salary:", self.salary, "\n Department:", self.department, "\n'

# creating Fulltime Employee Class and inherit the Employee class
class FulltimeEmployee(Employee):
    def __init__(self, name, family, salary, department):
        Employee.__init__(self, name, family, salary, department)


emp1 = Employee('nirmala', 'yarlla', 9000, 'electronics')
emp2 = Employee('navya', 'gorlla', 1000, 'computerscienmce')
fulltimeEmp = FulltimeEmployee('murphy', 'kolla', 6000, 'civil')

emp1.printEmployeeDetails()
emp2.printEmployeeDetails()
fulltimeEmp.printEmployeeDetails()

#print total number of employees
print("Total No. of Employees:", Employee.numOfEmployees)
#avg salaraly of employees
emp1.avgSalary()
```

Output:



```
Name: nimmu
Family: malla
Salary: 9000
Department: electronics


Name: navya
Family: gorlla
Salary: 1000
Department: computerscienmce


Name: murphy
Family: kolla
Salary: 6000
Department: civil

Total No. of Employees: 3
employees avg salary: 5333.333333333333
```

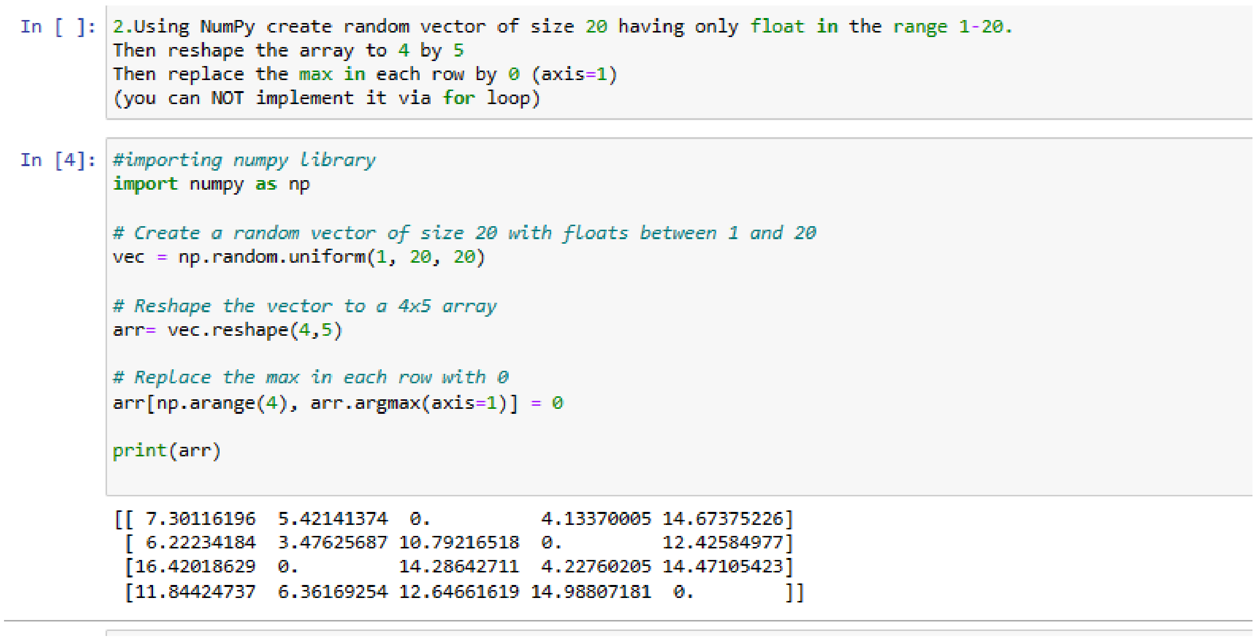
2. NumPy

Using NumPy create random vector of size 20 having only float in the range 1-20.

Then reshape the array to 4 by 5

Then replace the max in each row by 0 (axis=1)

(you can NOT implement it via for loop)



```
In [ ]: 2.Using NumPy create random vector of size 20 having only float in the range 1-20.
        Then reshape the array to 4 by 5
        Then replace the max in each row by 0 (axis=1)
        (you can NOT implement it via for loop)

In [4]: #importing numpy library
        import numpy as np

        # Create a random vector of size 20 with floats between 1 and 20
        vec = np.random.uniform(1, 20, 20)

        # Reshape the vector to a 4x5 array
        arr= vec.reshape(4,5)

        # Replace the max in each row with 0
        arr[np.arange(4), arr.argmax(axis=1)] = 0

        print(arr)

        [[ 7.30116196  5.42141374  0.          4.13370005 14.67375226]
         [ 6.22234184  3.47625687 10.79216518  0.         12.42584977]
         [16.42018629  0.         14.28642711  4.22760205 14.47105423]
         [11.84424737  6.36169254 12.64661619 14.98807181  0.        ]]
```

GitHub link: https://github.com/murthykolla/ICP--3-ASSIGN.git

Video link: https://drive.google.com/file/d/1_jWD30tUrYaEbwMm-3gPpIf0oCQcMRiP/view?usp=sharing