Machine Learning

| Bootstrap – Aggregate (Bagging)

Prof. Dr. Mostafa Elhosseini

https://youtube.com/ElhosseiniAcademy

Agenda

- Bagging and pasting ensembles

Popular ensemble methods

- Most popular ensemble methods

  - Voting classifiers
  - **Bagging and pasting ensembles**
  - Boosting
  - Stacking ensembles

Teamwork for Problem-Solving

- we have to **take an exam** that consists of **100 true/false** questions on **many different topics**, including math, geography, science, history, and music. Luckily, we are allowed to call our **five friends**—Adriana, Bob, Carlos, Dana, and Emily—to help us. There is a small constraint, which is that all of them work **full time**, and they **don't have time to answer all 100 questions**, but they are more than happy to help us with a **subset** of them.

- What techniques can we use to get their help?

Bagging

- For each of the friends, **pick several random questions**, and ask them to answer them (**make sure every question gets an answer from at least one of our friends**). After we get the responses, answer the test by selecting the option that was **most popular** among those who answered that question.

- For example, if two of our friends answered “True” and one answered “False” on question 1, then we answer question 1 as “True” (if there are ties, we can pick one of the winning responses randomly).

Bagging

- Short for bootstrap aggregating
- Multiple models (often of the same type) are trained on different subsets of the training data.
- These subsets are created by **randomly sampling** the original dataset with **replacement** (**bootstrapping**).
- Each model is **trained independently**, and their predictions are **combined** (e.g., by averaging for regression or majority voting for classification) to produce the final output.
- Bagging helps reduce variance and prevent overfitting by leveraging the diversity among the models.

Bagging

- Bagging creates multiple predictors by training them on random subsets (bootstrap samples) of the original dataset. Each subset is typically **smaller than** the original dataset, which means each individual predictor has less data to learn from.
- Less data can lead to simpler models or models that **cannot fully capture** the underlying patterns in the data.
- This **increases the bias** of individual predictors compared to training them on the entire dataset.

Bagging

Variance Reduction

- Variance refers to how much a model's predictions would vary if trained on different data samples. In bagging, predictions from multiple predictors are averaged (or aggregated).
- This averaging reduces the impact of any one predictor's fluctuations (high variance) on the final prediction.
- Mathematically, averaging random variables reduces their variance, assuming the predictors' errors are not perfectly correlated.

Bagging

Bias Reduction

- While individual predictors may have higher bias due to being trained on subsets, the aggregation of predictions tends to balance out errors made by individual predictors.
- Predictors trained on different subsets are likely to overestimate or underestimate the target in different ways. By averaging their predictions, the ensemble reduces the systematic errors of individual predictors.

Bagging

- **Bootstrapping** is a statistical resampling method that involves **randomly drawing samples with replacement** from a dataset to create many "new" samples, each the **same size as the original**.

- These samples are called bootstrap samples, and each contains some instances **multiple times** and may **omit others**.

Pasting

- **Pasting** is similar to bagging but with one key difference: the subsets of the training data are created by **sampling without replacement**.
- This means each data point is used **at most once** in creating the subsets for training the individual models.

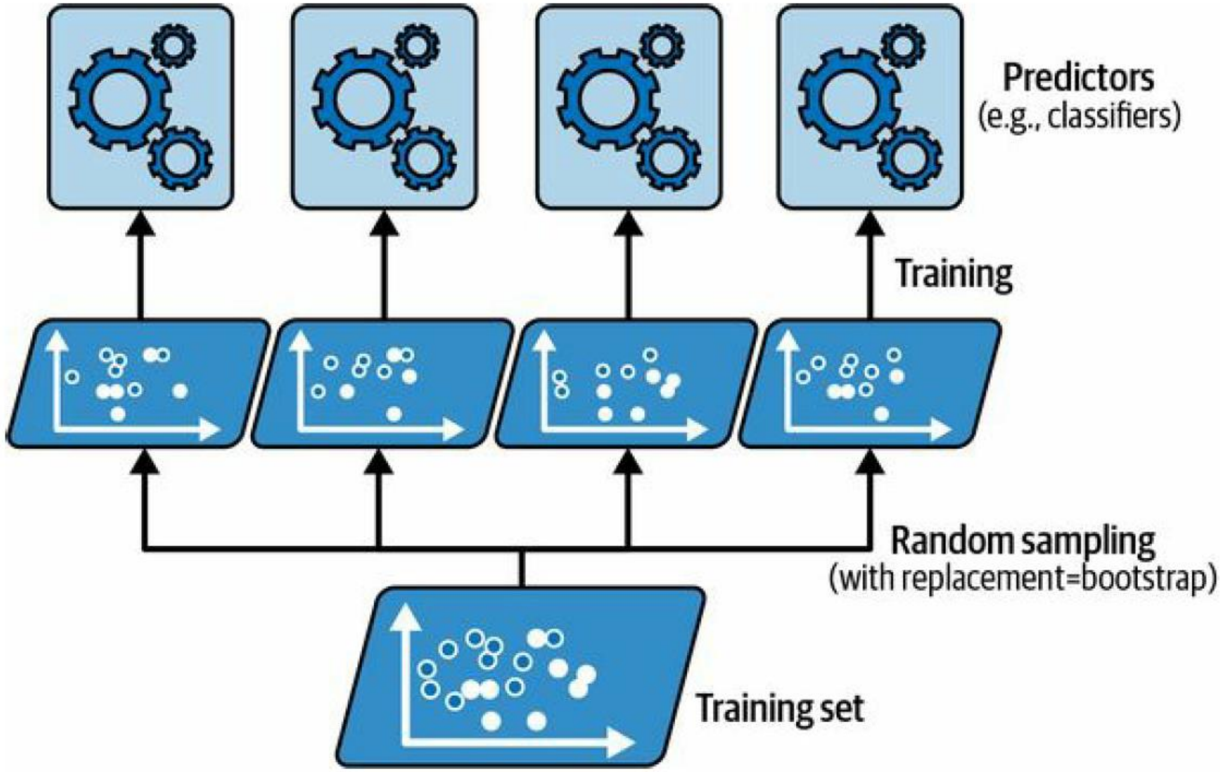
Bagging and Pasting

- Each individual predictor has a **higher bias** than if it were trained on the original training set, but **aggregation reduces both bias and variance**.

- Generally, the net result is that the ensemble has a similar bias but a **lower variance** than a single predictor trained on the original training set.

Bagging and Pasting

- **Predictors can all be trained in parallel**, via different CPU cores or even different servers. Similarly, **predictions** can be made in **parallel**.
- This is one of the reasons bagging and pasting are such popular methods: they **scale very well**.

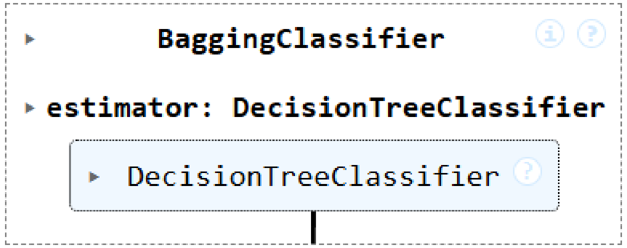


Bagging and Pasting in Scikit-Learn

- Scikit-Learn offers a simple API for both bagging and pasting: **BaggingClassifier** class (or **BaggingRegressor** for regression).

```python
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier

bag_clf = BaggingClassifier(DecisionTreeClassifier(), n_estimators=500,
                            max_samples=100, n_jobs=-1, random_state=42)
bag_clf.fit(X_train, y_train)
```



— 500 decision tree classifiers

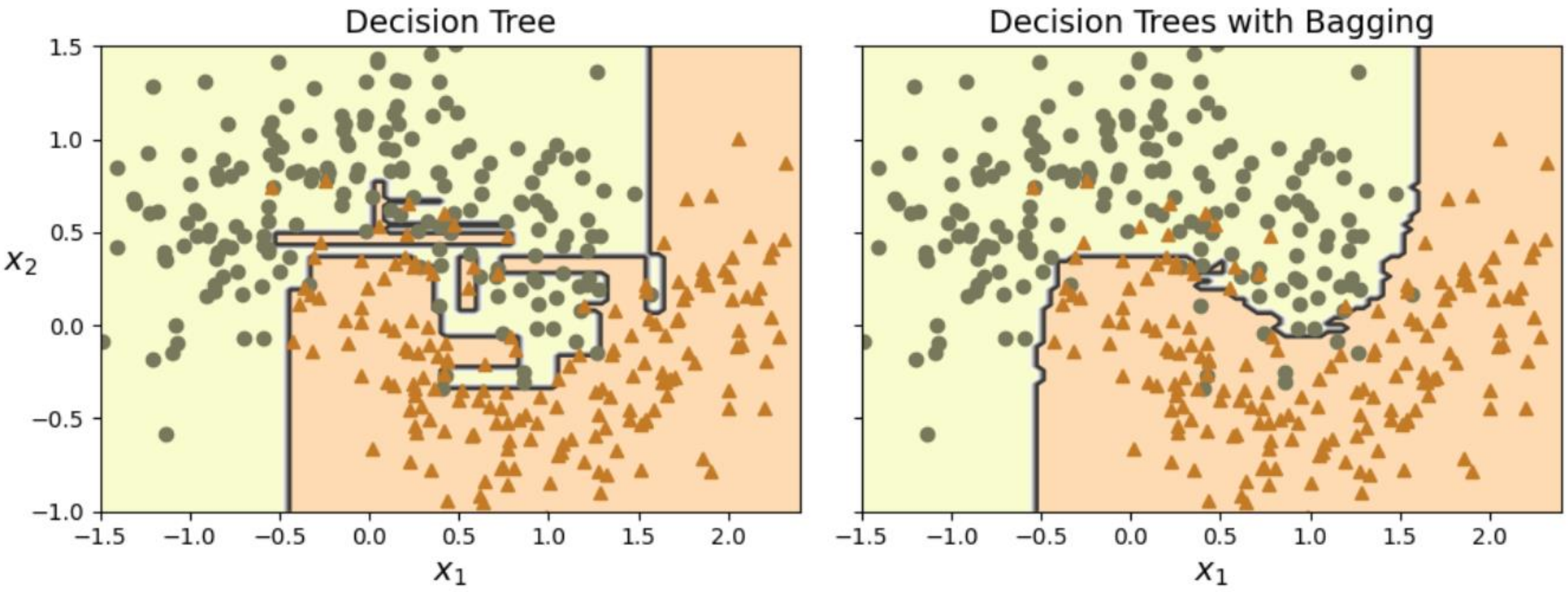
— 100 training instances

— –1 tells Scikit- Learn to use all available cores

— this is an example of **bagging**, but if you want to use **pasting** instead, just set **bootstrap=False**

Bagging and Pasting

- The ensemble has a comparable bias but a smaller variance (it makes roughly the same number of errors on the training set, but the decision boundary is less irregular).



Bagging and Pasting

- **Bagging** introduces a bit more diversity in the subsets that each predictor is trained on, so bagging ends up with a slightly higher bias than pasting; but the extra diversity also means that the predictors end up being less correlated, so the ensemble’s variance is reduced.
- Overall, **bagging** often results in better models, which explains why it’s generally preferred. But if you have spare time and CPU power, you can use **cross-validation** to evaluate both bagging and pasting and select the one that works best.

Next Lecture

- With bagging, some training instances may be **sampled several times** for any given predictor, while **others may not be sampled at all**.

- By default a BaggingClassifier samples m training instances with replacement (bootstrap=True), where m is the size of the training set.

- With this process, it can be shown mathematically that only about **63%** of the training instances are sampled on average for each predictor