SCALE FOR PROJECT PISCINE PYTHON DATA SCIENCE (/PROJECTS/PISCINE-PYTHON-DATA-SCIENCE) / RUSH 00 (/PROJECTS/PISCINE-PYTHON-DATA-SCIENCE-RUSH-00)

You should evaluate 2 students in this team

★

Git repository

```
git@vogsphere.kzn.21-school.ru:vogsphere/intra-uuid-38bfe4e5-0650-4e
```

Introduction

The methodology of School 21 makes sense only if
peer-to-peer assessments are done seriously.
This document will help you to do it properly.

- Please, stay courteous, polite, respectful and constructive in all
communications during this assessment. The bond of trust
between community 21 and you depends on it.

- Highlight possible malfunctions of the work done by the person
and take the time to discuss and debate it.

- Keep in mind that sometimes there can be differences
in interpretation of the tasks and the scope of features.
Please, stay open-minded to the vision of the other.

Guidelines

- Evaluate only the files that are on the GIT
repository of the student or group.

- Doublecheck that the GIT repository is the one
corresponding to the student
or the group as long as to the project.

- Meticulously check that nothing malicious has been used
to mislead you and have you assess something except
the content of the official repository.

- If you have not finished the project yet, it is compulsory
to read the entire instruction before starting the review.

- Use the special flags in the scale to report an empty
or non-functional solution as long as a case of cheating.
In these cases, the assessment is completed
and the final grade is 0 (or in a case of cheating is -42).
However, except for a case of cheating, you are encouraged to continue
reviewing the project to identify the problems that caused the situation
in order to avoid them for the next assessment.

- You must stop giving points from the first wrong exercise

even if the following exercises are correct.

Attachments

subject.pdf (https://cdn.intra.42.fr/pdf/pdf/49874/en.subject.pdf)

ml-latest-small.zip (/uploads/document/document/8795/ml-latest-small.zip)

movies.py (/uploads/document/document/8796/movies.py)

links.py (/uploads/document/document/8797/links.py) tags.py (/uploads/document/document/8798/tags.py)

ratings.py (/uploads/document/document/8799/ratings.py)

Preliminaries

Respect the rules

- The repository contains the work of the student (or group).
- The student is able to explain their work
at any time during the assessment.
- The general rules and any rules specific to the day are
respected throughout the assessment.
- There are indeed two files: the first with
the module and the second with the report.

Yes No

Mandatory part

Any hardcoded result is worth zero for the exercise.

Class Tests

- The tests check the data types of the return of the methods?
- The tests check the data types of the list elements where it is the case?
- The tests check if the data sorted correctly where it is the case?
- The tests cover all the methods from the classes?
- The code passes all the tests?
In all other cases, the test is failed.

Yes No

Class Ratings

- Data from ratings.csv and movies.csv is joined inside the constructor
and stored there in a field, and the field is used in the methods?
- The method top_by_ratings() works with average (default) and median values?
- The class Users is inherited from the class Movies?
- Check if the methods that require the usage of movie titles are really
using them instead of MovieIds?
In all other cases, the test is failed.

Yes No

Class Tags

- Data from tags.csv is stored in the constructor in a field, and the field

is used in the methods?
In all other cases, the test is failed.

Yes No

Class Movies

- Data from movies.csv is stored in the constructor in a field, and the field is used in the methods?
In all other cases, the test is failed.

Yes No

Class Links

- Data from links.csv and movies.csv is joined inside the constructor and stored there in a field, and the field is used in the methods?
- Unsuccessful requests to the webpages in the method get_imdb() are handled with the exceptions?
- Check if the methods that require the usage of movie titles are really using them instead of MovieIds?
In all other cases, the test is failed.

Yes No

Report

Here and after you work in the .ipynb file.
- Try to instantiate all 4 classes with wrong files (that have a different structure) and try to call any method from them. Are the exceptions are handled?
- All the methods from the module are used in the report?
- The report besides the code itself has the text that describes and explains its logic?
- The only used import is import movielens_analysis?
- No built-in functions are used? Everything is done by the methods from the module?
- Each cell with the code has %timeit?
In all other cases, the test is failed.

Yes No

Bonus part

Bonus

- Additional methods are added to the module?
- The new methods are tested?
- The tests check the correctness of method calculations?
- The report tells a story (it is written in an interactive style with questions and answers, it creates some intrigue, it sparks some emotions, it is interesting and easy to read)?

Rate it from 0 (failed) through 5 (excellent)

Ratings

Don't forget to check the flag corresponding to the defense

✔ Ok ★ Outstanding project

Empty work | No author file | Invalid compilation | Norme | Cheat | Crash | Leaks

⊘ Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation