Distributed Systems

Assignment 01

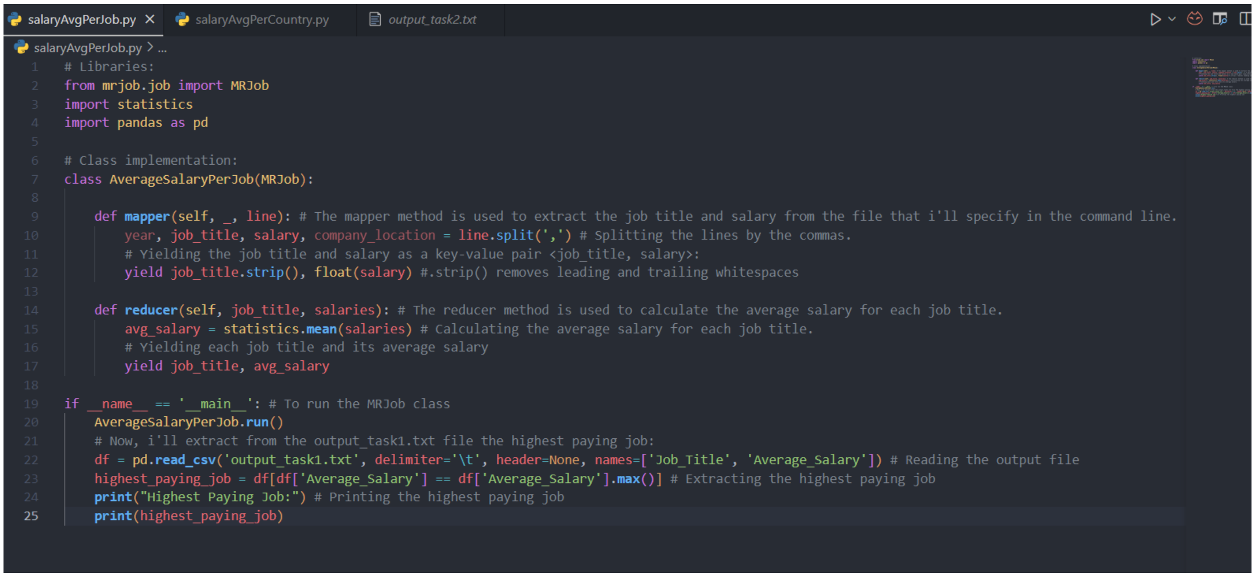
TASK #1:

I solved the 1st task by using 2 methods in the class AverageSalaryPerJob, which are: mapper and reducer.

a. **Mapper**: I used the mapper to map/ extract the job titles and salaries line by line from the ds_salaries text file (which I specified in the CLI). Then, I yielded the job titles and the salaries as key-value pairs.

b. **Reducer**: I used the reducer to reduce/ calculate the average salary for each job by its title. I used (statistics.mean(salaries)) to calculate the average salary for each job. Lastly, I yielded each job and its average salary.

- **The code file**:

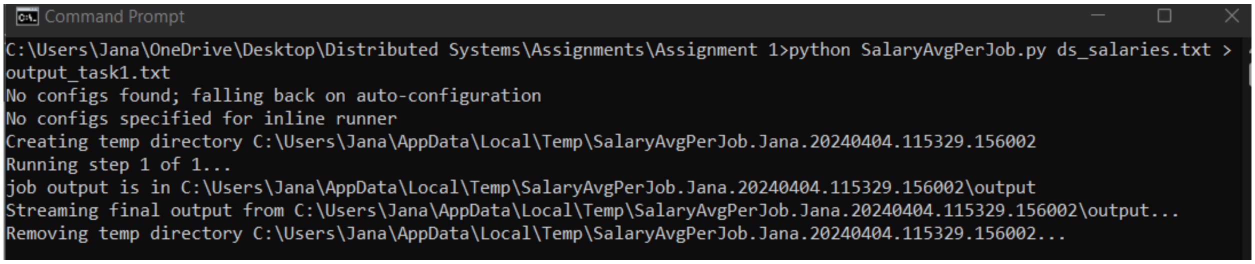


```python
# Libraries:
from mrjob.job import MRJob
import statistics
import pandas as pd

# Class implementation:
class AverageSalaryPerJob(MRJob):

    def mapper(self, _, line): # The mapper method is used to extract the job title and salary from the file that i'll specify in the command line.
        year, job_title, salary, company_location = line.split(',') # Splitting the lines by the commas.
        # Yielding the job title and salary as a key-value pair <job_title, salary>:
        yield job_title.strip(), float(salary) #.strip() removes leading and trailing whitespaces

    def reducer(self, job_title, salaries): # The reducer method is used to calculate the average salary for each job title.
        avg_salary = statistics.mean(salaries) # Calculating the average salary for each job title.
        # Yielding each job title and its average salary
        yield job_title, avg_salary

if __name__ == '__main__': # To run the MRJob class
    AverageSalaryPerJob.run()
    # Now, i'll extract from the output_task1.txt file the highest paying job:
    df = pd.read_csv('output_task1.txt', delimiter='\t', header=None, names=['Job_Title', 'Average_Salary']) # Reading the output file
    highest_paying_job = df[df['Average_Salary'] == df['Average_Salary'].max()] # Extracting the highest paying job
    print("Highest Paying Job:") # Printing the highest paying job
    print(highest_paying_job)
```

And then I Just ran the file by using the Command Line Prompt.

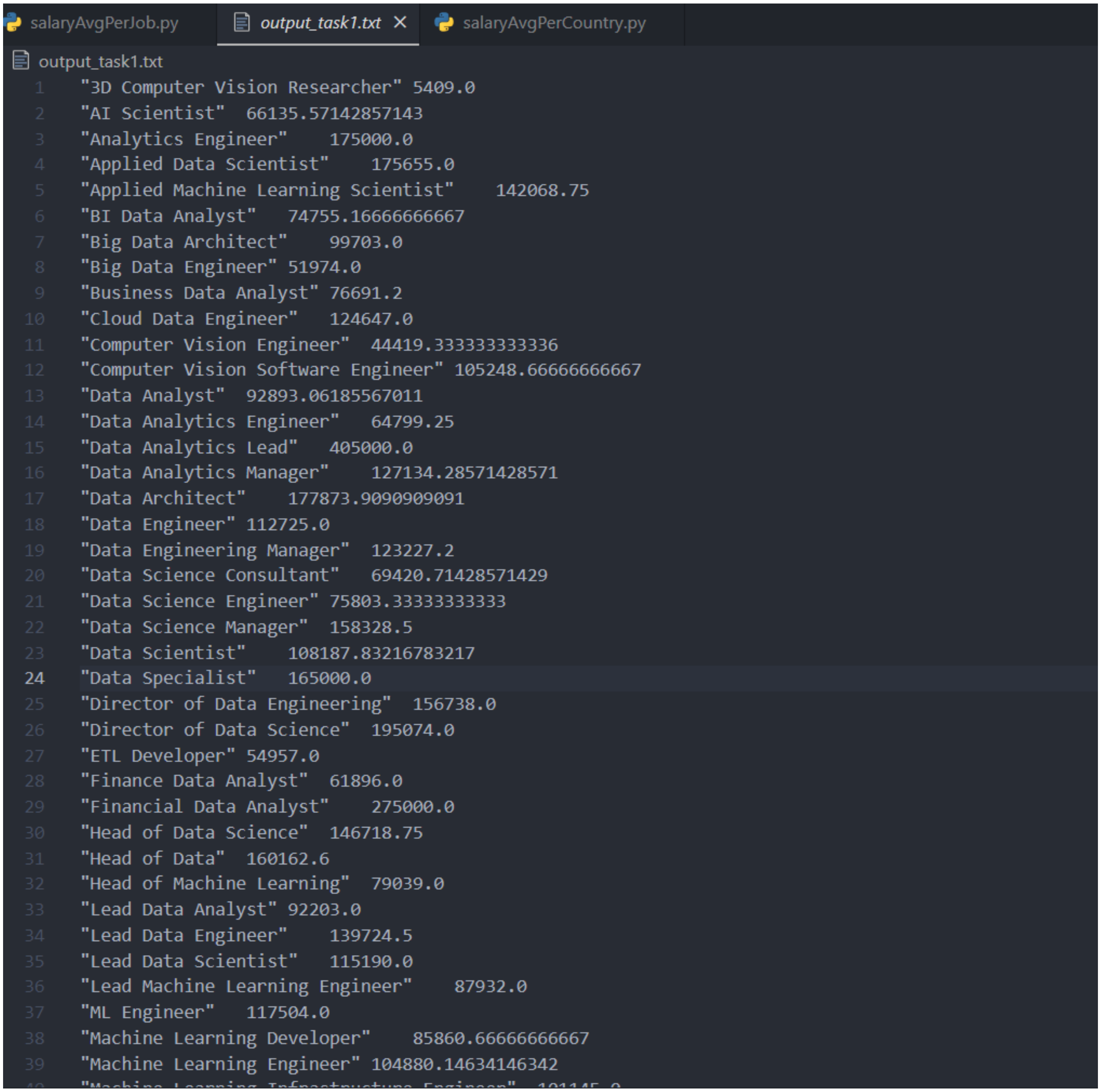
I used the command below to output the solution in an output file called: output_task1.txt

- **The command Line:**

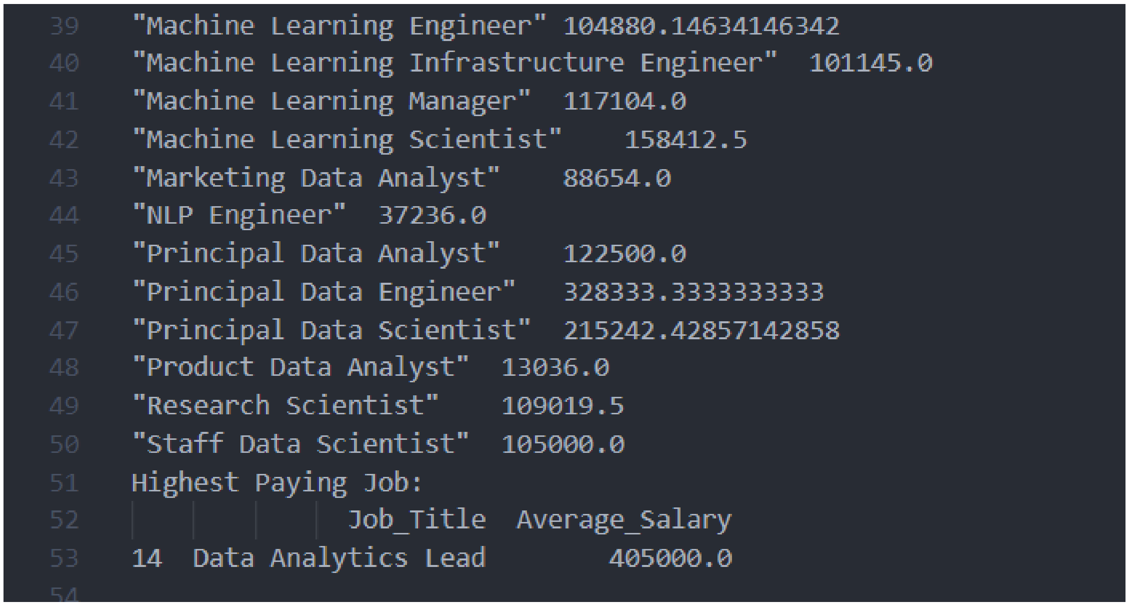


```
C:\Users\Jana\OneDrive\Desktop\Distributed Systems\Assignments\Assignment 1>python SalaryAvgPerJob.py ds_salaries.txt >
output_task1.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory C:\Users\Jana\AppData\Local\Temp\SalaryAvgPerJob.Jana.20240404.115329.156002
Running step 1 of 1...
job output is in C:\Users\Jana\AppData\Local\Temp\SalaryAvgPerJob.Jana.20240404.115329.156002\output
Streaming final output from C:\Users\Jana\AppData\Local\Temp\SalaryAvgPerJob.Jana.20240404.115329.156002\output...
Removing temp directory C:\Users\Jana\AppData\Local\Temp\SalaryAvgPerJob.Jana.20240404.115329.156002...
```

- **The output_task1.txt file:**



```
"3D Computer Vision Researcher" 5409.0
"AI Scientist"  66135.57142857143
"Analytics Engineer"    175000.0
"Applied Data Scientist"    175655.0
"Applied Machine Learning Scientist"    142068.75
"BI Data Analyst"   74755.16666666667
"Big Data Architect"    99703.0
"Big Data Engineer" 51974.0
"Business Data Analyst" 76691.2
"Cloud Data Engineer"   124647.0
"Computer Vision Engineer"  44419.333333333336
"Computer Vision Software Engineer" 105248.66666666667
"Data Analyst"  92893.06185567011
"Data Analytics Engineer"   64799.25
"Data Analytics Lead"   405000.0
"Data Analytics Manager"    127134.28571428571
"Data Architect"    177873.9090909091
"Data Engineer" 112725.0
"Data Engineering Manager"  123227.2
"Data Science Consultant"   69420.71428571429
"Data Science Engineer" 75803.33333333333
"Data Science Manager"  158328.5
"Data Scientist"    108187.83216783217
"Data Specialist"   165000.0
"Director of Data Engineering"  156738.0
"Director of Data Science"  195074.0
"ETL Developer" 54957.0
"Finance Data Analyst"  61896.0
"Financial Data Analyst"    275000.0
"Head of Data Science"  146718.75
"Head of Data"  160162.6
"Head of Machine Learning"  79039.0
"Lead Data Analyst" 92203.0
"Lead Data Engineer"    139724.5
"Lead Data Scientist"   115190.0
"Lead Machine Learning Engineer"    87932.0
"ML Engineer"   117504.0
"Machine Learning Developer"    85860.66666666667
"Machine Learning Engineer" 104880.14634146342
"Machine Learning Infrastructure Engineer"  101145.0
```



```
"Machine Learning Engineer" 104880.14634146342
"Machine Learning Infrastructure Engineer"  101145.0
"Machine Learning Manager"  117104.0
"Machine Learning Scientist"    158412.5
"Marketing Data Analyst"    88654.0
"NLP Engineer"  37236.0
"Principal Data Analyst"    122500.0
"Principal Data Engineer"   328333.3333333333
"Principal Data Scientist"  215242.42857142858
"Product Data Analyst"  13036.0
"Research Scientist"    109019.5
"Staff Data Scientist"  105000.0
Highest Paying Job:
              Job_Title  Average_Salary
14  Data Analytics Lead        405000.0
```

So, we deduce that the highest paying job is “Data Analytics Lead” with average salary 405000.0

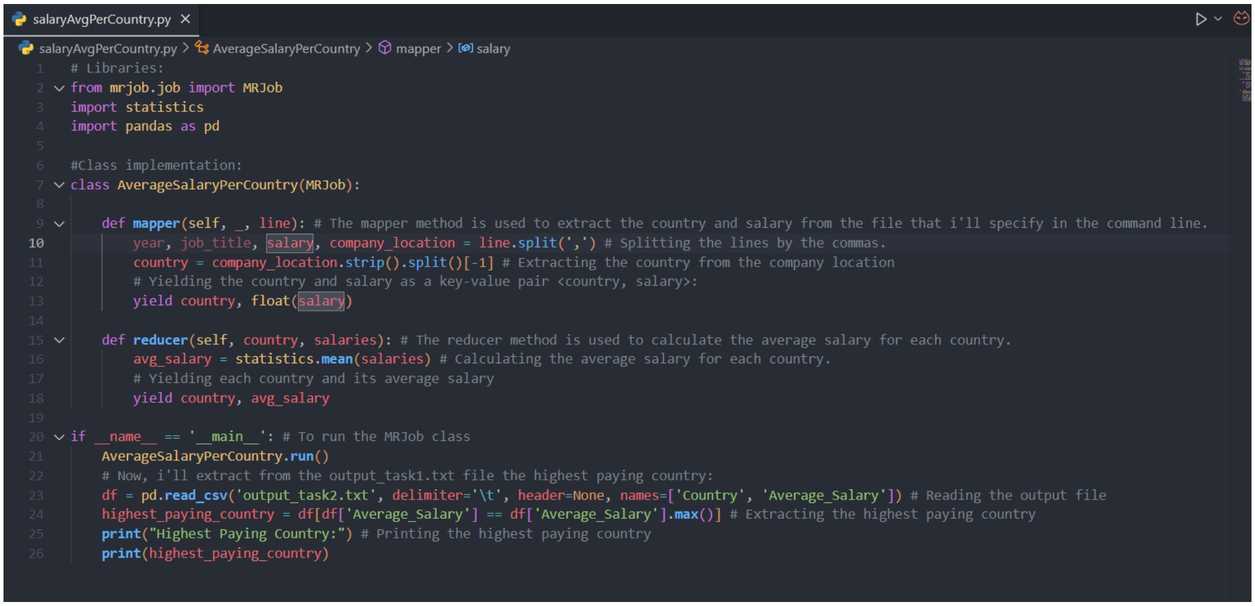
TASK #2:

I solved the 2nd task by using 2 methods in the class AverageSalaryPerCountry, which are: mapper and reducer.

a. **Mapper**: I used the mapper to map/ extract the countries and salaries line by line from the ds_salaries text file (which I specified in the CLI). I extracted the countries from the company_location column: company_location.strip().split()[-1]. Then, I yielded the countries and the salaries as key-value pairs.

b. **Reducer**: I used the reducer to reduce/ calculate the average salary for each country. I used (statistics.mean(salaries)) to calculate the average salary for each country. Lastly, I yielded each country and its average salary.

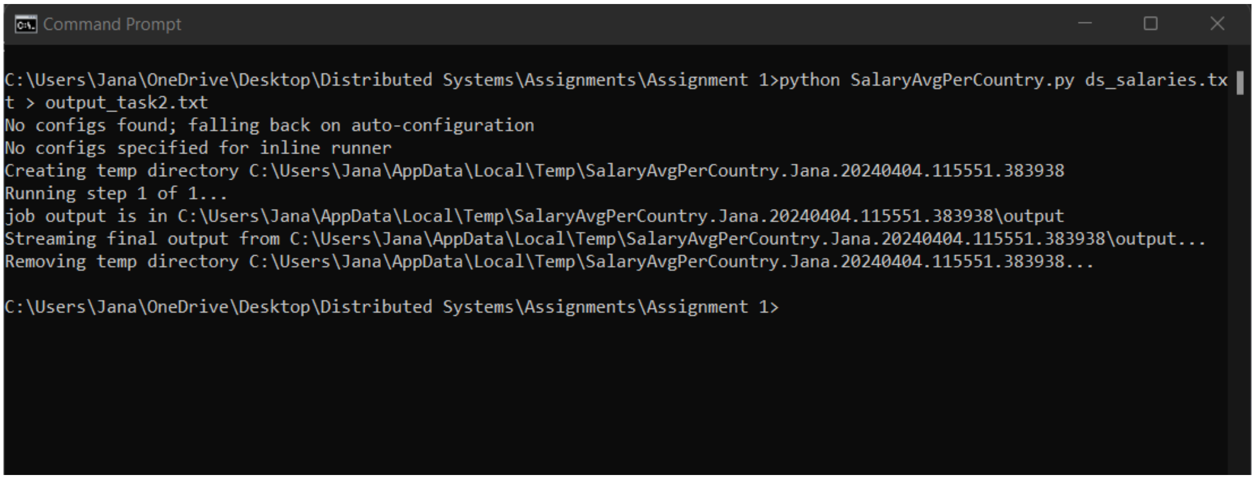
- **The code file:**



salaryAvgPerCountry.py
salaryAvgPerCountry.py > AverageSalaryPerCountry > mapper > salary

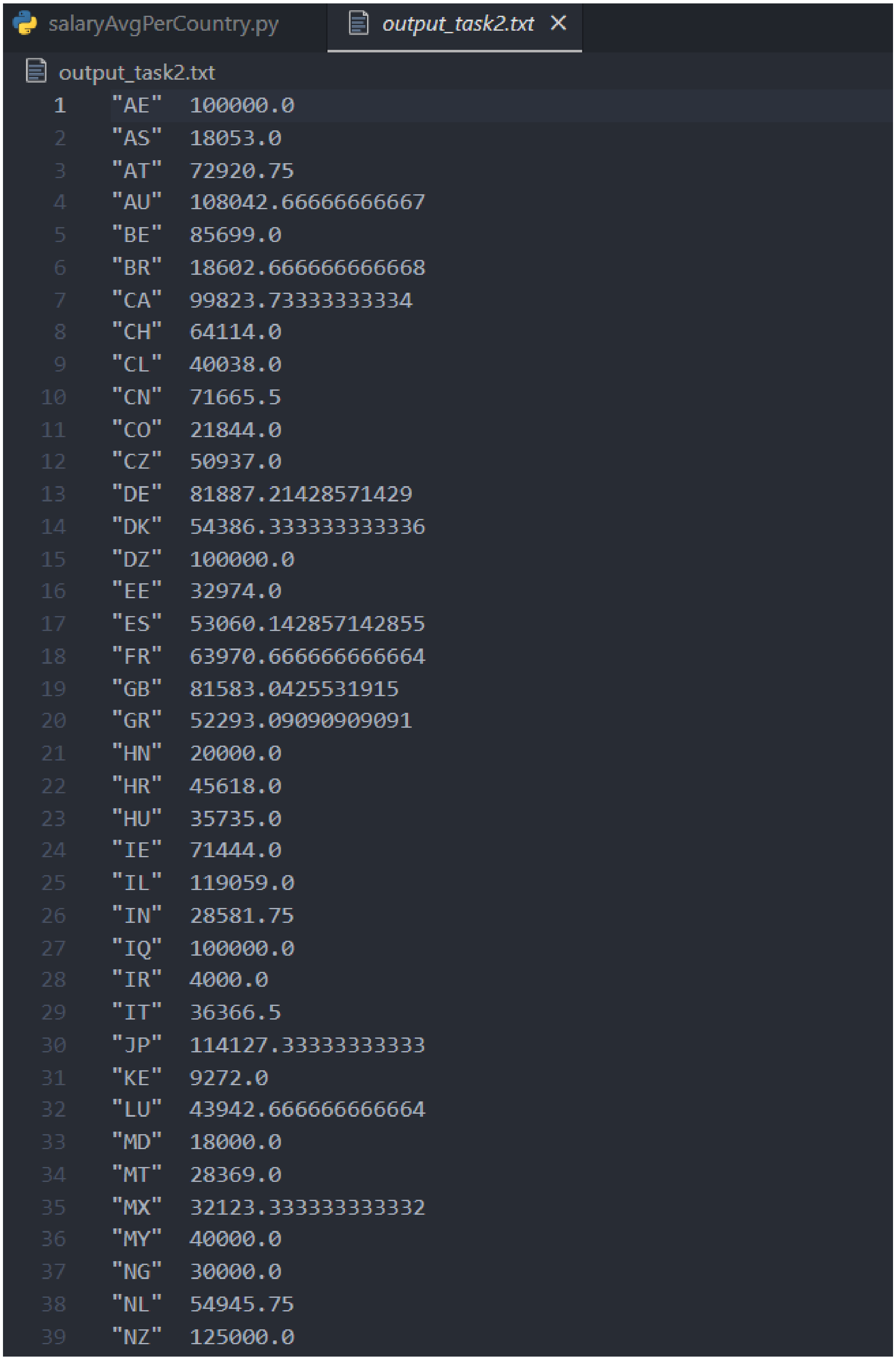
```python
# Libraries:
from mrjob.job import MRJob
import statistics
import pandas as pd

#Class implementation:
class AverageSalaryPerCountry(MRJob):

    def mapper(self, _, line): # The mapper method is used to extract the country and salary from the file that i'll specify in the command line.
        year, job_title, salary, company_location = line.split(',') # Splitting the lines by the commas.
        country = company_location.strip().split()[-1] # Extracting the country from the company location
        # Yielding the country and salary as a key-value pair <country, salary>:
        yield country, float(salary)

    def reducer(self, country, salaries): # The reducer method is used to calculate the average salary for each country.
        avg_salary = statistics.mean(salaries) # Calculating the average salary for each country.
        # Yielding each country and its average salary
        yield country, avg_salary

if __name__ == '__main__': # To run the MRJob class
    AverageSalaryPerCountry.run()
    # Now, i'll extract from the output_task1.txt file the highest paying country:
    df = pd.read_csv('output_task2.txt', delimiter='\t', header=None, names=['Country', 'Average_Salary']) # Reading the output file
    highest_paying_country = df[df['Average_Salary'] == df['Average_Salary'].max()] # Extracting the highest paying country
    print("Highest Paying Country:") # Printing the highest paying country
    print(highest_paying_country)
```

- **The CLI:**

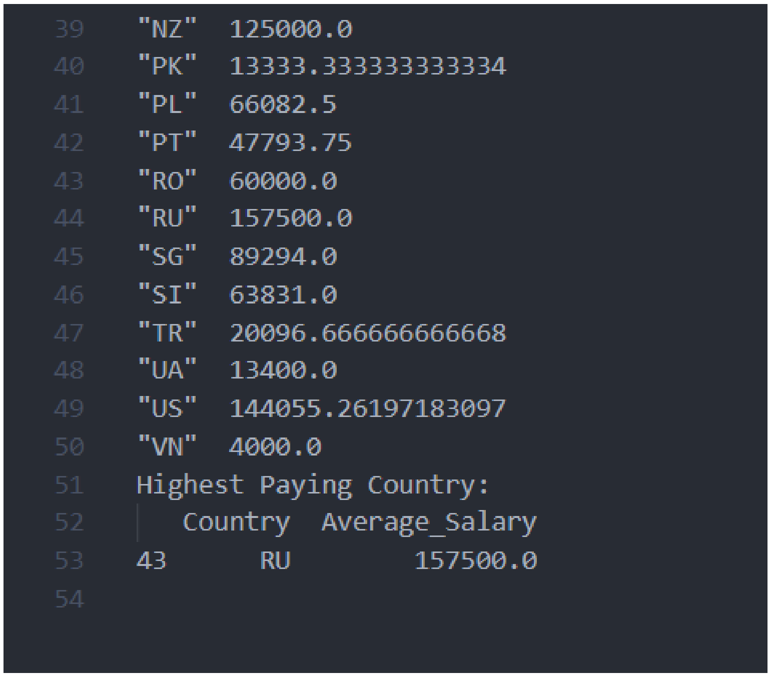


```
C:\Users\Jana\OneDrive\Desktop\Distributed Systems\Assignments\Assignment 1>python SalaryAvgPerCountry.py ds_salaries.tx
t > output_task2.txt
No configs found; falling back on auto-configuration
No configs specified for inline runner
Creating temp directory C:\Users\Jana\AppData\Local\Temp\SalaryAvgPerCountry.Jana.20240404.115551.383938
Running step 1 of 1...
job output is in C:\Users\Jana\AppData\Local\Temp\SalaryAvgPerCountry.Jana.20240404.115551.383938\output
Streaming final output from C:\Users\Jana\AppData\Local\Temp\SalaryAvgPerCountry.Jana.20240404.115551.383938\output...
Removing temp directory C:\Users\Jana\AppData\Local\Temp\SalaryAvgPerCountry.Jana.20240404.115551.383938...

C:\Users\Jana\OneDrive\Desktop\Distributed Systems\Assignments\Assignment 1>
```

- **The output_task2.txt file:**



```
"AE"   100000.0
"AS"   18053.0
"AT"   72920.75
"AU"   108042.66666666667
"BE"   85699.0
"BR"   18602.666666666668
"CA"   99823.73333333334
"CH"   64114.0
"CL"   40038.0
"CN"   71665.5
"CO"   21844.0
"CZ"   50937.0
"DE"   81887.21428571429
"DK"   54386.333333333336
"DZ"   100000.0
"EE"   32974.0
"ES"   53060.142857142855
"FR"   63970.666666666664
"GB"   81583.0425531915
"GR"   52293.09090909091
"HN"   20000.0
"HR"   45618.0
"HU"   35735.0
"IE"   71444.0
"IL"   119059.0
"IN"   28581.75
"IQ"   100000.0
"IR"   4000.0
"IT"   36366.5
"JP"   114127.33333333333
"KE"   9272.0
"LU"   43942.666666666664
"MD"   18000.0
"MT"   28369.0
"MX"   32123.333333333332
"MY"   40000.0
"NG"   30000.0
"NL"   54945.75
"NZ"   125000.0
```



```
"NZ"  125000.0
"PK"  13333.333333333334
"PL"  66082.5
"PT"  47793.75
"RO"  60000.0
"RU"  157500.0
"SG"  89294.0
"SI"  63831.0
"TR"  20096.666666666668
"UA"  13400.0
"US"  144055.26197183097
"VN"  4000.0
Highest Paying Country:
   Country  Average_Salary
43      RU        157500.0
```

So, we deduce that the highest paying country is “RU” with average salary 157500.0