SCALE FOR PROJECT ML MODULE 00 (/PROJECTS/ML-MODULE-00)

You should evaluate 1 student in this team

★

Git repository

git@vogsphere.42paris.fr:vogsphere/intra-uuid-6b6b5086-abfc-421f-b66

Comments

This sixth module is the evolution of two original projects created by the
Paris-based student association 42 AI.
They were named Bootcamp Python and Bootcamp Machine Learning.
active members of 42 AI re-designed both of them for the school curriculum.

Bootcamps has been developped between August 20119 and March/April 2020.
Active 42AI members organized severals sessions of 2 weeks to 42Paris students
to offer them the possibility to get famliar with Python and basics concepts
of machine learning.

The success of those sections brings the pedagogy to accept the idea to integrate
the 2 bootcamps to the curriculum (initial discussion (01-05/2019) with 42 Paris
pedago team highlighted a categorical opposition/refusal to this idea)

The transcription had been realized over the direction of Matthieu David
several 42AI members contributed to the redaction on the correction scales.
For futur corrections on the scale, please contact the 42AI association via
contact@42ai.fr or the current 42AI pedagogical supervisor.

Introduction

The Bootcamp Python and Bootcamp Machine Learning were originally created by
[42AI](https://github.com/42-AI) active members and were adapted to 'piscine'
format for the school 42 curriculum.
For any issue or suggestion: [42Paris](https://github.com/42-AI/bootcamp_python/issues) and
[42AI](https://github.com/42-AI/bootcamp_machine-learning/issues).

As usual, you have to observe the following courtesy rules:

- Remain polite, courteous, respectful, and constructive throughout the
evaluation process. The well-being of the community depends on it.

- Identify with the evaluated person or group the eventual dysfunctions of
the assignment. Take the time to discuss and debate the problems you may
have identified.

- You must consider that there might be some differences in the
understanding of and approach to project instructions, and the scope of
its functionalities, between you and your peers. Always remain open-minded
and grade them as fairly as possible. The pedagogy is valid only and only
if peer-evaluation is conducted seriously.

The goal of this module is to get started with the Python language.

You will study map, reduce, filter, args and kwargs ...

Disclaimer

The serie of modules started to be produce at the time of the release of Python 3.7. Students are free to use later version of Python as long as they verified the producted code complies with all the aspects precised in the subjects.
As a consequence we recommend to students to perform the modules with the the Python version 3.7 (but this is just an advice).
Version can be checked with the command ```python -V```.

Guidelines

General rules
- Only grade the work that is in the student or group's GiT repository.

- Double-check that the GiT repository does belong to the student. Ensure that the work is the one expected for the corrected exercise and don't forget to verify that the command "git clone" is run in an empty folder.

- Check carefully that no malicious aliases were used to make you evaluate files that are not from the official repository.

- To avoid any surprises, carefully check that both the evaluating and the evaluated students have reviewed the possible scripts used to facilitate the grading.

- If the evaluating student has not completed that particular project yet, it is mandatory for them to read the entire subject prior to starting the defense.

- Use the flags available on this scale to signal an empty repository, non-functioning program, a Norm error (specified next in general rules), cheating, and so forth.
In these cases, the grading is over and the final grade is 0, or -42 in case of cheating. However, except the exception of cheating, you are encouraged to continue to discuss your work even if the later is in progress in order to identify any issues that may have caused the project failure and avoid repeating the same mistake in the future.

- Use the appropriate flag.

- Remember that for the duration of the defense, no other unexpected, premature, or uncontrolled termination of the program, else the final grade is 0.

- You should never have to edit any file except the configuration file if the latter exists. If you want to edit a file, take the time to explain why with the evaluated student and make sure both of you agree on this.

- The Norm: The PEP 8 standards is not mandatory, but recommended.

- The function eval is never allowed.

- Your exercises are going to be evaluated by other students, make sure that your variable names and function names are appropriate and civil.

Attachments

subject.pdf (https://cdn.intra.42.fr/pdf/pdf/54998/en.subject.pdf)

Exercise 00: The Matrix

The goal of the exercise is to manipulate basic matrix operations.

Basic tests

- First, verify all the expected magic methods are implemented, data and values attributes are available within the class and when printing a matrix you observe an output similar to:
  '(Matrix [[n1, ..., n2], ..., [n5, ..., n6]])'
  or for vectors:
  '(Vector [[n1], ..., [n2]] or Vector [[n1, ..., n2]])'
- Run the 'test.py' to check the correctness of the class.

Yes No

Exercise 01: TinyStatistician

The goal of the exercise is to tackle with very basic statistic notions.

Basic tests

Run the following script:

```
import TinyStatistician as ts

data = [42, 7, 69, 18, 352, 3, 650, 754, 438, 2659]
epsilon = 1e-5
err = "Error, grade 0 :("

tstat = ts.TinyStatistician()
assert abs(tstat.mean(data) - 499.2) < epsilon, err
assert abs(tstat.median(data) - 210.5) < epsilon, err

quartile = tstat.quartile(data)
assert abs(quartile[0] - 18) < epsilon, err
assert abs(quartile[1] - 650) < epsilon, err

assert abs(tstat.percentile(data, 10) - 3) < epsilon, err
assert abs(tstat.percentile(data, 28) - 18) < epsilon, err
assert abs(tstat.percentile(data, 83) - 754) < epsilon, err

assert abs(tstat.var(data) - 654661) < epsilon, err
assert abs(tstat.std(data) - 809.11) < epsilon, err
```

Yes No

Exercise 02: Simple prediction

The goal of the exercise is to understand and manipulate the notion of hypothesis in machine learning.

Basic tests

Perform at least the following tests:

- Call simple_predict with theta=[0, 0] and any values for x. The function must always return an array full of 0.
- Call simple_predict with theta=[1, 0] and any values for x. The function must always return an array full of 1.
- Call simple_predict with theta=[0, 1] and any values for x. The function must always return the same array as the x parameter.
- Call simple_predict with theta=[1, 1] and any values for x. The function must always return the same array as the x parameter,

with 1 added to each value.

Yes No

Exercise 03: Add Intercept

The goal of the exercise is to Understand and manipulate the notion of hypothesis in machine learning

Basic tests

- Call add_intercept with a vector or matrix containing any values. The function must return a matrix containing the same values, with a column full of 1 added to the left.

Yes No

Exercise 04 - Prediction

The goal of the exercise is to understand and manipulate the notion of hypothesis in machine learning.

Basic tests

Perform at least the following tests:

- Call predict_ with theta=[0, 0] and any values for x. The function must always return an array full of 0.
- Call predict_ with theta=[1, 0] and any values for x. The function must always return an array full of 1.
- Call predict_ with theta=[0, 1] and any values for x. The function must always return the same array as the x parameter.
- Call predict_ with theta=[1, 1] and any values for x. The function must always return the same array as the x parameter, with 1 added to each value.

Yes No

Exercise 05 - Let's Make Nice Plots

Plotting

Perform at least the following tests:

- Pass the following arguments to the function: plot(np.array([0, 1]), np.array([0, 1]), np.array([0, 1])). The function must plot a line and two points. The points must be located at (0, 0) and (1, 1), and must be lying on the line.
- Pass the following arguments to the function: plot(np.array([0, 1]), np.array([0, 1]), np.array([1, 1])). The function must plot a line and two points. The points must be located at the same positions, but the line must be above them.
- Pass the following arguments to the function: plot(np.array([0, 2]), np.array([0, 0]), np.array([-1, 1])). The function must plot a line and two points. The points must be located at (0, 0) and (2, 0). The line must be separating the points.

Yes No

Exercise 06 - Loss Function

The goal of the exercise is to understand and manipulate the notion of loss function in machine learning.

Basic tests

Perform at least the following tests:

- Define a vector with any values in it. Pass the same vector for both arguments of both functions. loss_elem_ should return a vector full of zeros and loss_ should return 0.
- Define two vectors y_hat=np.array([[1], [2], [3], [4]]) and y=np.array([[0], [0], [0], [0]]) and pass them to the functions. loss_elem_ should return [[0.125], [0.5], [1.125], [2]] and loss_ should return 3.75.

- Try with other values and check that it corresponds to the formula.

Yes No

Exercise 07 - Vectorized Loss Function

The goal of the exercise is to Understand and manipulate the notion of loss function in machine learning.

Basic tests

Perform at least the following tests:

- Define a vector with any values in it. Pass the same vector for both arguments of both functions. The function should return 0.
- Define two vectors y_hat=np.array([[1],[2],[3],[4]]) and y=np.array([[0],[0],[0],[0]]) and pass them to the function. The function should return 3.75.
- Try with other values and check that it corresponds to the formula.

Yes No

Exercise 08 - Lets Make Nice Plots Again

The goal of the exercise is to Understand and manipulate the notion of loss function in machine learning.

Plotting

Perform the same tests as exercise 06. But this time, the function must plot additional lines between each points and the predicted values.

Yes No

Exercise 09 - Other Loss Functions

The goal of the exercise is to Understand and manipulate the notion of loss function in machine learning.

MSE

- Define a vector with any values in it. Pass the same vector for both arguments of both functions. The function should return 0.
- Define two vectors y_hat=[[1],[2],[3],[4]] and y=[[0],[0],[0],[0]] and pass them to the functions. The function should return 7.5.
- Try with differents vectors values. You can compare with the scikit-learn method.

Yes No

RMSE

- Perform the same tests as for MSE. The results of rmse_ function should be 2.738...
- Try with differents vectors values. You can compare with the scikit-learn method.

Yes No

MAE

- Define a vector with any values in it. Pass the same vector for both arguments of both functions. The function should return 0.
- Define two vectors a=[[1],[2],[3],[4]] and b=[[0],[0],[0],[0]]. The function mae_ should return 2.5.
- Try with differents vectors values. You can compare with the scikit-learn method.

Yes No

R2score

- Define two vectors a=[[1],[2],[3],[4]] and b=[[0],[0],[0],[0]]. r2score_ should return -5.
- Try with differents vectors values. You can compare with the scikit-learn method.

☑ Yes ☒ No

Peers discussion

If no deep (or moderate) discussions had taken place during the correction this is (a shame obviously) an opportunity to discuss between peers on the different concepts of the module. No need to thank us, you are welcome. No points are distributed for the disccusion. (You must validate the questions)

Concatenation tricks

Why do we used the concatenation tricks already? Why does it represent?
(hint: y-intercept)?

☑ Yes ☒ No

Square it or not square it?

Why do we square the distance between the predicted values and the real values in the loss function? What could happens if we do not squared the distances?
Can we raised the distances to other powers? what happens comparatively to the power 2?

☑ Yes ☒ No

Meaning of the loss function

What does the loss function represent in the form we have seen it here?
Do you think we can used other loss function?

☑ Yes ☒ No

The best loss

Which is the best value we want in theory for the loss function?
What does it mean if J is zero?
Can the loss function treated here can be negative? What do you think of the parity of the loss function?

☑ Yes ☒ No

Commutativity

What is a commutative operator?
Does the matrix multiplication is a commutative operator?

If you have time, take two matrices: A with size n * m and B with size m * n:

- what is the size of C = A * B?
- what is the size of D = B * A? What happens if A and B are squared matrices? Can A * B be equal to B * A?

☑ Yes ☒ No

Ratings

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

✔ Ok

Empty work Cheat Crash Incomplete group Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation