DroNET Simulator

Course: *Autonomous Networking - Prof. Gaia Maselli (A.A. 2020-2021)*

Speaker: *Dr. Andrea Coletta - 18-12-2020*

SAPIENZA
UNIVERSITÀ DI ROMA

BAYESIAN OPTIMIZATION

Performance of Machine Learning algorithms are usually dependent on the choice of hyperparameters.

How to pick/search the optimal hyperparameter?

- Manual
- Grid search
- Random search
- …?
- Instead could we use a model to select which hyperparameters will be good next?

Slide by: Franco Lin, Tahmid Mehdi, Jason Li

BAYESIAN OPTIMIZATION

BAYESIAN OPTIMIZATION : is global optimization method for noisy black-box functions (i.e., machine learning) which can be used to hyperparameter optimization!

Bayesian Optimization uses all of the information from previous evaluations and performs some computation to determine the next point to try

If our model takes days to train, it would be beneficial to have a well structured way of selecting the next combination of hyperparameters to try

Bayesian Optimization is much better than a person finding a good combination of hyperparameters

BAYESIAN OPTIMIZATION

Intuition:

GOAL: We want to find the peak of our true function (eg. accuracy as a function of hyperparameters)

$x^* = \text{argmax}_x\ f(x)$ or in case of error $x^* = \text{argmin}_x\ f(x)$

To find this peak, we will fit a **Gaussian Process** to our **observed points** and **pick our next best point** where **we believe** the maximum will be.

This next point is determined by an acquisition function - that trades of exploration and exploitation

BAYESIAN OPTIMIZATION

Idea:

In contrast to random or grid search, we keep track of past evaluation results which they use to form a probabilistic model mapping hyperparameters to a probability of a score on the objective function:

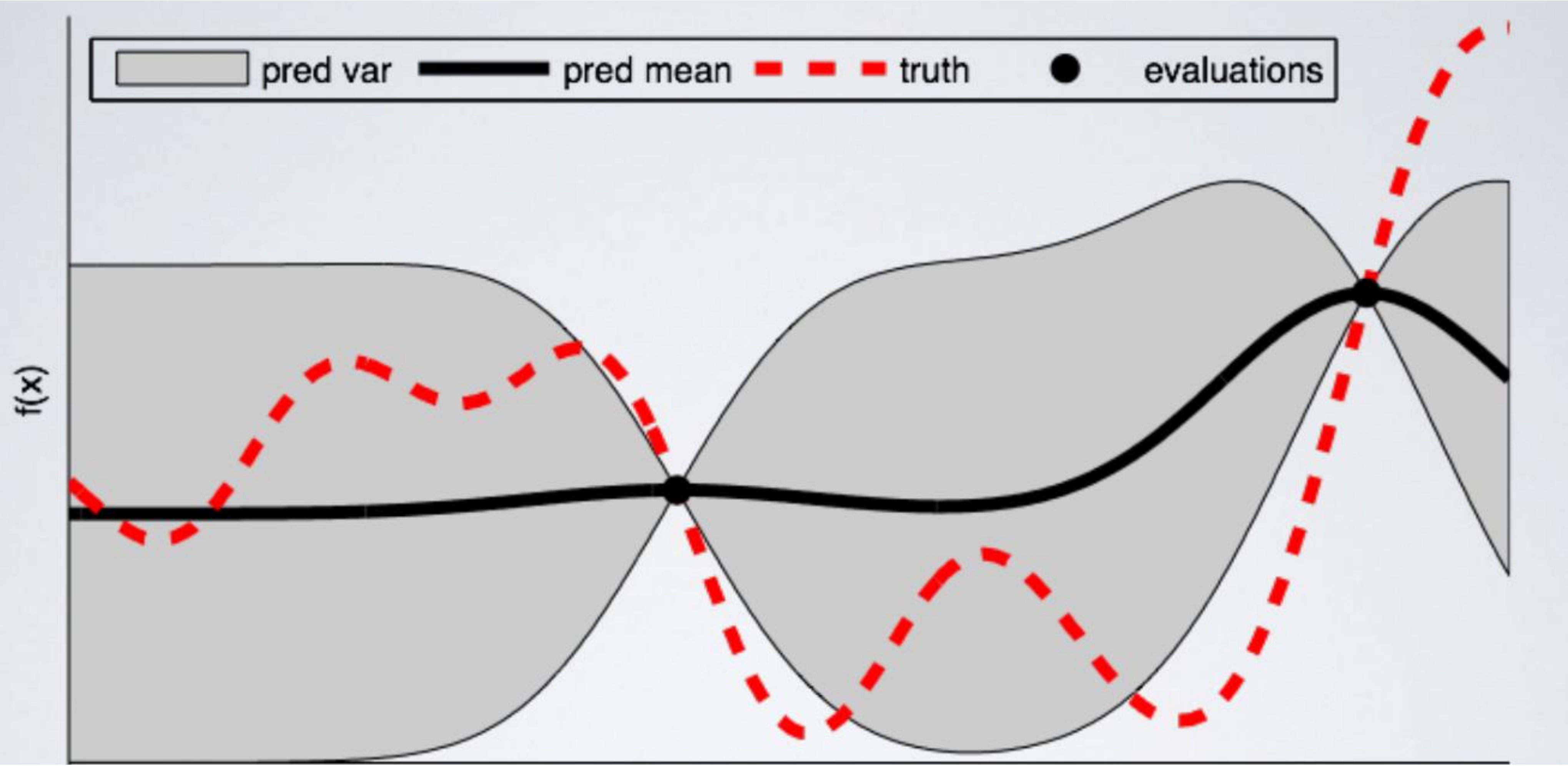
P(score | hyperparameters) (surrogate obj function)

BAYESIAN OPTIMIZATION

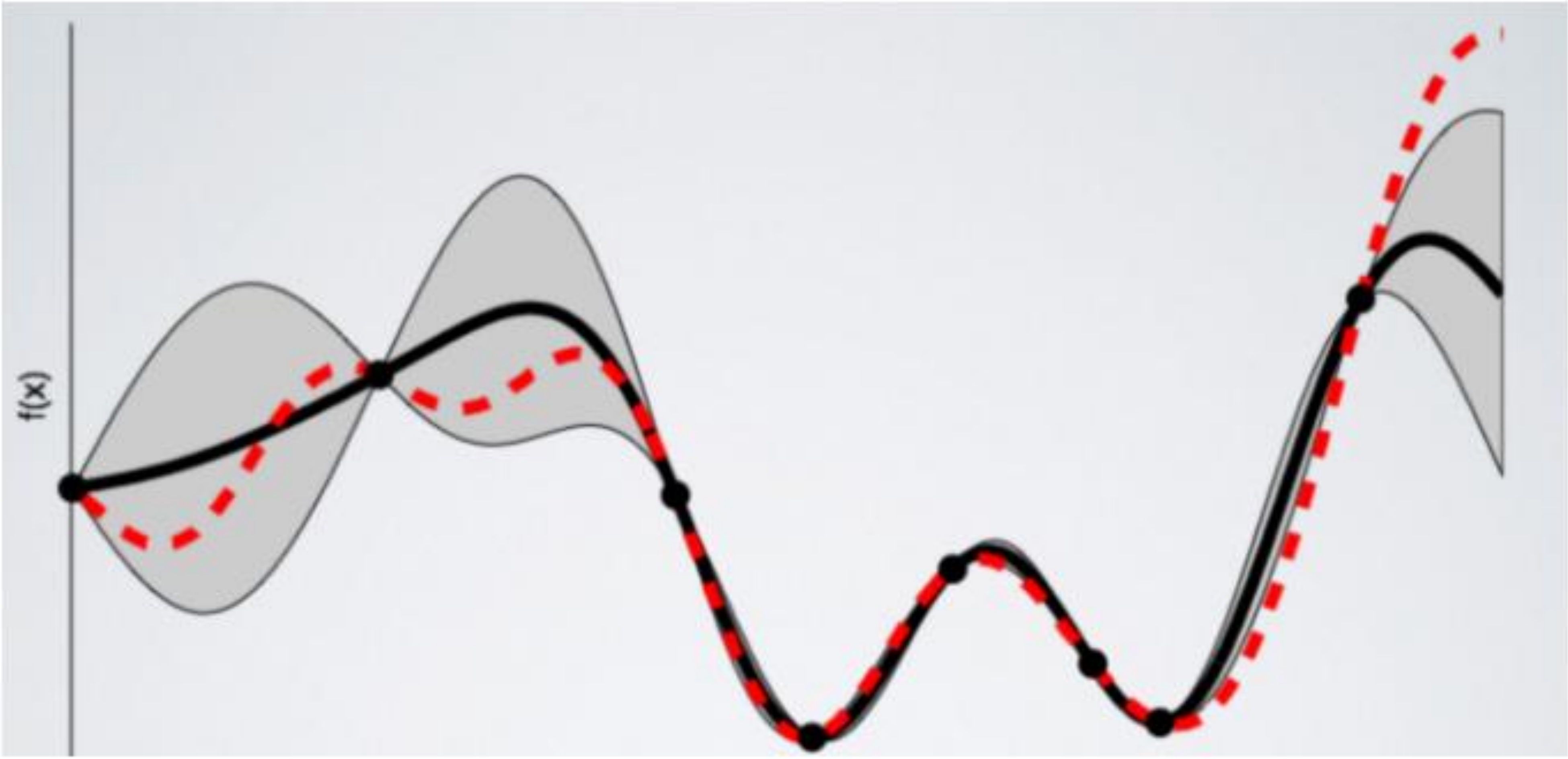
Workflow:

1) **Build a surrogate probability model of the objective function**
2) **Find the hyperparameters that perform best on the surrogate**
3) **Apply these hyperparameters to the true objective function**
4) **Update the surrogate model incorporating the new results**
5) **Repeat steps 2–4 until max iterations or time is reached**

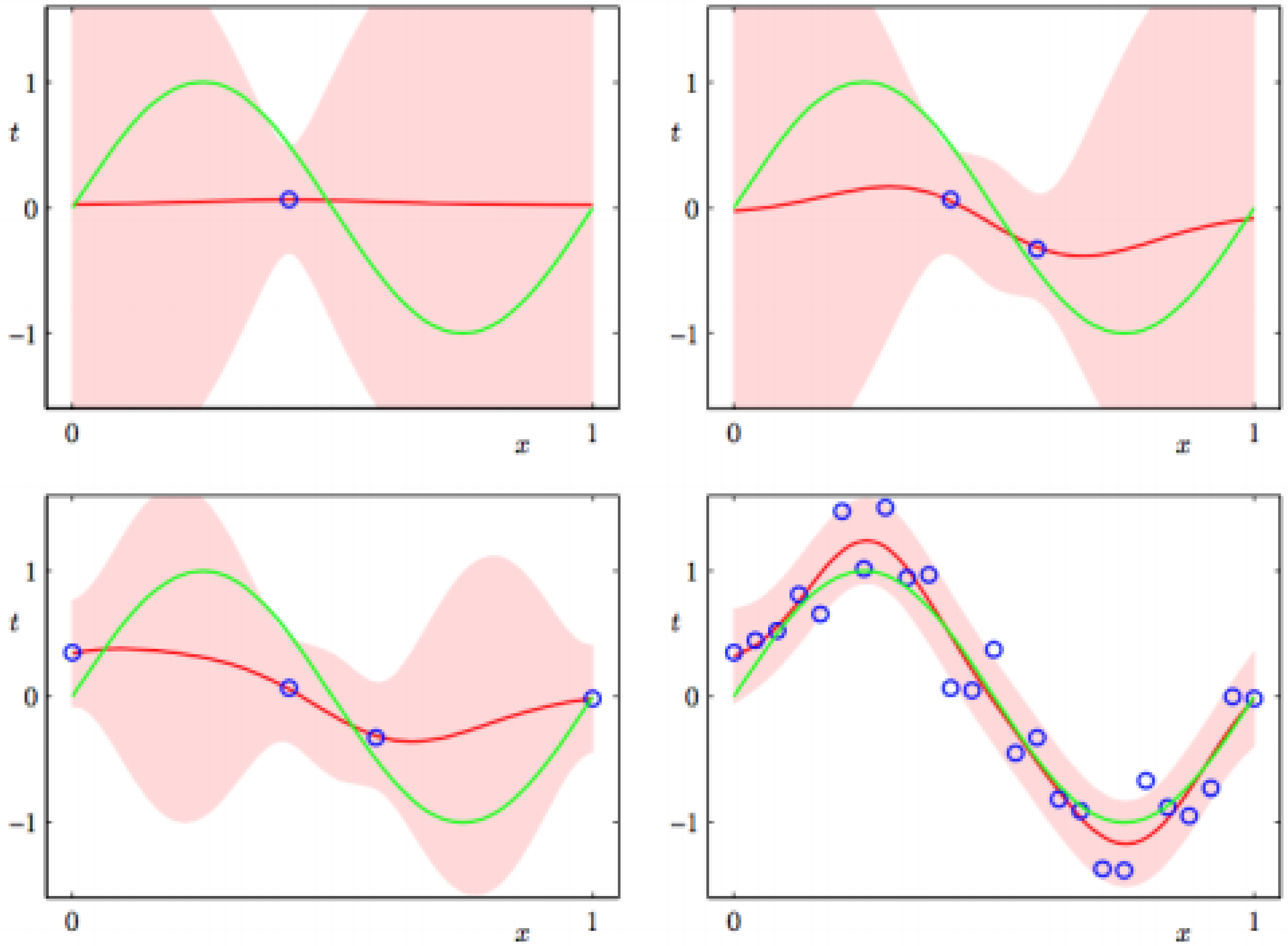
BAYESIAN OPTIMIZATION



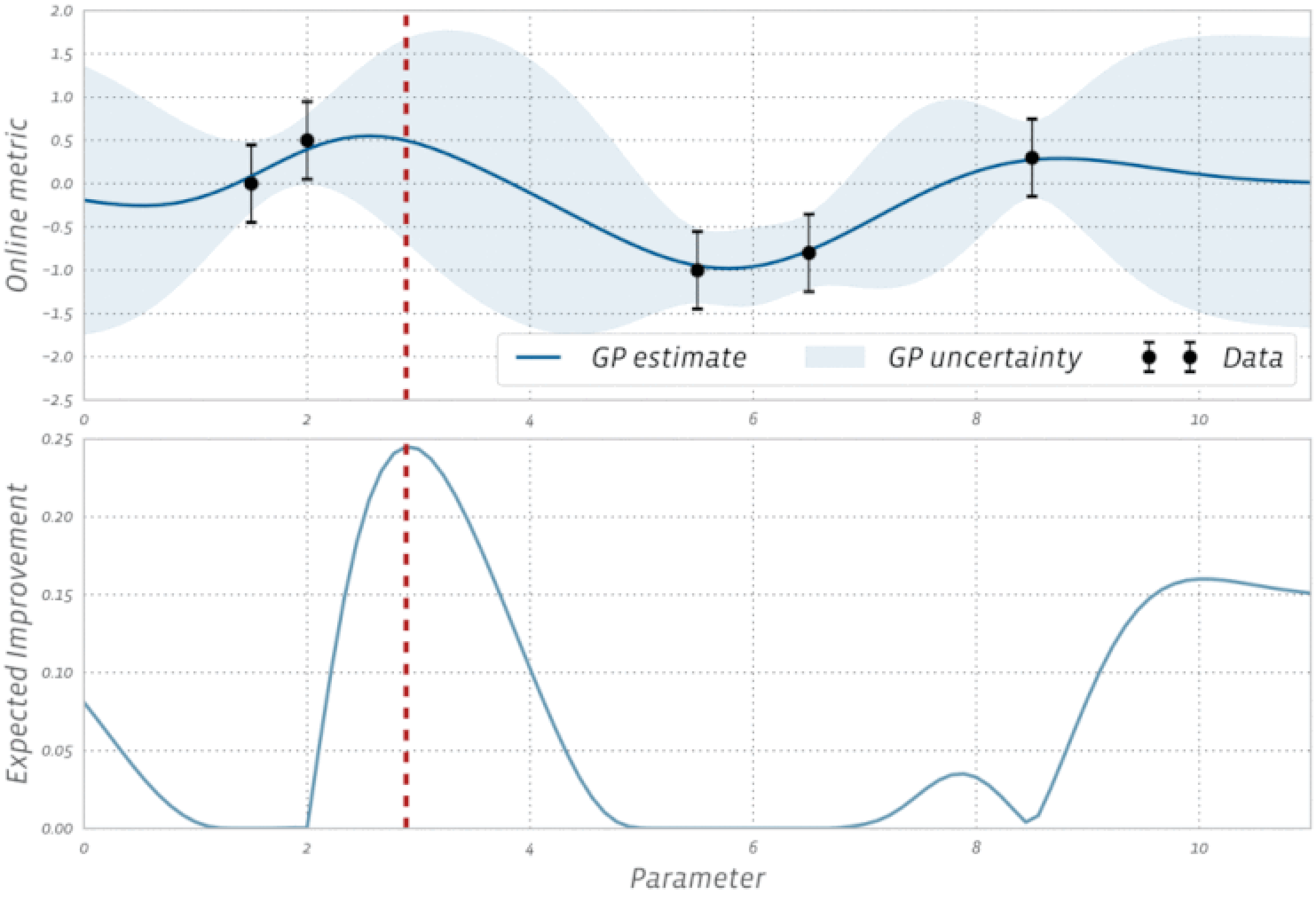
BAYESIAN OPTIMIZATION



BAYESIAN OPTIMIZATION – ANOTHER EXAMPLE



BAYESIAN OPTIMIZATION – ANOTHER EXAMPLE



GAUSSIAN PROCESS OPTIMIZATION IN PYTHON

Library : skopt*

1) Step - Declaration of hyperparameters to tune and their domain

hyperparameters = [(0, 1), (0, 1)] #by default are float!

2) Create Bayesian optimizer

```
optimizer = gp_minimize(ml_model, hyperparameters, n_calls=N_ITER,
                        n_random_starts=INIT_POINT, random_state=BAY_SEED)
```

* https://scikit-optimize.github.io/stable/auto_examples/bayesian-optimization.html

GAUSSIAN PROCESS OPTIMIZATION IN PYTHON

```
hyperparameters = [ (0, 1), (0, 1) ] #by default are float!

optimizer = gp_minimize(ml_model, hyperparameters, n_calls=N_ITER,
                        n_random_starts=INIT_POINT, random_state=BAY_SEED)
```

Note:

```
def ml_model(parameters):
    alpha, gamma = parameters
    model.train(alpha, gamma)
    accuracy or error = model.test()
    ....
    ....
    return accuracy/error (the actual value evaluated by the Bayesian)
```

CONTACTS

Andrea Coletta:

coletta@di.uniroma1.it