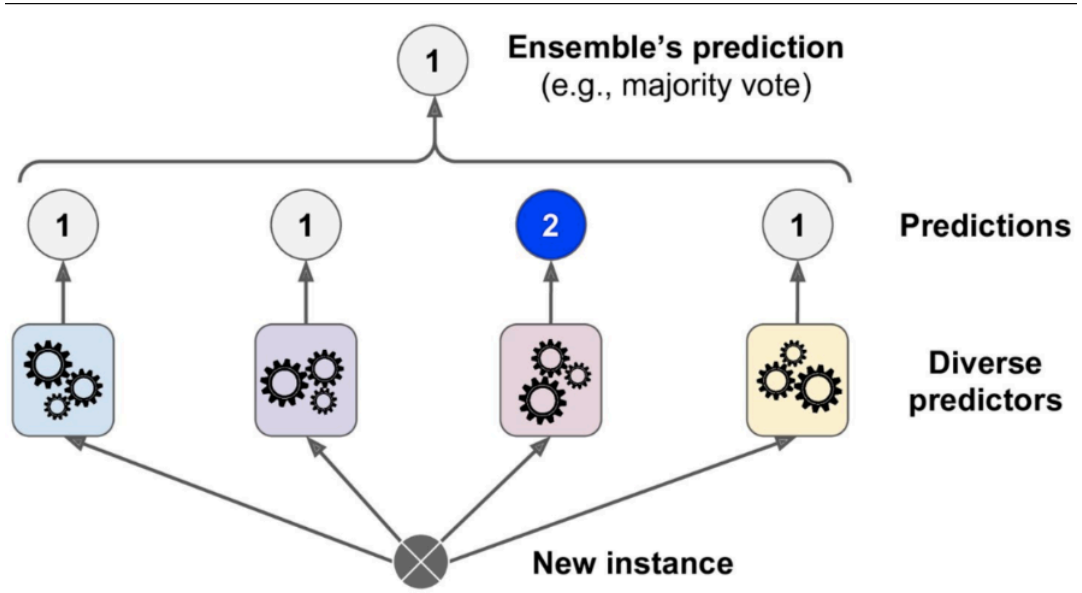
Ансамблирование моделей • Ensembling

Сегодня

Эльбрус буткемп

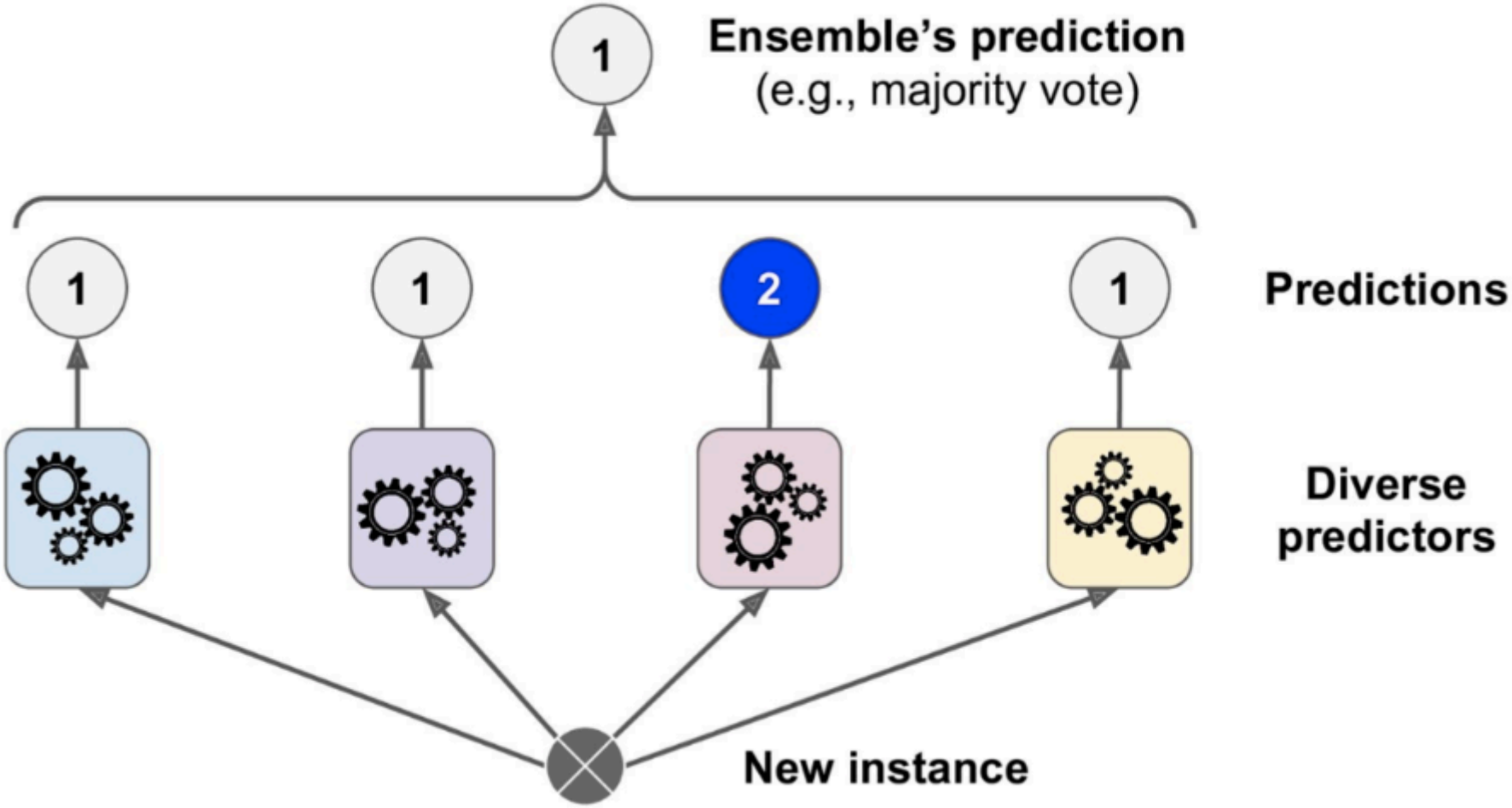
- ответим на вопрос "как можно соедининть несколько моделей?"
- разберем наиболее популярные варианты
- установим и запустим самые мощные библиотеки для работы с ансамблями моделей

Голосование / Voting



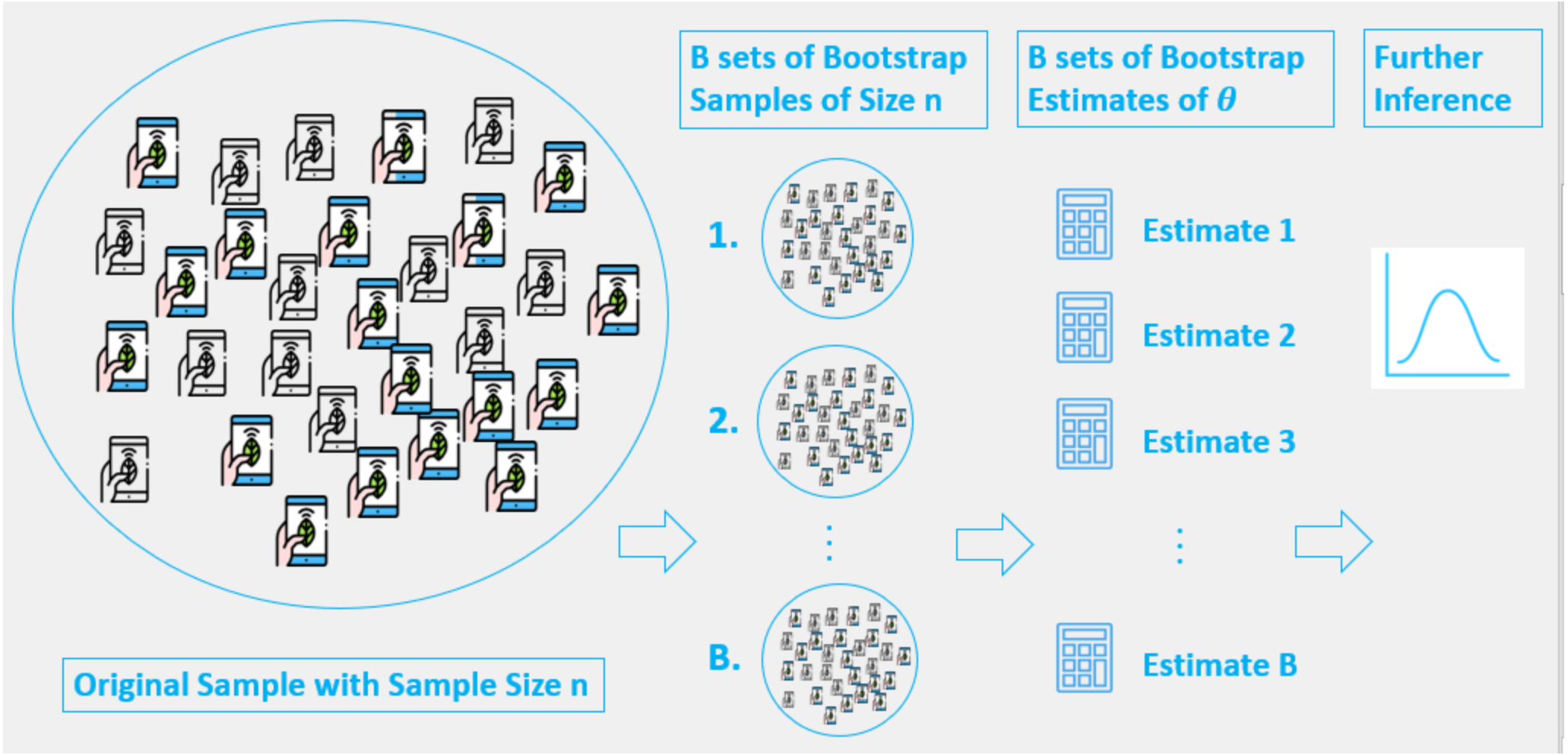
- **Hard voting**: классификаторы предсказывают классы - выбираем наиболее частотный
- **Soft voting**: классификаторы предсказывают вероятности - усредняем вероятности по классам

Голосование / Voting



> Что делать с задачей регрессии?

Бутстреп / Bootstrap



Бэггинг / Bagging

Эльбрус буткемп

1. Основная идея: формируем *boostrap* выборки
    - Обучаем одну простую модель на каждой выборке
2. Агрегируем предсказания всех моделей

Случайный лес / Random Forest

Эльбрус буткемп

1. Формируем *N* boostrap-выборок из исходного датасета
2. Обучаем **одно** дерево на каждой из *N* выборок
3. Для предсказания агрегируем результаты каждого дерева: наиболее частотный класс в случае классификации, усредненное значение в случае регрессии:

- Классификация - самый частотный класс
- Регрессия - усредняем предсказания

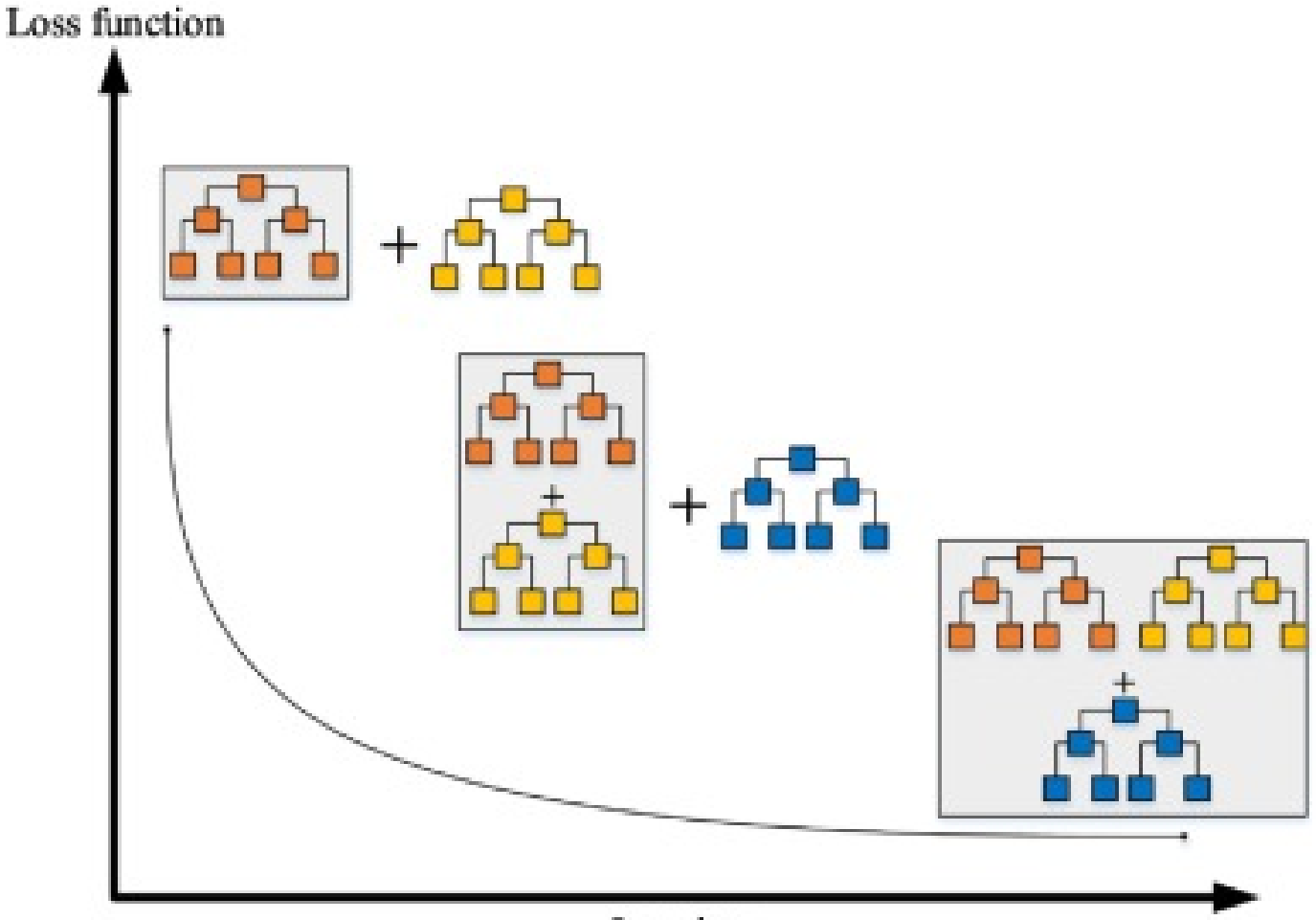
Параметры

- число деревьев: `n_estimators`
- + все те же, что для деревьев

`sklearn.ensemble` : scikit-learn.org

Бустинг / Boosting

- Строим базовые алгоритмы(почти всегда дерево) последовательно
- Каждый следующий исправляет суммарную ошибку предыдущих
- Решение принимаем взвешенным голосованием



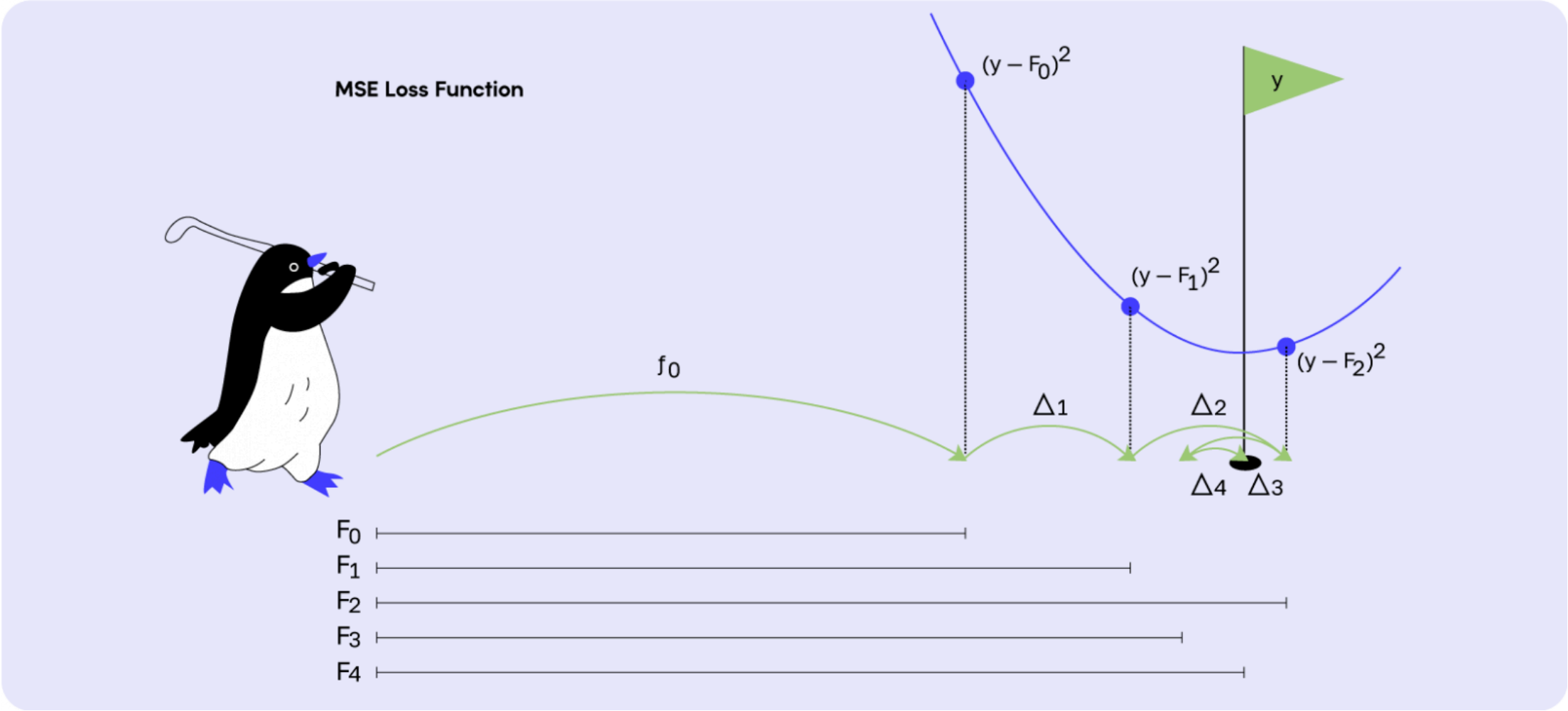
Градиентный бустинг / Gradient Boosting

Эльбрус буткемп

- x^i – это объекты, которые мы используем для обучения модели, их количество обозначается как N
- y^i – это правильные ответы для этих объектов
- $\hat{f}_i(x)$ - это предсказание i-ого дерева, которое мы обучаем
- $\hat{f}(x)$ - это итоговое предсказание всего бустинга
- $\hat{f}(x) = c_1 \cdot \hat{f}_1(x) + c_2 \cdot \hat{f}_2(x) + \cdots + c_k \cdot \hat{f}_k(x)$
- $Loss(y, f(x)) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(y^i, a(x^i))$ – это функция, которая показывает, насколько хорошо наша модель работает, чем меньше значение, тем лучше

Градиентный бустинг: Визуализация

Эльбрус буткемп



📝 Gradient boosting: Distance to target

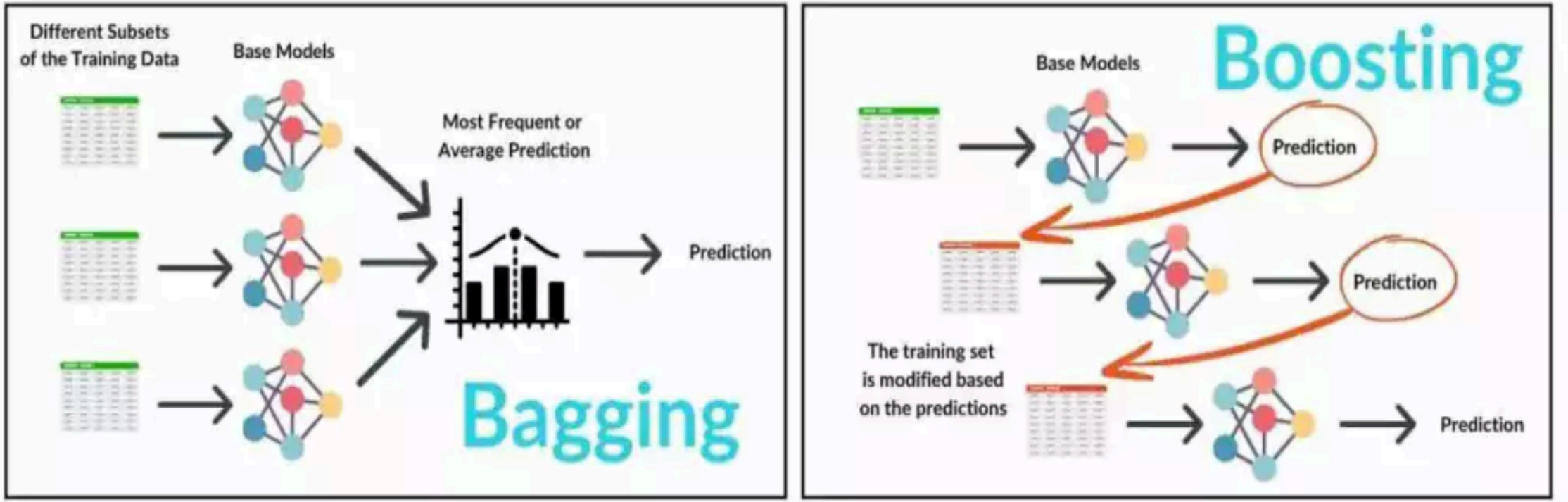
Градиентный бустинг / Gradient Boosting

Эльбрус буткемп

- Например, построены k алгоритмов, нужно построить $\hat{f}_{k+1}$ - ое дерево с c_{k+1} весом
- Функция потерь для одного i-го объекта:

$$\mathcal{L}(y^i, \hat{f}(x^i)) = \mathcal{L}(y^i, c_1 \cdot \hat{f}_1(x^i) + \cdots + c_k \cdot \hat{f}_k(x^i) + c_{k+1} \cdot \hat{f}_{k+1}(x^i))$$

Bagging vs Bosting



Градиентный бустинг / Gradient Boosting

Эльбрус буткемп

- Базовые модели могут быть любыми, но лучше деревьями
- Решает задачи классификации, регрессии, ранжирования (не очень быстро)
- Нужна большая выборка
- Реализации:
  - `sklearn.ensemble`: scikit-learn.org
  - Extreme Gradient Boosting / `xgboost`: xgboost.readthedocs.io
  - Light Gradient Boosting Machine / `lighgbm`: lightgbm.readthedocs.io
  - Categorical Boosting / `catboost`: catboost.ai
  - Про сравнения и детали реализаций можно почитать тут:
    - CatBoost vs. Light GBM vs. XGBoost
    - When to Choose CatBoost Over XGBoost or LightGBM [Practical Guide]

Итоги

Эльбрус буткемп

- **Voting**: берем произвольные модели, а результаты агрегируем произвольным образом
- **Бэггинг**: строим бустреп подвыборки и обучаем простые модели на каждой из выборок
  - пример: случайный лес(RandomForest)
- **Бустинг**: последовательно строим модели, каждая следующая исправляет ошибку предыдущих
  - Бустинг пока еще лучшее решение для табличных данных.