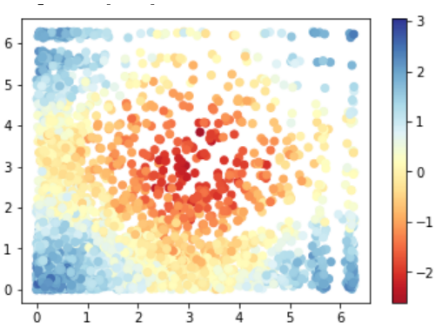
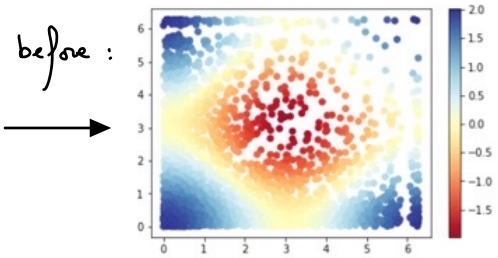
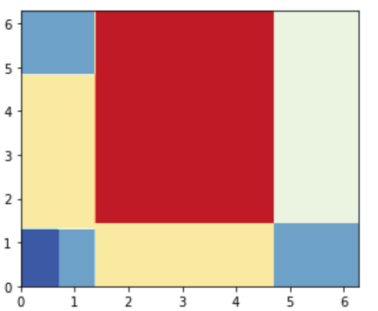
The new data distribution : the output (that I try to predict) is more noisy !



before :



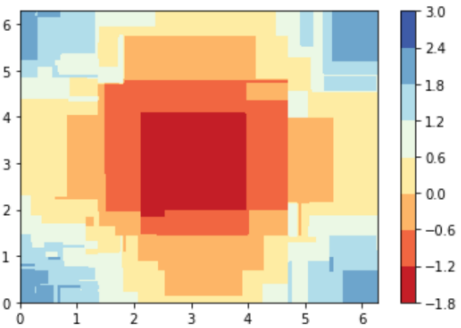
Decision trees :



max_depth = 3

↳ we fit poorly the train data

→ "underfitting"

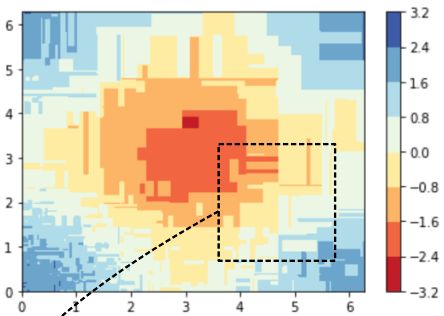


max_depth = 8

fit is on ! ✓

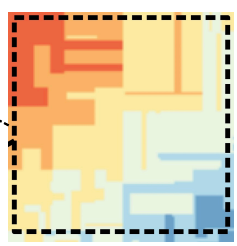
prediction does not seem "too complicated" ✓

→ on !



max_depth = 14

↳ the trained tree fits too much the learning data, and varies too much :



→ overfitting

We can check on the cross-validation score (test.score) of a tree over multiple train-test splits :

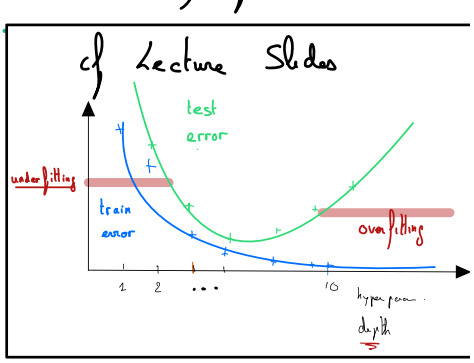
```
depth 1 score 0.3059732077354048
depth 2 score 0.5798514521408162
depth 3 score 0.7397198103247626
depth 4 score 0.8207684060499227
depth 5 score 0.8553084624998055
depth 6 score 0.875657408869172
depth 7 score 0.8845319514551624
depth 8 score 0.8825718728674463
depth 9 score 0.8790144221038204
depth 10 score 0.8699275541790745
depth 11 score 0.858754696223517
depth 12 score 0.8473440408526696
depth 13 score 0.8388846695486368
depth 14 score 0.8322986395600955
```

underfitting

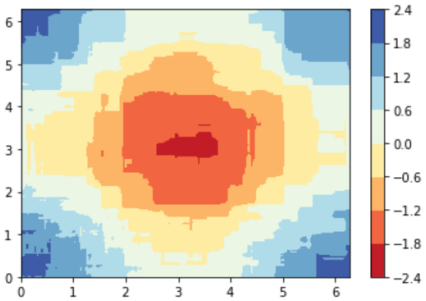
→ optimal depth for a single tree

overfitting

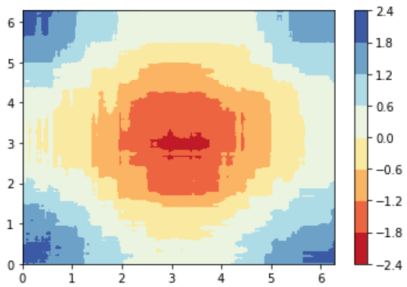
cf Lecture Slides



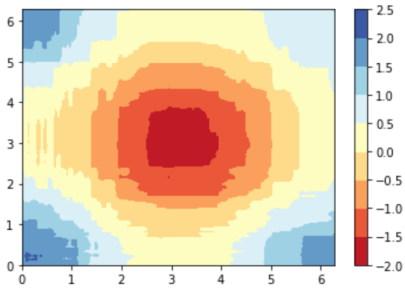
Still, the prediction does not look great - let us use random forests!



with 200 trees



with some parameters
selected w. RandomizedSearchCV
(tested 20 values of the 8000 possible
set of hyperparameters



with 2000 trees,
min_samples_leaf = 6.