

Программная инженерия. Разработка ПО (Python для продвинутых специалистов. Машинное обучение)

Модуль: Предобработка данных и машинное обучение

Лекция 9: Ансамбли моделей

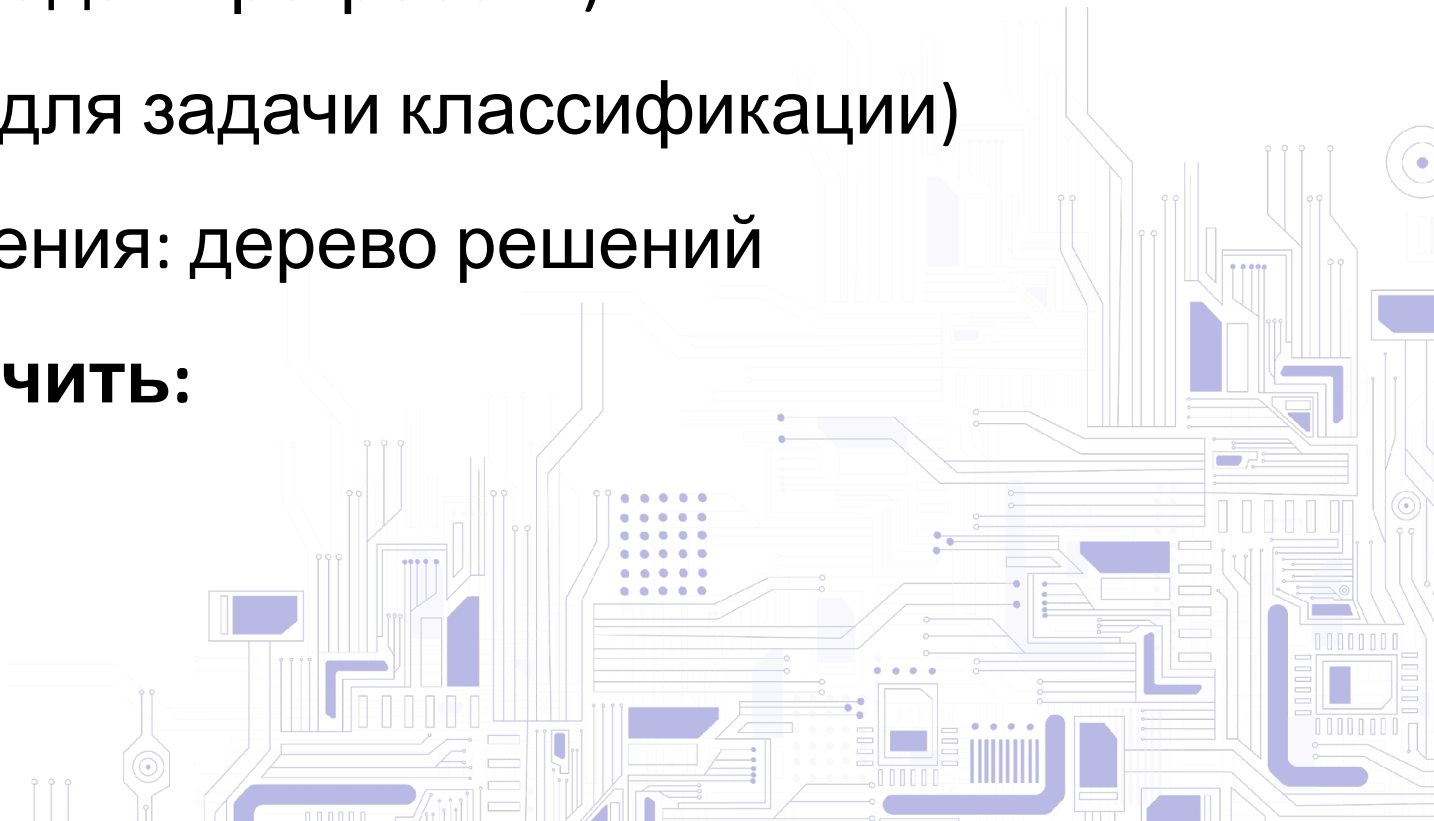
Дата: 16.06.2025

Что мы уже изучили:

- как готовить данные для моделей машинного обучения
- линейная регрессия (для задачи регрессии) и логистическая регрессия (для задачи классификации)
- алгоритм машинного обучения: дерево решений

Что планируем сегодня изучить:

- ансамбли моделей



Решающее правило

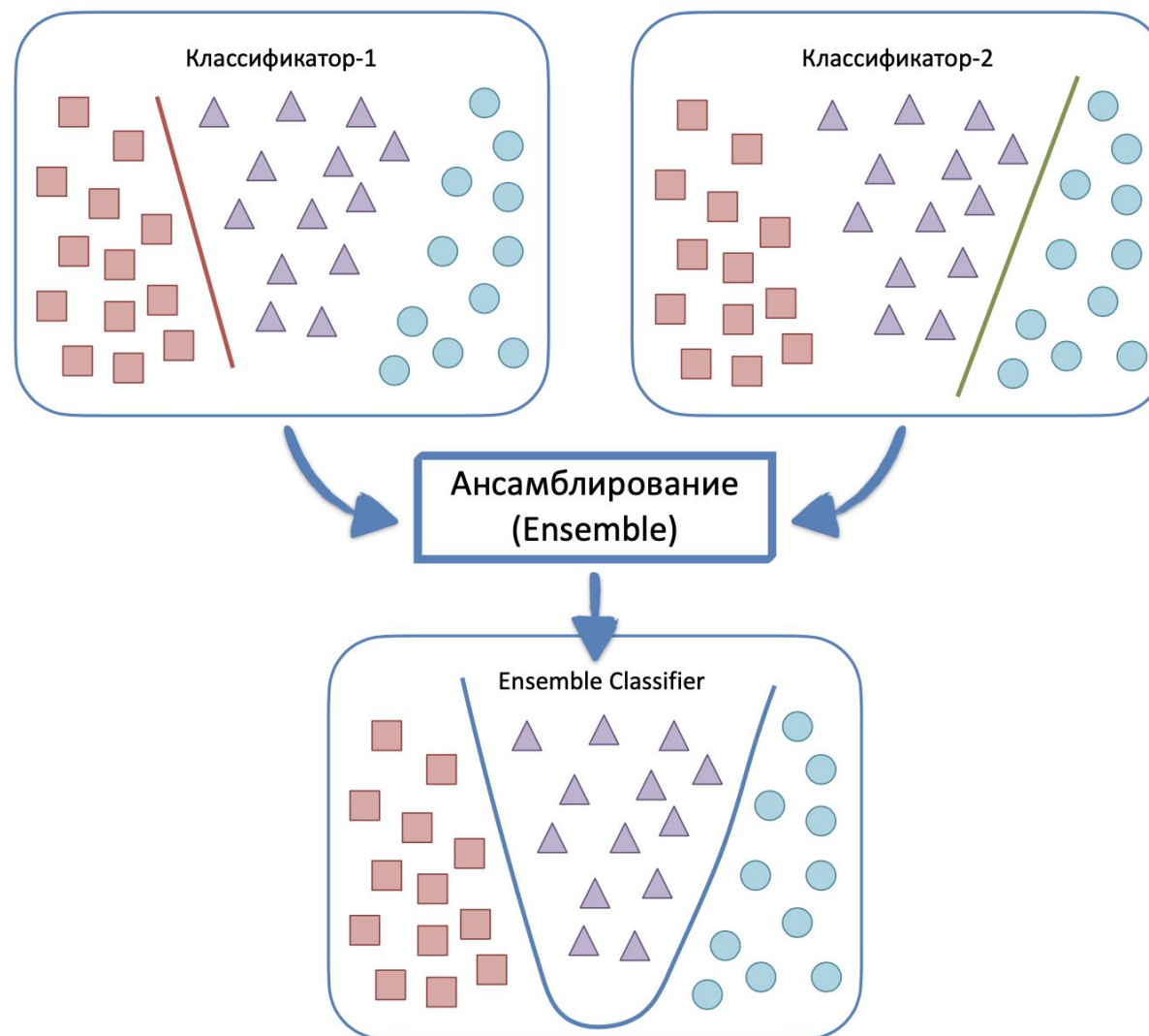
Еще **Аристотель** в 350 г. до н.э писал в «Политике»:

«Возможно, что многие, хоть по отдельности и не совсем хорошие люди, но когда они соберутся вместе, могут быть лучше, не индивидуально, а коллективно, точно так же, как общественные обеды, в которые многие вносят свой вклад, лучше, чем если бы поставлялся за счет одного человека».

Среднее мнение группы людей может быть точнее, чем мнение одного эксперта (эффект «мудрости толпы»). Например, в 1906 году **Фрэнсис Гальтон** заметил, что среднее арифметическое массы быка на ярмарке была точнее оценки любого отдельного участника.

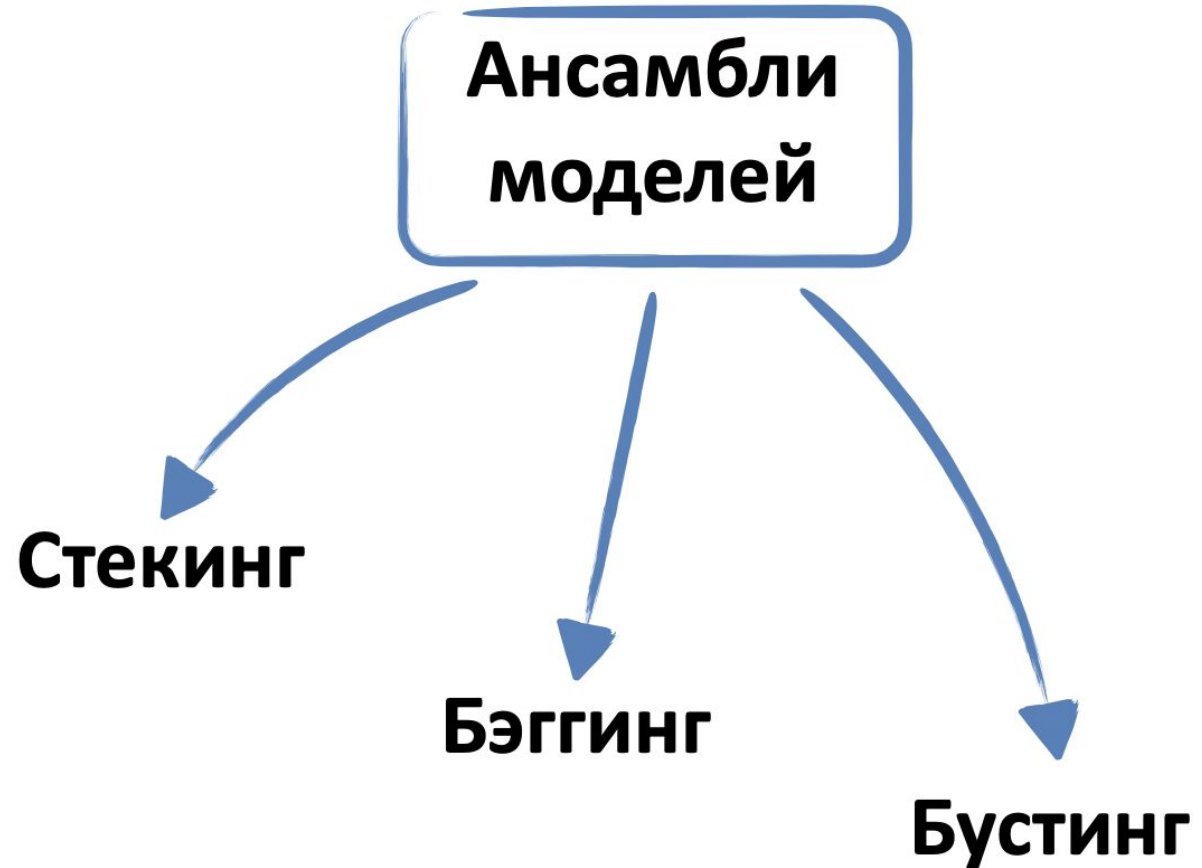
Идея ансамблей моделей

Основная идея
заключается в том, что
отдельно обучаются
несколько моделей, а
далее их предсказания
комбинируются



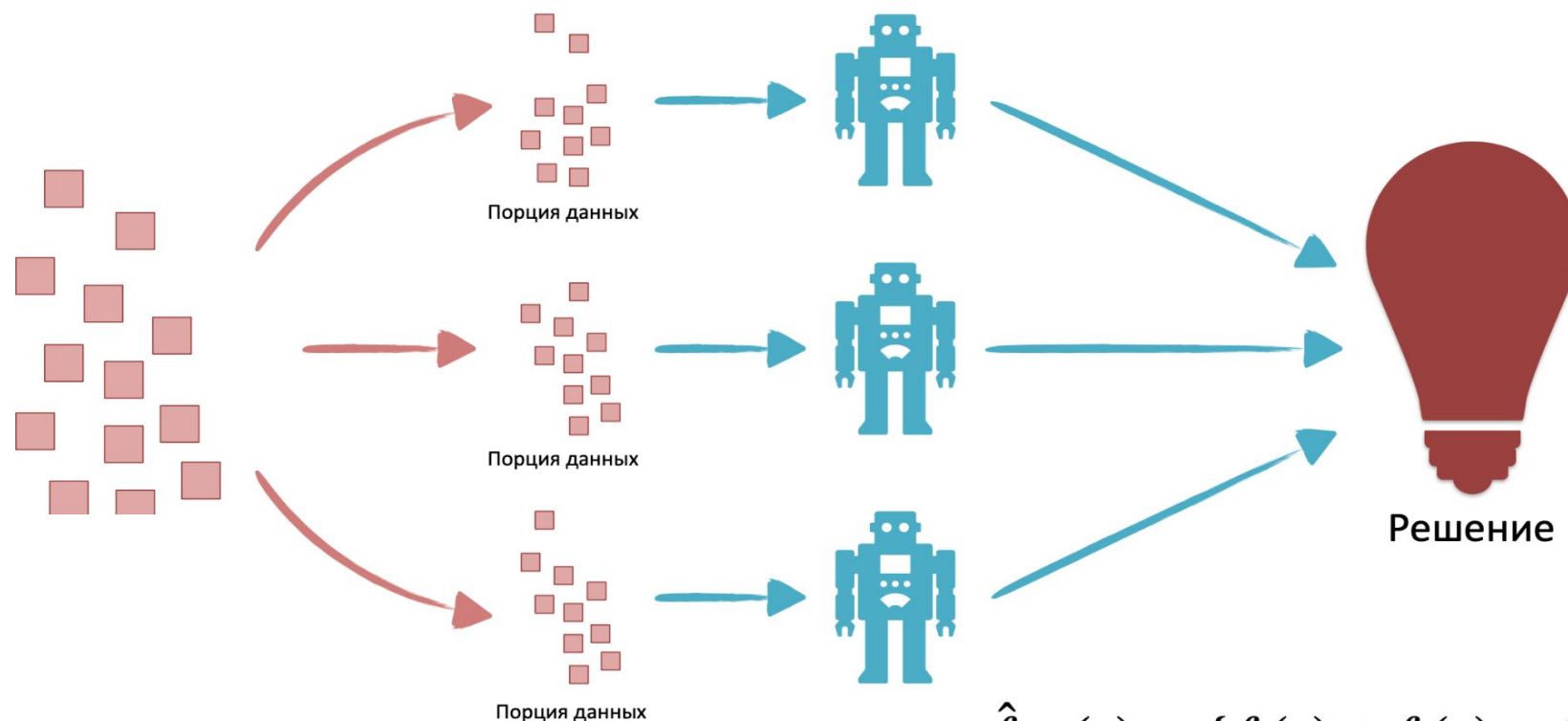
Виды ансамблей моделей

- Стекинг
- Бэггинг
- Бустинг



Bagging (Bootstrap aggregating)

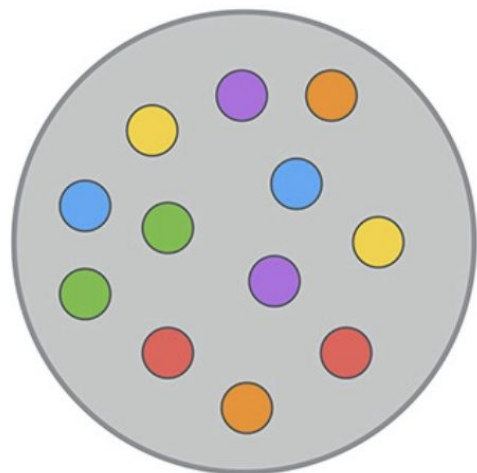
Идея - несколько моделей обучаются **независимо** на **различных подмножествах данных**, далее прогнозы объединяются



$$\hat{f}_{bag}(x) = \{f_1(x) + f_2(x) + f_B(x)\} / B$$

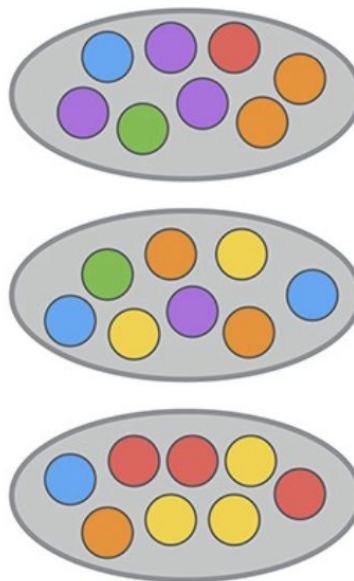
Bagging (Bootstrap aggregating)

Исходная выборка



Статистика по
выборке

Бутстрэп выборки



Статистики по
бутстрэп выборкам

Статистика 1

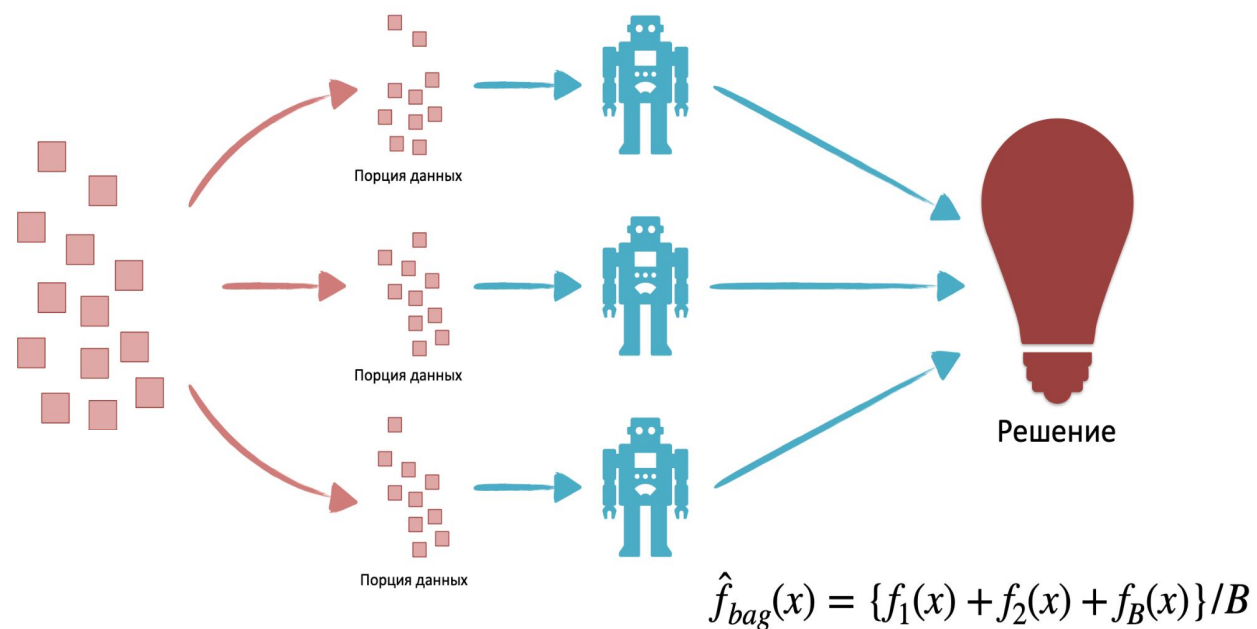
Статистика 2

Статистика 3

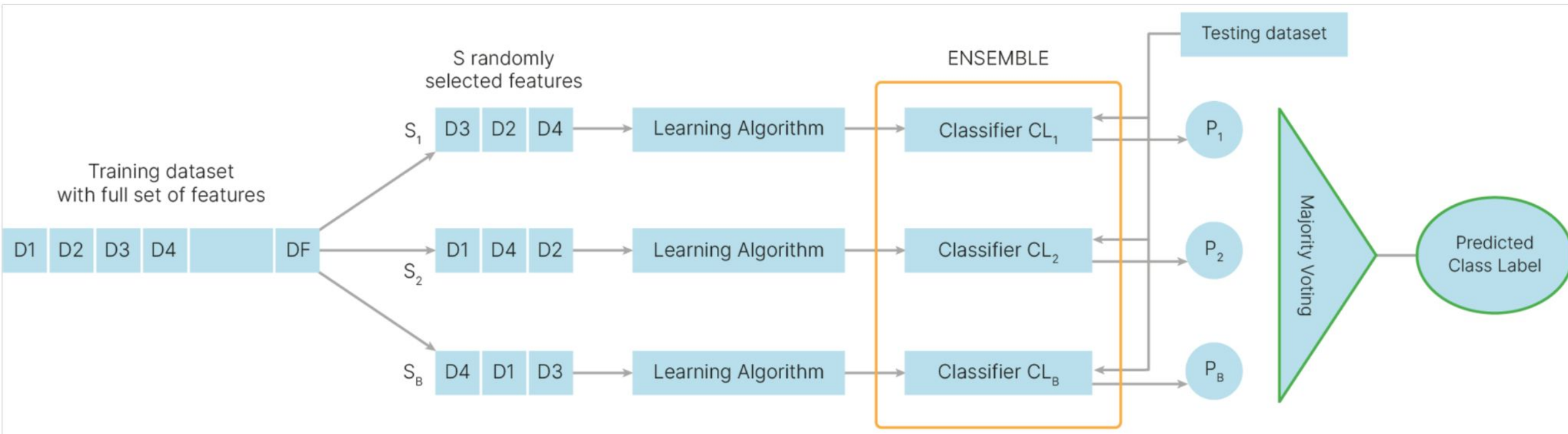
Бутстрэп
распределение

Bagging. Алгоритм

- 1) Создание подвыборки:** Из обучающего набора данных случайным образом выбираются подвыборки (выборки с возвращением), которые могут пересекаться.
- 2) Обучение базовых моделей:** На каждой из подвыборок данных обучается своя базовая модель (например, решающее дерево)
- 3) Прогнозы:** Обученные модели используются для генерации прогнозов на новых данных
- 4) Агрегация прогнозов:** Прогнозы, сгенерированные разными моделями, объединяются, например, путем голосования для классификации или усреднения для регрессии.



Метод случайных подпространств



Случайный лес



Лео Брейман в 2003 году

RF (random forest) объединяет в себе 2 идеи:

1. бэггинг
2. случайные подпространства признаков

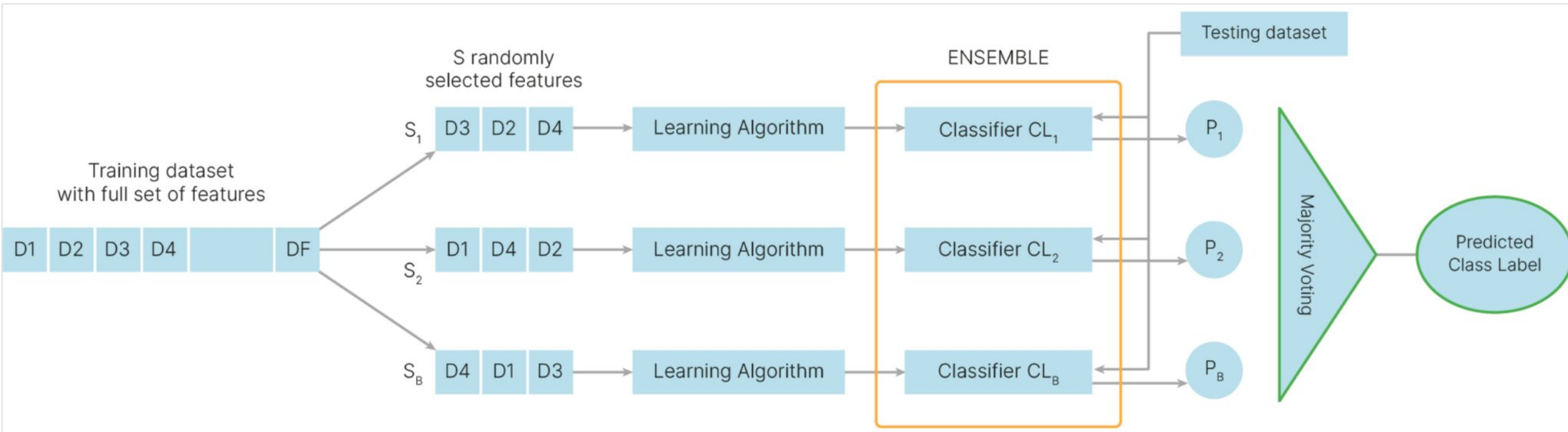
Используется:

1. для классификации
2. для регрессии



Адель Катлер

Метод случайных подпространств



В RF реализовано с помощью гиперпараметра **max_features**

<https://scikit-learn.org/1.5/modules/generated/sklearn.ensemble.RandomForestRegressor.html>

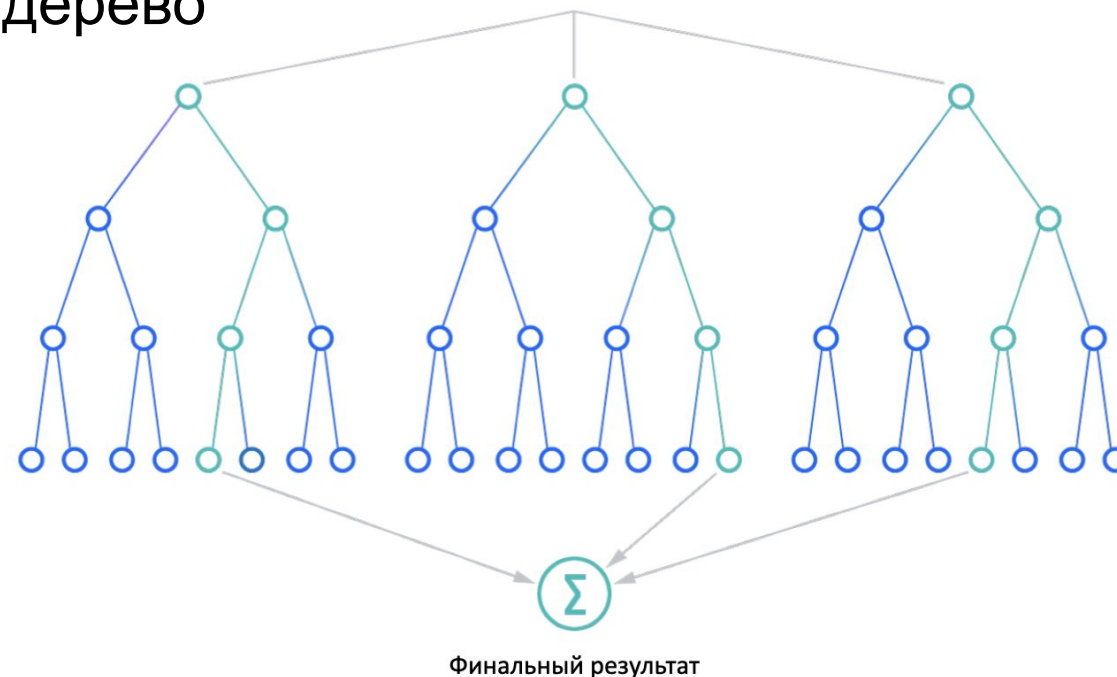
Случайный лес

RF (random forest) — это
множество решающих деревьев.

В задаче **регрессии** их ответы
усредняются,

в задаче **классификации**
принимается решение
голосованием по большинству.

с помощью **bootstrap** формируются
выборки и на каждой выборке строится
дерево



Случайный лес

- Выбирается **подвыборка** обучающей выборки размера `sample size` (м.б. с возвращением) – по ней строится дерево. (для каждого дерева — своя **подвыборка**)
- Для построения каждого разделения в дереве просматриваем **`max_features`** случайных признаков (**для каждого нового разделения — свои случайные признаки**)
- Выбираем наилучший признак и разделение по нему (по заранее заданному критерию). Дерево строится, как правило, до исчерпания выборки (пока в листьях не останутся представители только одного класса) - то есть **глубокие переобученные деревья**.

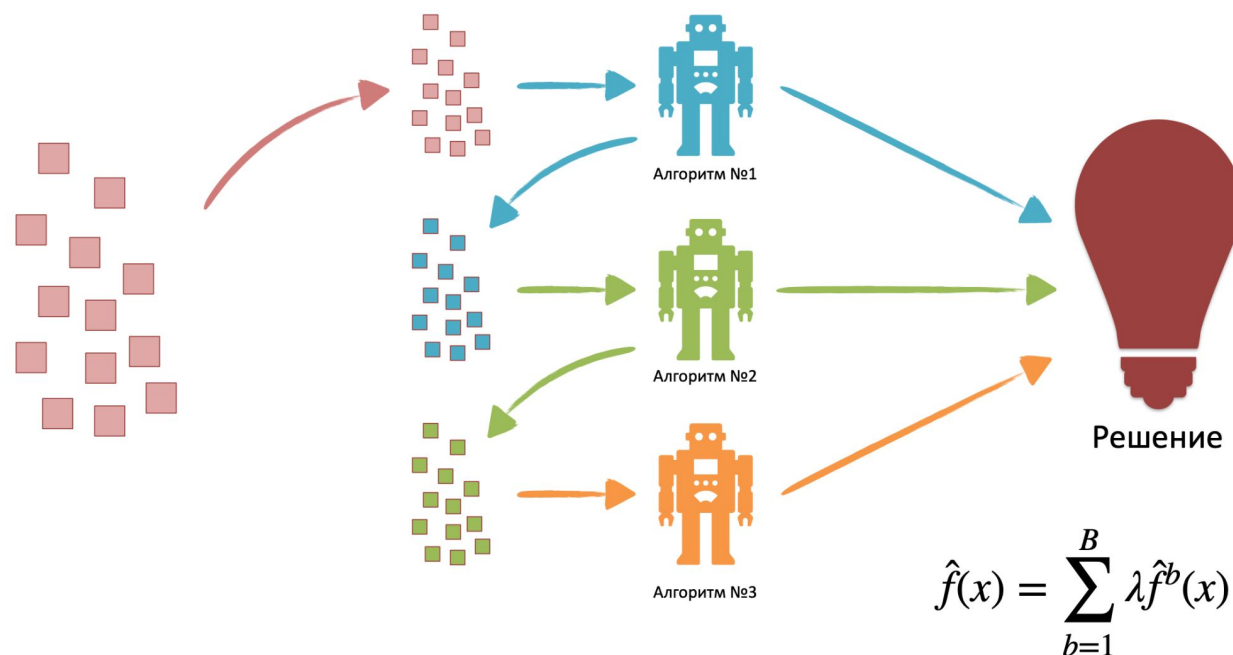
Случайный лес

Практика



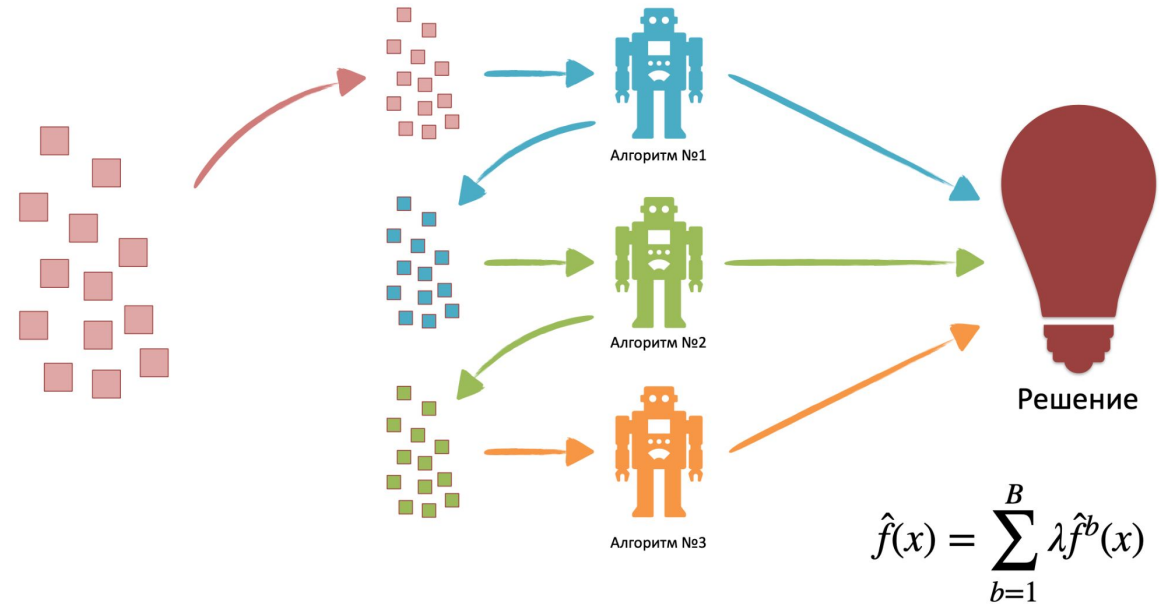
Бустинг (Boosting)

Идея: базовые модели строятся **последовательно**, причем каждая следующая базовая модель **уточняет** предсказание предыдущих (при этом всех).



Бустинг (Boosting). Алгоритм

- **Обучение базовой модели:** обучается базовая модель на исходных данных
- **Вычисление ошибок:** Для каждого образца в обучающем наборе рассчитывается ошибка базовой модели, сравнивая её предсказание с фактическим значением.
- **Создание новой модели:** Создаётся новая модель, которая пытается скорректировать ошибки предыдущей модели. Она уделяет больше внимания тем образцам, на которых предыдущая модель ошиблась.
- **Взвешенное голосование:** Прогнозы всех созданных моделей комбинируются с разными весами. Обычно модели, справившиеся с ошибками лучше, получают



Как появился бустинг?

В 1995 году Йоав Фройнд (Yoav Freund) и Роберт Шапир (Robert Schapire) предложили более общую схему для композиции алгоритмов – с разными весами (ADABOOST)

$$a(x) = \sum_{i=1}^T w_i a_i(x), \text{ где } w_i - \text{это веса, } a_i(x) - \text{это базовые алгоритмы}$$



Yoav Freund



Robert Schapire

AdaBoost. Можно ли его улучшить?

Jerome Isaac Friedman



Можно придумать много других функций для разработки (синтеза) новых алгоритмов бустинга. И здесь возникает естественный вопрос. А можно ли создать универсальный алгоритм бустинга, который бы работал с произвольной гладкой и дифференцируемой функцией потерь?

Оказывается ДА, можно и такой подход получил название **градиентного бустинга**.

Впервые градиентный бустинг представил Jerome Friedman (Джером Фридман) в 1999 году

Градиентный бустинг. Алгоритм

1. Инициализация:

- Инициализировать ансамбль средним предсказанием: $F_0(x) = avg$.
- Вычислить начальные остатки: $r_{i0} = y_i - F_0(x_i)$, где y_i - истинное значение, x_i - обучающий пример.

2. Для каждой итерации t от 1 до T , где T - количество базовых моделей:

a. Обучение базовой модели a_t на обучающих данных, предсказывающей остатки r_{it} .

b. Вычислить множитель γ_t путем решения задачи оптимизации:

$$\gamma_t = \arg \min_{\gamma} \sum_{i=1}^N L(y_i, F_{t-1}(x_i) + \gamma \cdot a_t(x_i))$$

где L - функция потерь (например, квадратичная), y_i - истинное значение,

$F_{t-1}(x_i)$ - предсказание ансамбля на предыдущей итерации, $a_t(x_i)$ - предсказание базовой модели a_t .

c. Обновить ансамбль: $F_t(x) = F_{t-1}(x) + \gamma_t \cdot a_t(x)$.

d. Обновить остатки: $r_{it} = y_i - F_t(x_i)$.

3. Окончательное предсказание: Итоговое предсказание для нового примера x : $F_T(x) = F_0(x) + \sum_{t=1}^T \gamma_t \cdot a_t(x)$

Бустинг



ПЕРЕДОВАЯ
ИНЖЕНЕРНАЯ ШКОЛА
УНИВЕРСИТЕТА ИННОПОЛИС

Практика



1. Ансамбли в машинном обучении. URL: <https://education.yandex.ru/handbook/ml/article/ansambli-v-mashinnom-obuchenii>
2. Random Forest, метод главных компонент и оптимизация гиперпараметров: пример решения задачи классификации на Python. URL: <https://habr.com/ru/companies/ruvds/articles/488342/>
3. Объединение моделей для методов ансамблевого обучения. URL: <https://github.com/xsolare/neyronki/blob/master/README.md#25>
4. Открытый курс машинного обучения. Тема 5. Композиции: бэггинг, случайный лес. URL: <https://habr.com/ru/companies/ods/articles/324402/>
5. Градиентный бустинг. URL: <https://education.yandex.ru/handbook/ml/article/gradientnyj-busting>
6. ДЕРЕВЬЯ И БУСТИНГ. URL: <https://logic.pdmi.ras.ru/~sergey/teaching/mlspsu22/16-boosting.pdf>



Передовые
инженерные
школы



МИНОБРНАУКИ
РОССИИ



УНИВЕРСИТЕТ
ИННОПОЛИС



онлайн
университет

Спасибо за внимание

