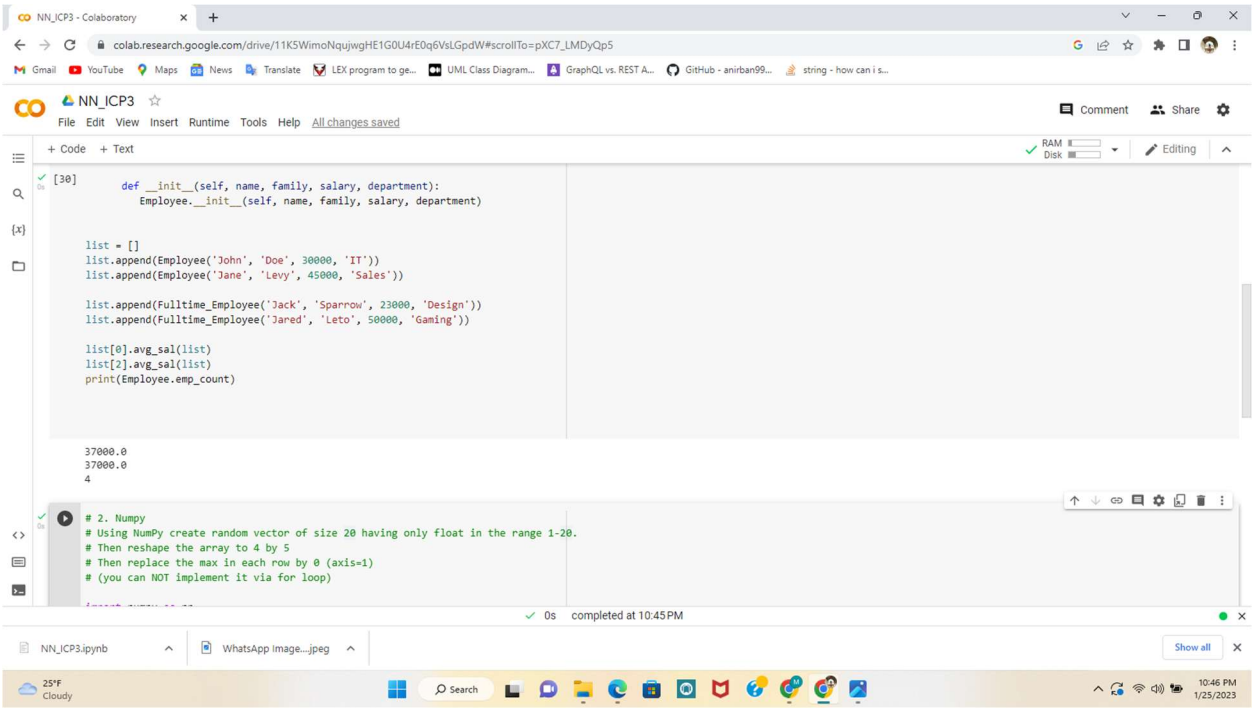
NEURAL_NETWORKS

ICP-3

```
# 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.

class Employee:
  emp_count = 0

  def __init__(self, name, family, salary, department):

        self.name = name
        self.family = family
        self.salary = salary
        self.department = department
        Employee.emp_count = Employee.emp_count + 1
  def avg_sal(self, emps):
      sum_sal = 0
      for i in emps:
            sum_sal= sum_sal+ i.salary

      print(sum_sal/len(emps))

class Fulltime_Employee(Employee):

      def __init__(self, name, family, salary, department):
         Employee.__init__(self, name, family, salary, department)


list = []
list.append(Employee('John', 'Doe', 30000, 'IT'))
list.append(Employee('Jane', 'Levy', 45000, 'Sales'))

list.append(Fulltime_Employee('Jack', 'Sparrow', 23000, 'Design'))
list.append(Fulltime_Employee('Jared', 'Leto', 50000, 'Gaming'))

list[0].avg_sal(list)
```

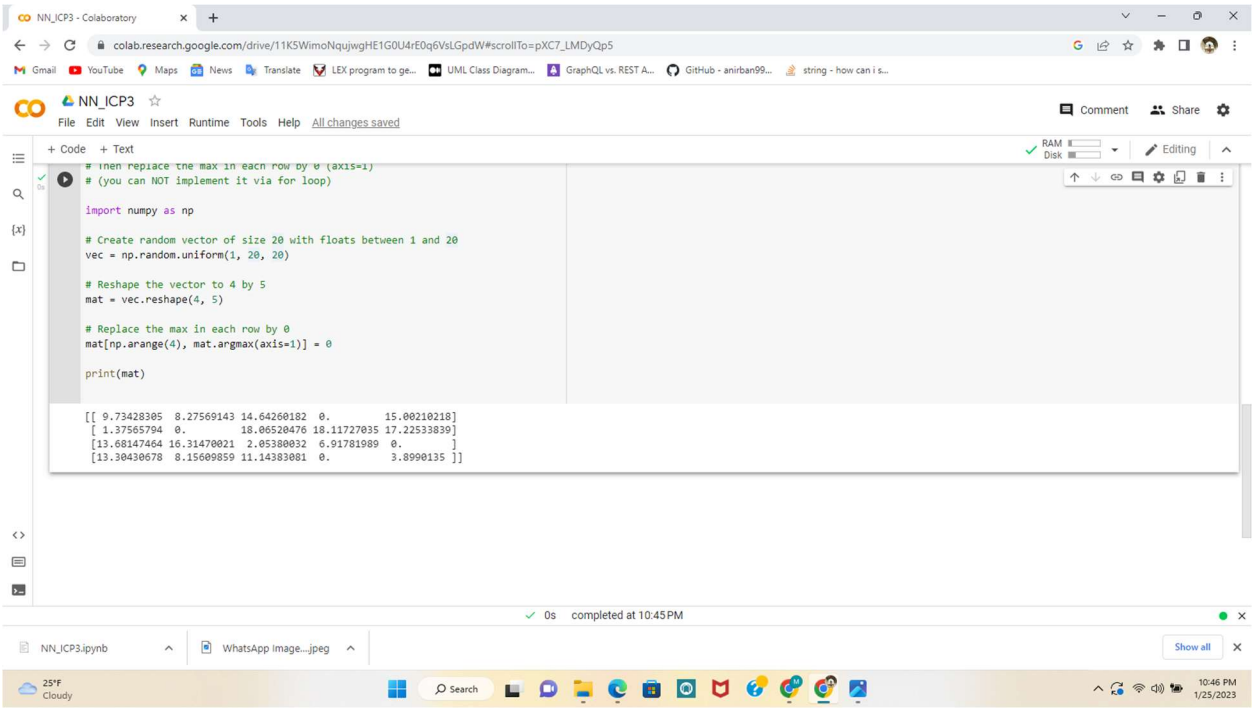
```
list[2].avg_sal(list)
print(Employee.emp_count)
```



Here in the first question, created a data member count, Created a constructor to initialize name, family, salary, department and created a function average salary and Created a Fulltime Employee class and it inherited the properties of Employee class and lastly created instances for the classes and called the respective member functions.

```
# 2. Numpy
# Using NumPy create random vector of size 20 having only float in the ran
ge 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)

import numpy as np

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

# Reshape the vector to 4 by 5
```

```
mat = vec.reshape(4, 5)

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

print(mat)
```



Here in the second question, Using NumPy created a random vector of size 20 having only float in the range 1-20. Then reshaped the array to 4 by 5 Then replaced the max in each row by 0 (axis=1).