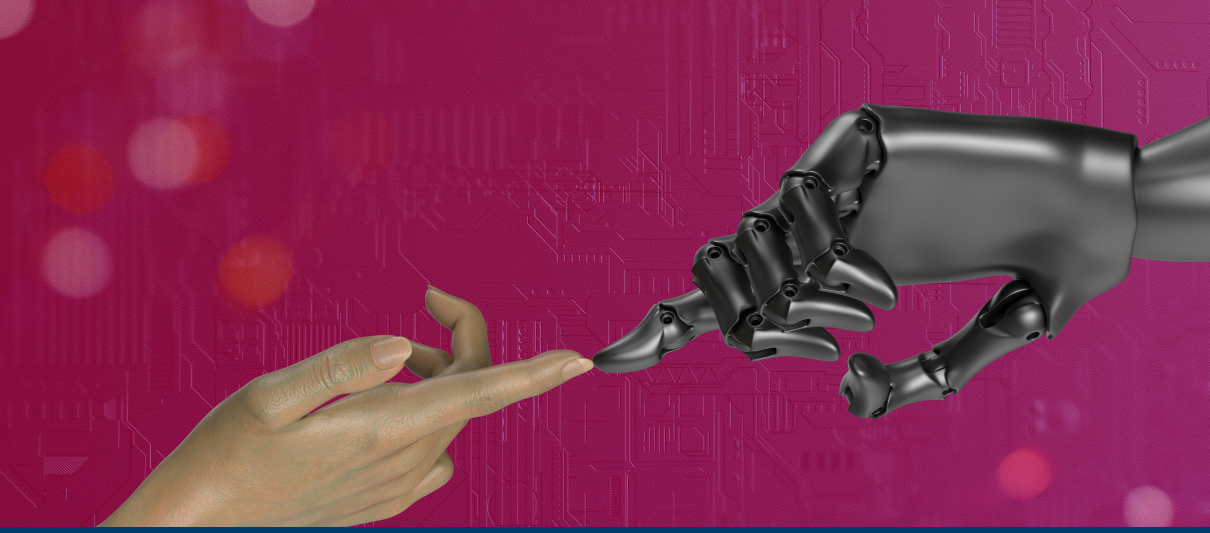


Hands-On Machine Learning

Dr. Derek Bridge | 2026/2027

UCC Coláiste na hOllscoile Corcaigh
University College Cork, Ireland

School of Computer Science and Information Technology
Scoil na Ríomheolaíochta agus na Teicneolaíochta Faisnéise

10. Ensembles

The wine dataset that we use in this lecture is available from the following dataset repository: https://archive.ics.uci.edu/dataset/186/wine+quality. However, we are using a version that I have modified slightly: (a) I merged separate CSV files for red and white wines into a single file, and (b) I reduced class imbalance a little.

The Wisdom of the Crowd

- The aggregated opinions of a diverse group of independent individuals yields the best judgment.
- E.g. trial-by-jury, committees, democracies, social network ratings.

Ensembles

- We can *combine* models into what is called an **ensemble**.
- The ensemble often out-performs the individual models that it contains.
- Ensembles can turn **weak learners** into **strong learners**.
  - Weak learners: one that are doing only slightly better than chance.
  - Strong learners: ones with much higher accuracy/lower error than chance.
- If this is to happen, there needs to be diversity among the individual models.
- Although an ensemble may be more accurate than the individual models within it, training costs and inference costs will both be higher.

Voting/Averaging

- The simplest approach.
- E.g. for classification:
  - Training: Train several different classifiers (e.g. Decision Tree, Logistic).
  - Inference: each model classifies the example; they vote on the result.
- Question: what do we do for regression?

Bagging

See next slides.

Boosting

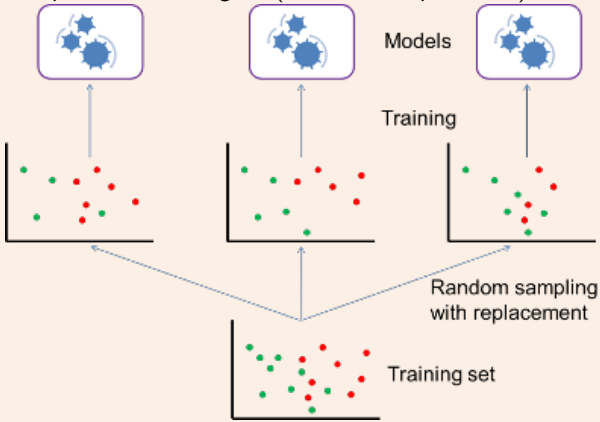
See next slides.

Stacking

Not covered in CS3315.

Bagging ('bootstrap aggregating')

- Training: train several models *of the same kind* (e.g. several Logistic Regression classifiers) *but on different random subsamples of the training set* (chosen *with replacement*).



- Inference: each model makes a prediction and these predictions are combined using voting/averaging.

Aggregation in Bagging

- In bagging, the aggregation (voting or averaging) is *not* usually weighted, i.e. all the models in the ensemble are treated equally.
- Contrast with boosting.

How many models should we use?

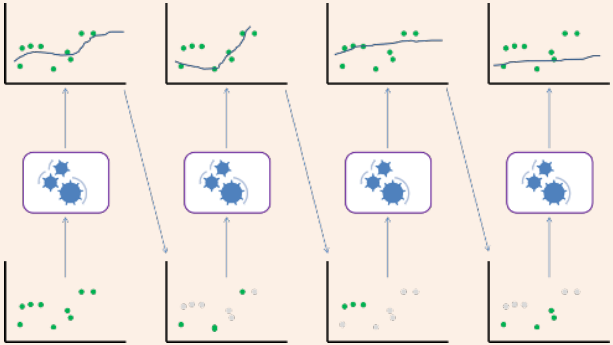
- To some extent, 'the more, the merrier': predictions typically become more reliable as the ensemble gets larger.
- Seldom is the accuracy of the ensemble lower than the accuracy of any of its members — but this is *not guaranteed*.
- The number of models is a **hyperparameter**: we can either guess how many models to use or choose its value using, e.g., grid search.

Random Forests

- A **Random Forest** uses **bagging** to create an ensemble of **Decision Trees**.
- Very popular and very successful.
- There are ways of increasing the diversity of the Decision Trees in a Random Forest — in addition to the random subsampling of the training set.
- For example, when learning the trees, instead of selecting the best feature for splitting, each tree might select the best among a randomly-chosen subset of the features.

Boosting

- Training: train several models *of the same kind* but *sequentially*.
  - In the first model, all training examples are treated equally.
  - In subsequent models, training examples on which the previous model made an error receive greater weight.



- Inference: each model makes a prediction and these predictions are combined using voting/averaging. However, we use a weighted vote/weighted mean, where a model's weight is based on its performance on the training set.

AdaBoost for Classification

- Training:

> Assign equal weights to each example in the training set;
>
> **for** $i \in 1 \ldots n\_estimators$ **do**
>
> Train classifier i on the weighted training set;
>
> Compute training accuracy for classifier i;
>
> Misclassified examples are given new, higher weights;
>
> **end**

- Inference:
  - Each classifier in the ensemble classifies the example.
  - Take a weighted vote using the training accuracies as the weights.

Weighted Training Examples

- AdaBoost assumes that, when we train the classifiers in the ensemble, they are sensitive to the weights of the training examples.
- So far, none of the classifiers we have studied does this — although it is easy to adapt kNN.
- Question: Suppose a classifier cannot handle weighted training examples. What can we do to give the same effect?

Gradient Boosting for Regression

- **Gradient Boosting** is similar to AdaBoost: it trains several models of the same kind, sequentially
- But, it does *not* use a weighted training set.
- Instead, each regressor is trained on the **errors** of the previous regressor.

Gradient Boosting: Training

- Train regressor 1 on $\boldsymbol{X}$ and $\boldsymbol{y}_1 = \boldsymbol{y}$.
- Use regressor 1 to predict $\hat{\boldsymbol{y}_1}$ for $\boldsymbol{X}$.
- Train regressor 2 on $\boldsymbol{X}$ and $\boldsymbol{y}_2 = \hat{\boldsymbol{y}_1} - \boldsymbol{y}_1$.
- Use regressor 2 to predict $\hat{\boldsymbol{y}_2}$ for $\boldsymbol{X}$.
- Train regressor 3 on $\boldsymbol{X}$ and $\boldsymbol{y}_3 = \hat{\boldsymbol{y}_2} - \boldsymbol{y}_2$.

Gradient Boosting: Inference

- Each regressor in the ensemble makes a prediction for the example.
- But the final prediction is the *sum* of these predictions.
- Question: Why does this make sense?

Image credits

Taken from tiktok.com

I based these diagrams on ones to be found in A. Géron: Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow (2nd edn), O'Reilly, 2019