# SCALE FOR PROJECT DSLR (/PROJECTS/42CURSUS-DSLR)

You should evaluate 1 student in this team

★

Git repository

```
git@vogsphere.42kl.edu.my:vogsphere/intra-uuid-f82dce45-b4c2-46b4-b8
```

## Introduction

For the smooth running of this evaluation, please respect the following rules:

- Remain polite, kind, respectful and constructive whatever happens during this conversation. It's a matter of confidence between you and the 42 community.

- Highlight the potential problems you 've had with the work you're presented to the person or the group you're grading, and take the time to talk about and discuss those issues.

- Accept the fact that the exam subject or required functions might lead to different interpretations. Listen to your discussion partner's perspective with an open mind (are they right or wrong ?) and grade them as fairly as possible.
42's teaching methods can make sense only if peer-evaluation is taken seriously.

## Guidelines

- You must only evaluate what you will find in the student's or group's GiT repository.

- Take the time to check that the GiT repository matches the student or group and the project.

- Double check that no malicious alias was used to mislead you and make you grade something different from the official repository content.

- If a script supposed to help evaluate the exam is supplied by either side, the other side will have to strictly check it to avoid nasty surprises.

- If the evaluating student has not yet taken this project, they will have to read the exam subject in its entirety before starting the evaluation.

- Use the flags available on this grading system to signal an empty or non-funcional project, a norm flaw, cheating, etc. In that case, evaluation stops and final grade is 0 (or -42 if it's a cheating problem). However, if it's not a cheating problem, you are invited to keep talking about the work that has been done (or not done, as a matter of fact) in order to identify the issues that lead to this stalemate and avoid it next time.

## Attachments

- evaluate.py (https://cdn.intra.42.fr/document/document/15461/evaluate.py)
- dataset_truth.csv (https://cdn.intra.42.fr/document/document/15462/dataset_truth.csv)
- subject.pdf (https://cdn.intra.42.fr/pdf/pdf/80971/en.subject.pdf)
- datasets.tgz (https://cdn.intra.42.fr/document/document/15463/datasets.tgz)

# Data analysis

*In this part, we will study the succinct data analysis through the 'describe' function.*

**The describe function**

Execute the 'describe' function with 'dataset_train.csv' in parameter. Does the output respect the requirement of the subject? That is: count, mean, std, min, 25%, 50%, 75% and max.

Yes No

**Hands in code**

Open the 'describe' source and talk about the code together. Make sure the assessed student doesn't use any third party library that would replace one of the requested results. For instance: no 'mean' function prompting the student would not have coded himself.

If the assessed student is using a prohibited function, check the Cheat flag and end the evaluation. Validate only if they coded everything themselves.

Yes No

**Notions explanations**

Ask the assessed student to explain the following notions:

- What is the average (mean)?
- What is the standard deviation (std)?
- What is a quartile (25% - 50% - 75%)?

1 correct answer = 1 point, 2 correct answers = 3 points, 3 correct answers = 5 points.

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

5

# Data visualization

*Here, we're going to tackle data visualization. This section will require a little thinking more than just development skills. You will be the one to judge if the assessed student answers the question and if his explanations are satisfying. If you're not satisfied with an answer, it might be wise to sit and think of another solution together. There might be more than one anwser to a given question.*

**Histogram**

Launch the `histogram` script.

Does the displayed graphic help you answer the question:
Which Hogwarts class has an homogenous grade repartition between the four houses?

Ask the assessed student to explain what you see and why they believe it answers the question.

Yes No

**Scatter plot**

Launch the `scatter_plot` script.

Does the displayed graphic help you answer the question:
which two features are similar?

Ask the assessed student to explain what you see and why they believe it answers the question. For this part, there should only be one obvious answer.

Yes No

**Pair plot**

Launch the `pair_plot` script.

Does the graphic help you answer the question:
from this graph, which characteristics will you use to train your coming logistic regressions?

Ask the assessed student to explain what you see and why they believe it answers the question.

☑ Yes ☒ No

# Logistic regression

*We are going to evaluate the multi-classifier.*

**Discussions**

Before launching any program, ask the assessed student how the logistic regression works.

We're not here to nitpick but to make sure the assessed student has understood the following points: how logistic regression works compared to to linear regression, point in nornmalising the data, what's the one-vs-all method. Of course, you can go further than these elements, but don't try to push or trick the student.

Did the student give the correct explanations?

☑ Yes ☒ No

**Machine learning!**

Time to evaluate the algorithme. First, execute `logreg_train` with `dataset_train.csv` . This should create a file containing the weights for each model. Is this the case?

☑ Yes ☒ No

**Predictions**

Once you have trained your models, execute `logreg_predict` with the weights and `dataset_test.csv` as parameters. This should create a file named `houses.csv` .

In order to evaluate the multi-classifier performance, use the script `evaluate.py` which will compare the files `houses.csv` with `dataset_truth.csv` containing the truth (that is, the real houses the students belong to).

Mc Gonagall had asked for a minimum score of 98% (equals 0.98). If this is so, you can validate. Otherwise... Too bad.

☑ Yes ☒ No

# Bonus

*Reminder: if, somehow, the program doesn't react as it should (bus error, segfault etc...), evaluation ends and the grade is 0. Use the respective flags. This instruction works during the whole evaluation. Bonus will be taken into account only if the mandatory part is PERFECT. PERFECT meaning it is completed, that its behavior cannot be faulted, even because of the slightest mistake, improper use, etc... Practically, it means that if the mandatory part is not validated, none of the bonus will be taken in consideration.*

**Let's talk, now.**

Feel free to grade any additionnal features in the project. It will remain at your discretion as long as you have good reasons to do so.

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

3

# Ratings

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

✔ Ok ★ Outstanding project