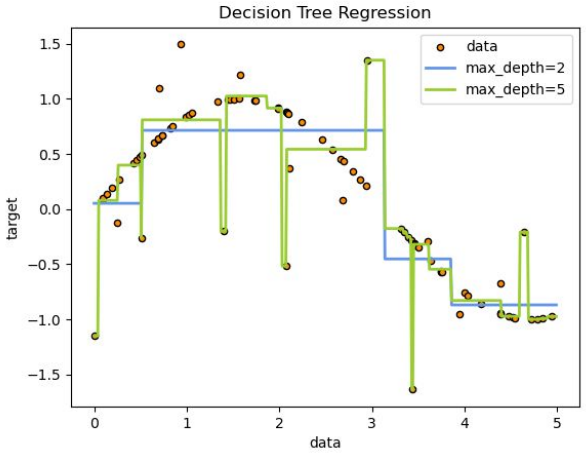
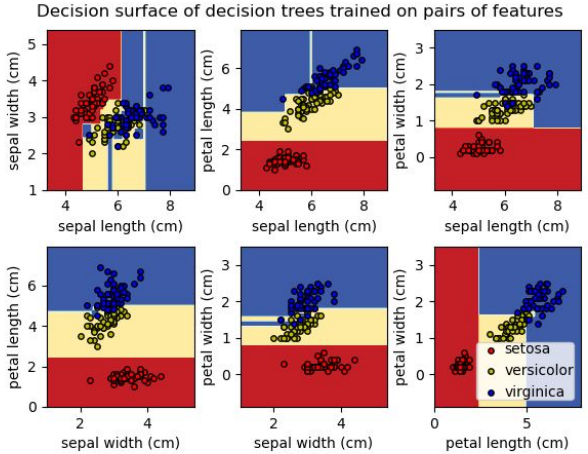
Beyond decision trees

Problem with decision trees

Decision trees can easily overfit the data

A large tree will get perfect classification on the training set





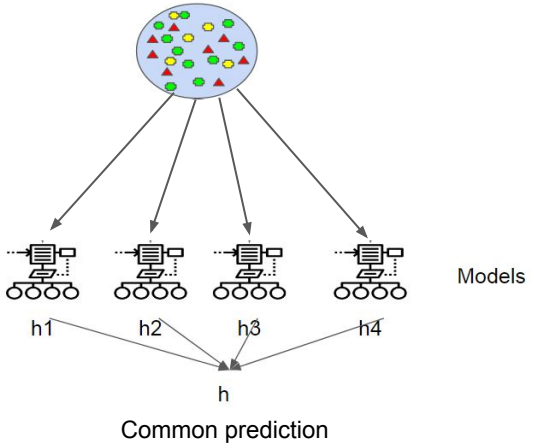
Pictures from https://scikit-learn.org/stable/modules/tree.html

Ensemble

One solution to overcome overfitting

Training several models and aggregating their predictions

Condition:
To be effective, the learners must be as complementary as possible.
If they are all the same, there is no point in doing ensemble.
Each learner should be specialist in a subdomain of the problem



- Best choice: simple learners, slightly better than random guess, “weak learners”.

- Good candidates for “weak learners”: Small decision trees

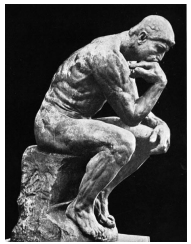
Reference: https://smlbook.org/book/sml-book-draft-latest.pdf, Chap. 7

Wisdom of the crowd?

- Weaker learned are not following a crowd
- They should not be correlated, they should not have the same “point of view”

- Ensemble is more like a board or committee with people of different backgrounds rather than a crowd!

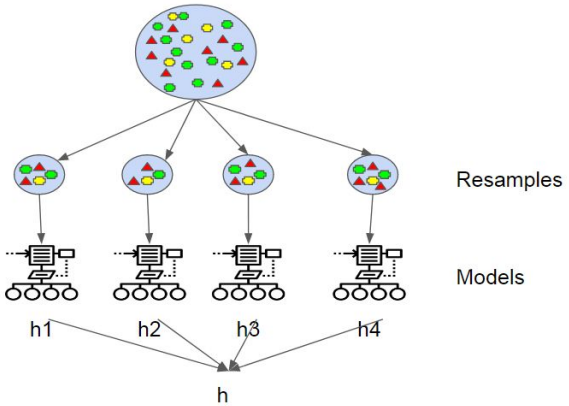
Philosophical note: diversity is important for making good choices



Bagging (Bootstrap aggregating)

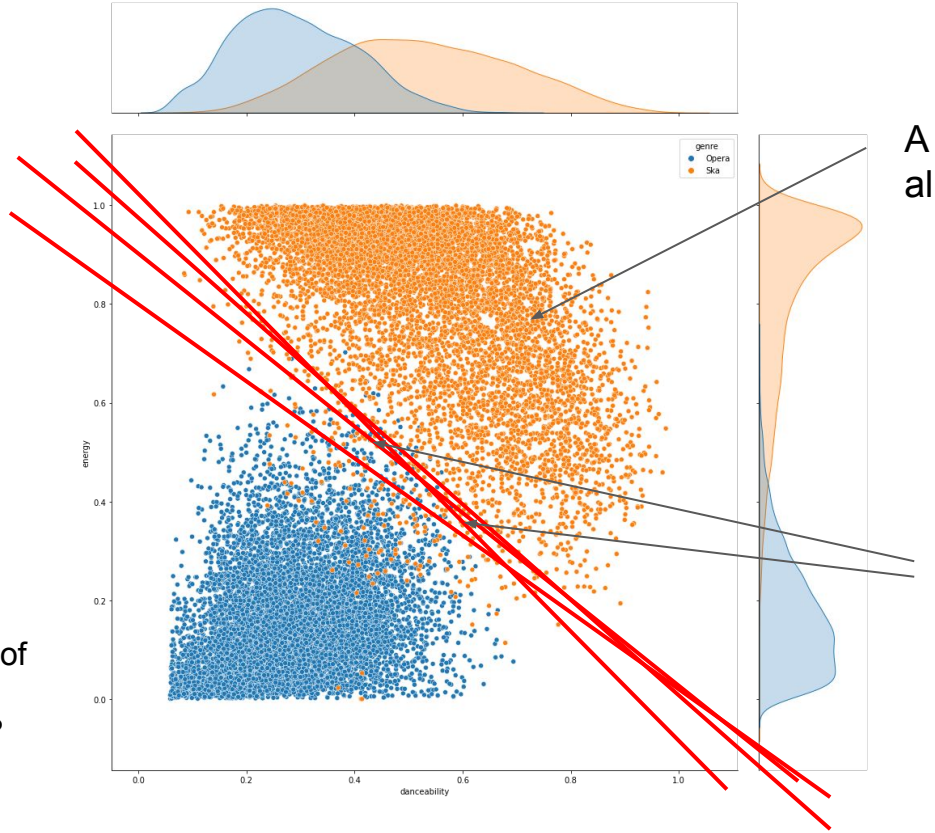
- Sample randomly with replacement the training set to get x subsets
- Train x ML methods, one per subset
- Aggregate the x methods
  - use the average of the x methods to predict the target value of a new sample (regression)
  - use majority voting of the x methods to predict the class of a new sample (classification)

The x weak learners are trained on different datasets -> they should make different predictions for at least some of the samples.



Weak learners

Learners with different decision boundaries



A region where they all agree

we can use majority voting to decide in these regions

Take this example with caution: logistic regression is not really a weak learner. Here the diversity of decisions is not very high.
Could we increase the diversity?

Random forest

An ensemble of small trees

each tree is trained on:

- a subset of the training set (bagging)
- a subset of the features

A subset of features bring diversity and avoid to pick always the same most important features.

Boosting

Add weak learners iteratively while optimizing a loss function.

- The new learner is trained to predict the samples classified wrong at the previous step.
- Several ways to do that Adaboost, gradient boosting, XGboost
- Example of gradient boosting: adding a learner is a gradient step

F_m Model at step m $\quad F_m(x_i)$ Prediction for x_i at step m

Weak learner

$$F_{m+1}(x_i) = F_m(x_i) + h_m(x_i) = y_i \quad \text{or} \quad h_m(x_i) = y_i - F_m(x_i)$$

Residual, Error in the prediction at step m

Example of Loss function:

$$L_{\text{MSE}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - F(x_i))^2$$

Gradient:

$$-\frac{\partial L_{\text{MSE}}}{\partial F(x_i)} = \frac{2}{n}(y_i - F(x_i)) = \frac{2}{n}h_m(x_i)$$

Learning step: $F_{m+1} = F_m - \alpha \nabla L(F_m)$

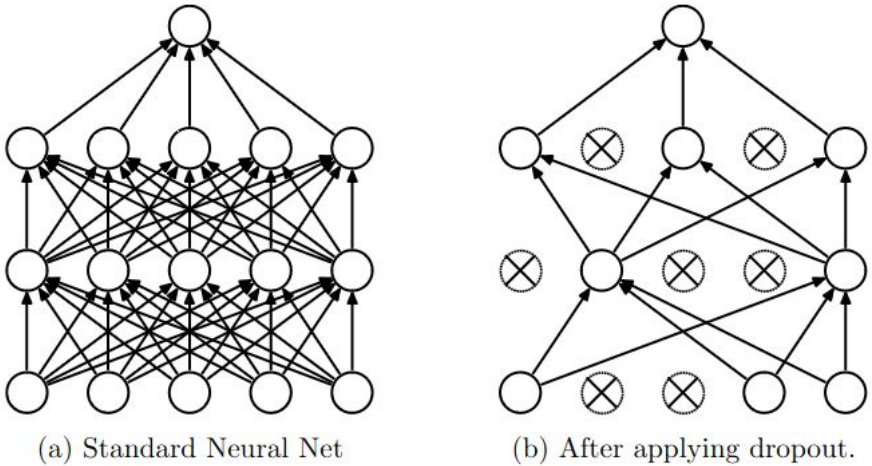
Boosting

Boosting in practice, with decision trees:
https://colab.research.google.com/github/lewtun/hepml/blob/master/notebooks/lesson04_intro-to-gradient-boosting.ipynb

Dropout

Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov; JMLR 2014



(a) Standard Neural Net

(b) After applying dropout.

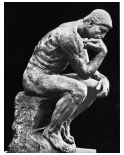
Some randomly chosen neurons are ignored during training.

The ignored neurons change over training: Each reduced network is a weak learner

At inference time, the full network is considered, seen as an ensemble of weak learners

Very efficient to reduce overfitting. Very popular in deep learning.

I recommend reading the motivation part of the paper : hypothesis for the superiority of sexual reproduction



The lottery ticket hypothesis

Frankle, Jonathan, and Michael Carbin, "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks." *International Conference on Learning Representations*. 2018.

Based on these results, we articulate the *lottery ticket hypothesis*: dense, randomly-initialized, feed-forward networks contain subnetworks (*winning tickets*) that—when trained in isolation—reach test accuracy comparable to the original network in a similar number of iterations. The winning tickets we find have won the initialization lottery: their connections have initial weights that make training particularly effective.

No ensemble of weak learners, but one strong learner hidden in the crowd ?