EMMA ding

emmading.com

info@datainterviewpro.com

Ensemble Learning: Bagging, Boosting and Stacking

Lesson Structure

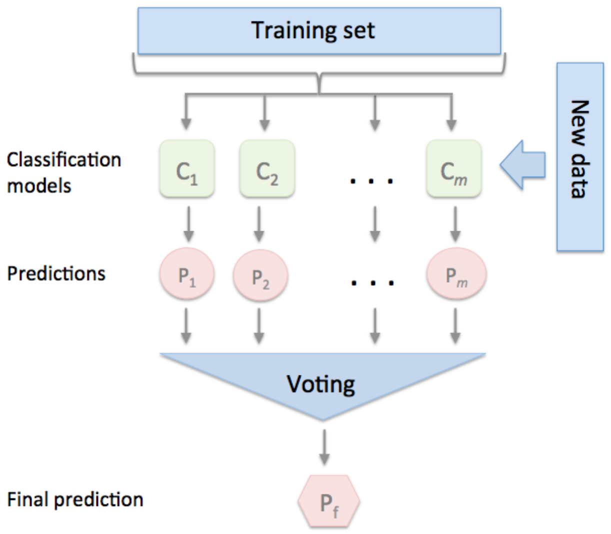
📌 Interview Questions

- What is ensemble learning? What are the examples of ensemble learning?
- What are boosting and bagging?
- What are the advantages of bagging and boosting?
- What are the differences between bagging and boosting?
- Why boosting models are good?
- Explain stacking.

▼ Ensemble Methods

📍 The main idea behind ensemble methods is that a group of "**weak learners**" can come together to form a "**strong learner**".

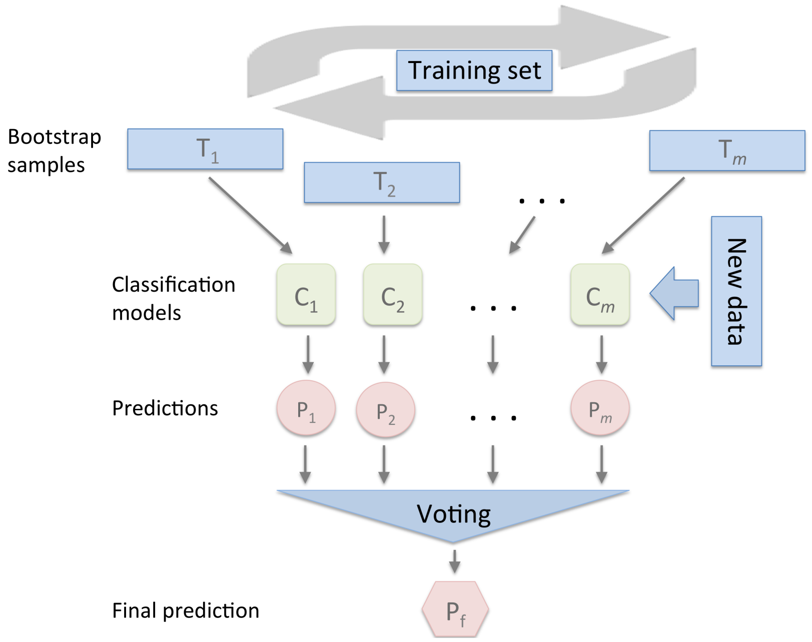
- Better **predictive performance** than could be obtained from any of the base leaners alone.
- If one base leaner is erroneous, it can be auto-corrected by others, so the final model is typically less prone to overfitting and more robust, unlikely to be influenced by small changes in the training data.



Ensemble for classification tasks. Image source: Raschka, S., Liu, Y., Mirjalili, V., & Dzhulgakov, D. (2022). *Machine learning with pytorch and Scikit-Learn: Develop machine learning and deep learning models with python*. Packt Publishing.

▼ Bagging (Bootstrap aggregation)

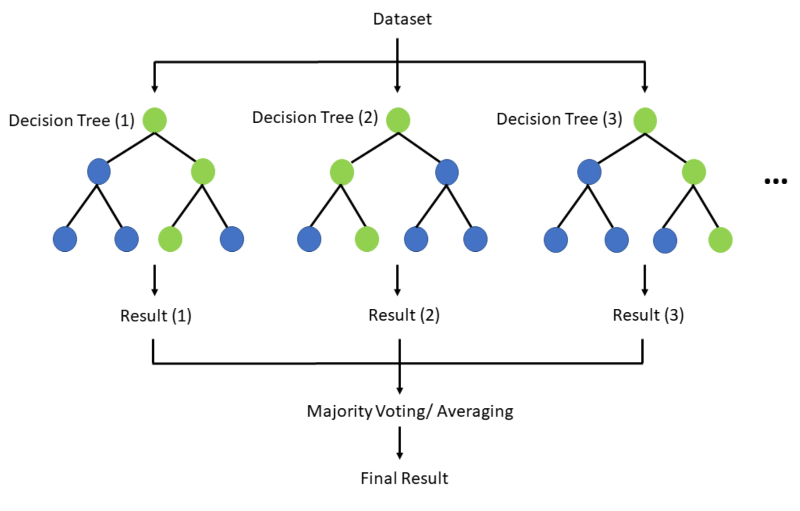
📍 Bagging is short for bootstrap aggregation. It builds several instances of an estimator on **bootstrap** samples of the original training data and then **aggregate** their individual predictions to form a final prediction.



Bagging for classification taksed. Image source: Raschka, S., Liu, Y., Mirjalili, V., & Dzhulgakov, D. (2022). *Machine learning with pytorch and Scikit-Learn: Develop machine learning and deep learning models with python*. Packt Publishing.

1. Create **bootstrap samples (**sampling **with** replacement**)** from the training data.
    - Ensure each sample is independent from others, as it does not depend on previous chosen samples when sampling.
2. Use a single learning algorithm to build a model using each sample.
    - Those models are built in parallel.
3. Use multiple models to make predictions and the predictions are combined using **voting** or **averaging**.
    - **Voting** for classification**:** the **most frequently** result (mode) is the final prediction.
    - **Averaging** for regression: **average** the results produced is the final result.

▼ Example: **Random forest**



Random Forest

Credit: TseKiChun via Wikimedia Commons

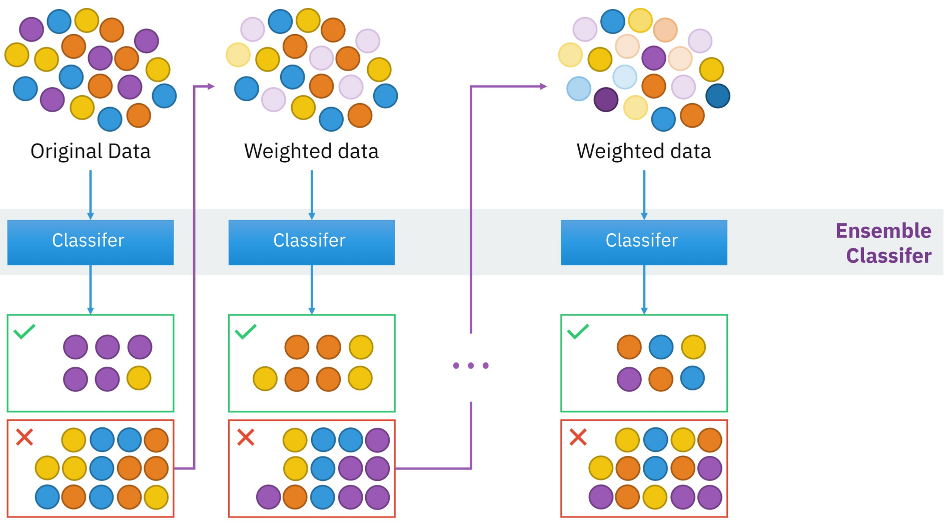
- Combines a set of **decision trees** to make predictions
- Decision trees are good candidates for bagging - they can capture complex interactions in the data, and if grown sufficiently deep, have relatively low bias.
- Since trees are noisy, they benefit greatly from the voting/averaging.

🌵 From a bias-variance tradeoff point of view, random forest starts with **low bias + high variance** (each tree is fully grown) and work towards **reducing variance** (by taking majority vote or averaging across trees).

▼ Boosting

🌻 Boosting improves the prediction power by training weak learners **sequentially**, each compensating the weaknesses of its predecessors.

- Start with a weak learner, gradually turn it into a strong learner by letting future weak learners focus on c**orrecting mistakes** made by previous learners.
    - Misclassified examples gain a higher weight than examples that are classified correctly, so future learners focus more on the examples that previous learners misclassified.
- Reduce the **bias** of the weak learner.



Credit: Sirakorn

1. Start with a weak learner (e.g., a shallow tree) better than random guess.
    a. Some examples that are correctly classified and some are not.
2. In the next iteration, the **weights** of the data are re-adjusted such that they can be corrected in the succeeding round.
    - lower weights to those were classified correctly.
    - higher weights to those were classified incorrectly.
3. This sequential process of giving higher weights to misclassified predictions continues until a **stopping criterion** is reached.
4. The final prediction is a weighted result of all weak learners.

▼ Example: **Gradient boosted trees**

- Train decision trees sequentially
- Optimize the residual loss of a tree by adding another tree
- It appears to outperform bagging on lots of problems and become the preferred choice.

🌱 From a bias-variance tradeoff point of view, gradient boosted trees start with **high bias + low variance** (first tree is shallow) and work towards reducing **bias** (by making the tree more complicated).

▼ Bagging vs Boosting

	individual learners	bias-variance	example
Bagging	independent, built in parallel	reduce variance	random forest
Boosting	dependent, built sequentially	reduce bias	gradient-boosted tree

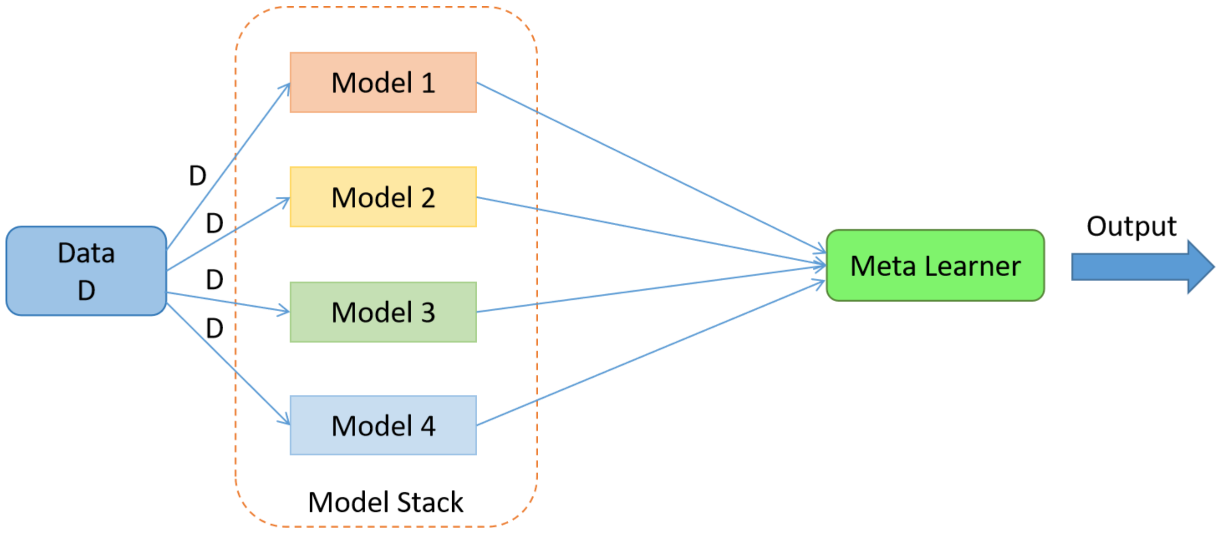
🍡 Bagging methods work best with strong and complex models (e.g., fully developed decision trees), while boosting methods usually work best with weak models (e.g., shallow decision trees).

▼ Stacking

Building a meta-model that takes the output of base learners as input.

Combining estimators to reduce their **biases**.

Can be applied to classification and regression problems.



Credit: Yash Khandelwal, https://www.analyticsvidhya.com/blog/2021/08/ensemble-stacking-for-machine-learning-and-deep-learning/

Two-level ensemble:

- Individual estimators that feed their predictions to the second level
- A combiner estimator is fit to the level-one estimator predictions to make the final prediction

▼ **Example**

A stacking model for a classification task.

- Individual classifiers: random forest and SVM
- Stacking classifier: logistic regression

▼ **Pros and Cons**

- In practice, a stacking predictor predicts as good as the best predictor of the base layer and sometimes outperforms it by combining the different strengths of the these predictors.
- Training a stacking predictor is computationally expensive.