

NNDL: ICP2

Jyothi Kiran Boddada
700769023

GITHUB LINK : <https://github.com/jyothikiranboddada/Neural-Network-Deep-Learning.git>

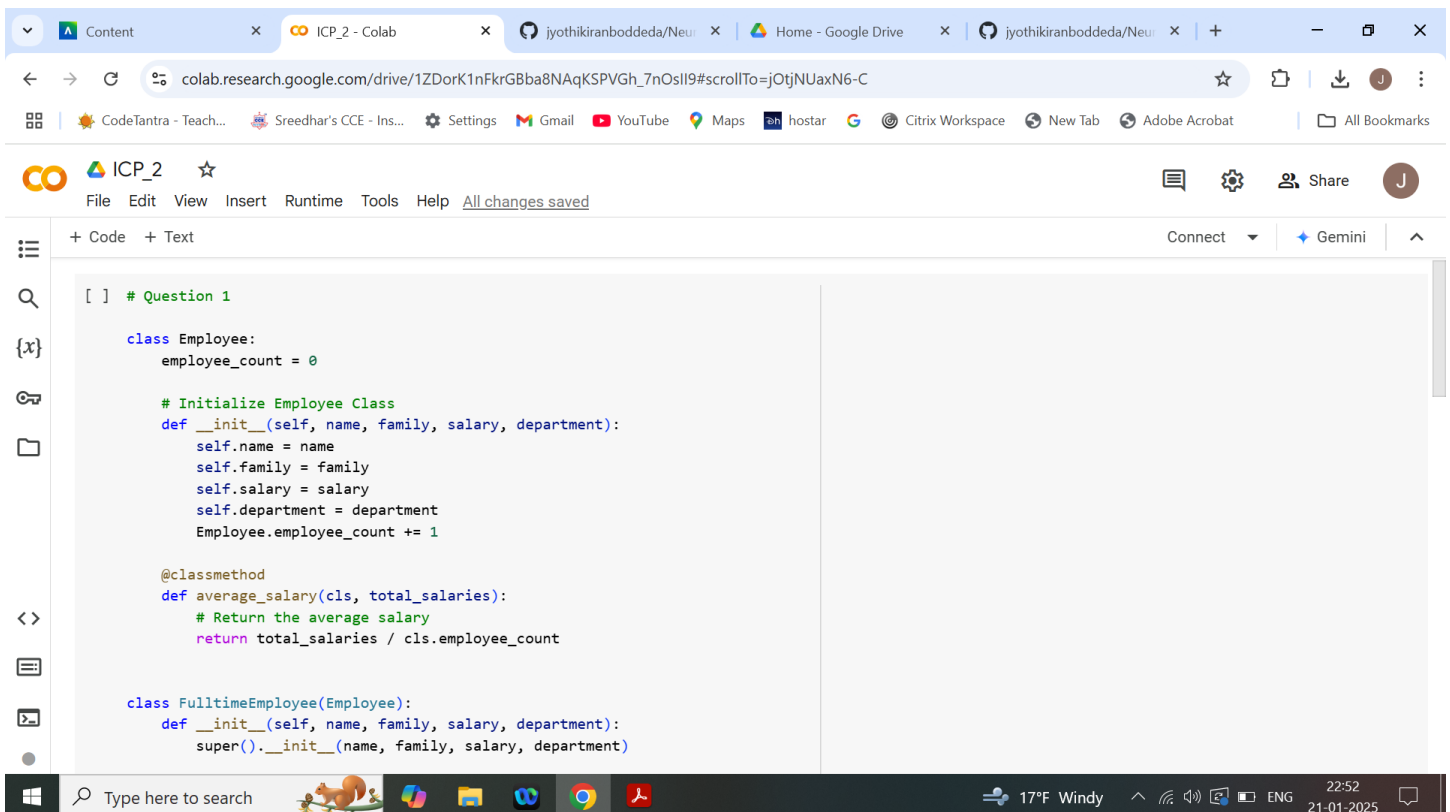
VIDEO LINK :

https://drive.google.com/file/d/1hACyICixftEk_v_veHCA00MrZJ2OUExw/view?usp=drive_link

In class programming:

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



```
[ ] # Question 1

class Employee:
    employee_count = 0

    # Initialize Employee Class
    def __init__(self, name, family, salary, department):
        self.name = name
        self.family = family
        self.salary = salary
        self.department = department
        Employee.employee_count += 1

    @classmethod
    def average_salary(cls, total_salaries):
        # Return the average salary
        return total_salaries / cls.employee_count

class FulltimeEmployee(Employee):
    def __init__(self, name, family, salary, department):
        super().__init__(name, family, salary, department)
```

ICP_2

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Connect Gemini

```
[ ]
class FulltimeEmployee(Employee):
    def __init__(self, name, family, salary, department):
        super().__init__(name, family, salary, department)

# Create instances of Employee and FulltimeEmployee
emp1 = Employee("Kiran", "kumar", 50000, "IT")
emp2 = Employee("Srikanth", "pasam", 60000, "HR")
full_time_emp1 = FulltimeEmployee("sai", "Brown", 90000, "finance")
full_time_emp2 = FulltimeEmployee("abhi", "ram", 80000, "Marketing")

# Calculate total salary
total_salary = emp1.salary + emp2.salary + full_time_emp1.salary + full_time_emp2.salary

# Display the average salary
print(f"Average Salary: {Employee.average_salary(total_salary)}")

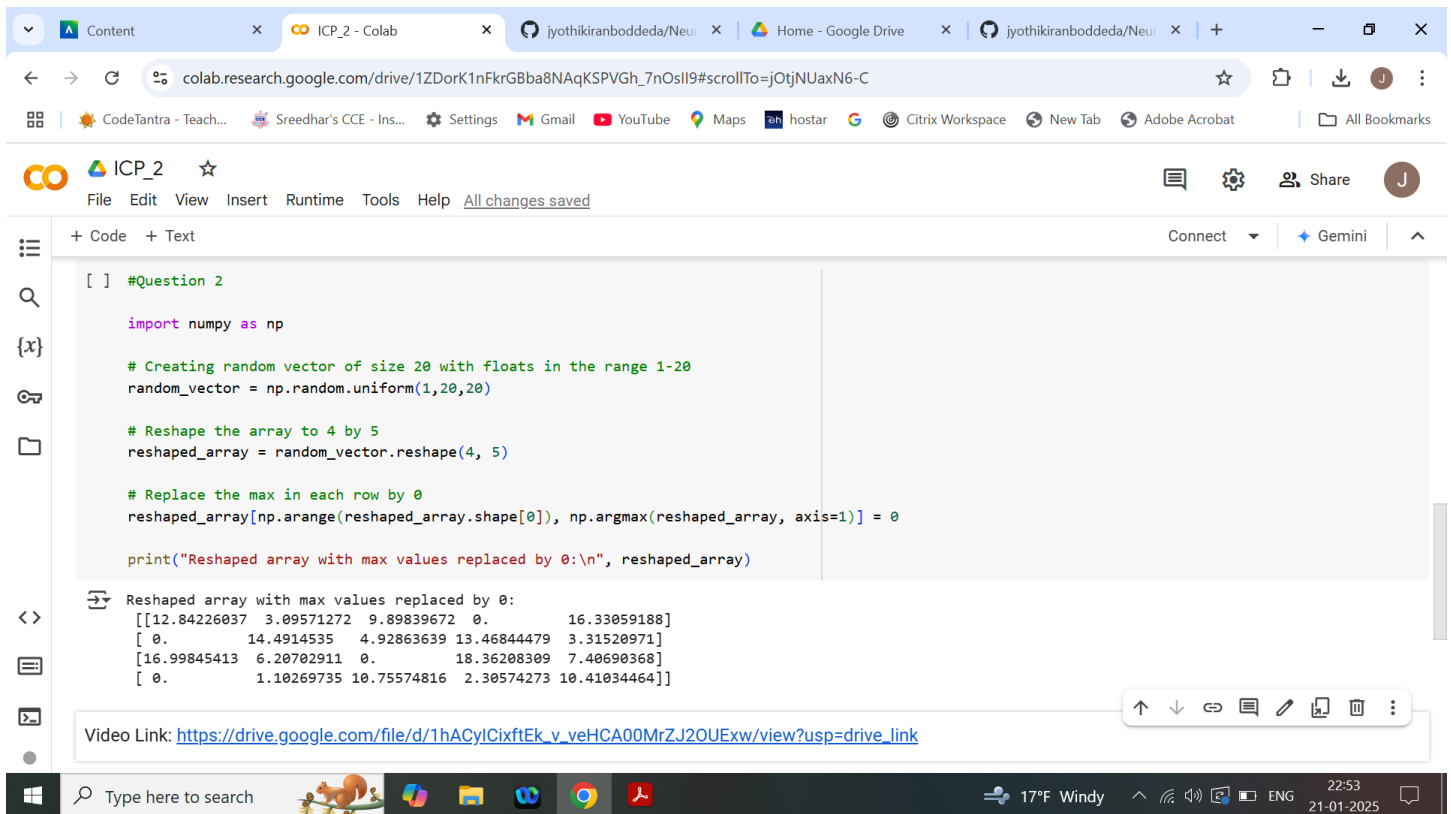
# Display the total number of employees
print(f"Total Employees: {Employee.employee_count}")
```

Average Salary: 70000.0
Total Employees: 4

Type here to search 17°F Windy 22:52 21-01-2025

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)



```
[ ] #Question 2

import numpy as np

# Creating random vector of size 20 with floats in the range 1-20
random_vector = np.random.uniform(1,20,20)

# Reshape the array to 4 by 5
reshaped_array = random_vector.reshape(4, 5)

# Replace the max in each row by 0
reshaped_array[np.arange(reshaped_array.shape[0]), np.argmax(reshaped_array, axis=1)] = 0

print("Reshaped array with max values replaced by 0:\n", reshaped_array)
```

Reshaped array with max values replaced by 0:

```
[[12.84226037  3.09571272  9.89839672  0.          16.33059188]
 [ 0.          14.4914535   4.92863639 13.46844479  3.31520971]
 [16.99845413  6.20702911  0.          18.36208309  7.40690368]
 [ 0.          1.10269735 10.75574816  2.30574273 10.41034464]]
```

Video Link: https://drive.google.com/file/d/1hACyICixftEk_v_eHCA00MrZJ2OUEw/view?usp=drive_link