GeekBrains

Ансамблирование и использование деревьев решений в задачах машинного обучения.

Урок 7
На этой лекции вы найдете ответы на такие вопросы как:

- Как работают деревья решений
- Чем отличаются деревья решений регрессии и классификации
- Критерий остановки алгоритма
- Ансамблирование

Булгакова Татьяна

Преподаватель в GeekBrains, Нетология,Skillfactory

С 2010 года занимаюсь DataScience и NN. Фрилансер

- Участвовала в разработке программы по настройке оборудования для исследования пространственного слуха китообразных НИИ ИПЭЭ РАН
- Участвую в разработке рекомендательных систем по настройке нейростимуляторов для медицинских центров
- Работаю над курсом по нейронным сетям

План курса

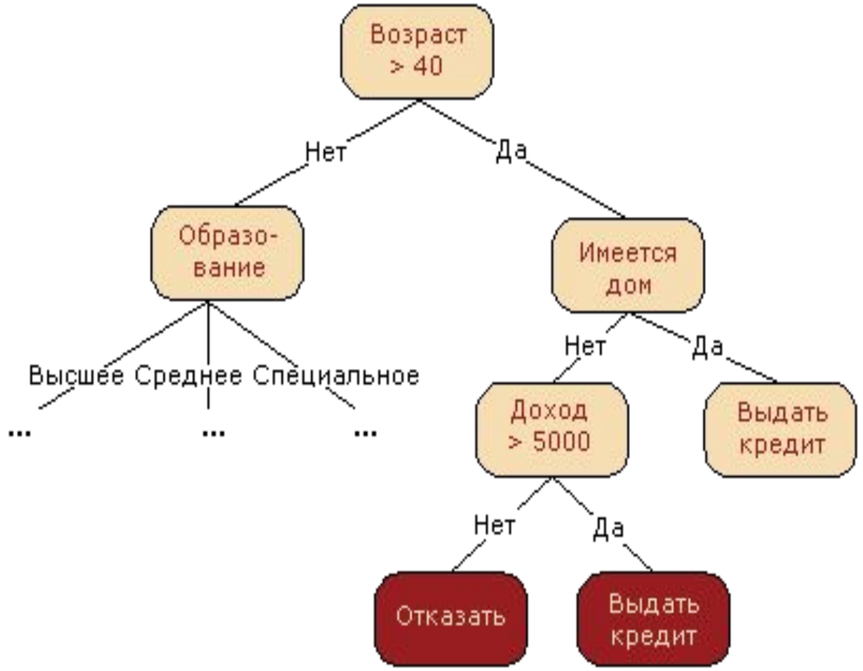
1. Первичный и визуальный анализ данных
2. Описательные статистики в контексте EDA. Корреляция и корреляционный анализ
3. Регрессия и использование библиотеки Scikit-learn в задачах обучения с учителем
4. Классификация и использование логистической регрессии в задачах классификации
5. Функционалы ошибки и поиск оптимальных параметров в задачах машинного обучения
6. Проблема переобучения и недообучения модели. Кросс-валидация и регуляризация.
7. Ансамблирование и использование деревьев решений в задачах машинного обучения
8. Генерация признаков. Методы отбора признаков. Подбор гиперпараметров.
9. Обучение без учителя. Понижение размерности. Алгоритмы понижения размерности
10. Кластеризация и решение задачи группировки данных в машинном обучении

Что будет на уроке сегодня

- 📌 Как работают деревья решений
- 📌 Чем отличаются деревья решений регрессии и классификации
- 📌 Критерий остановки алгоритма
- 📌 Ансамблирование

Что такое дерево решений

Дерево решений представляет собой способ организации решающих правил в иерархической структуре, состоящей из узлов и листьев. Узлы содержат решающие правила и выполняют проверку соответствия примеров этим правилам по определенному атрибуту обучающего набора данных.



Что такое дерево решений

1. Дерево решений моделирует принятие решений в виде древовидной структуры.
2. Узлы дерева представляют собой вопросы или условия на основе признаков данных.
3. Ветви от корня разделяют данные на подгруппы в зависимости от значений признаков.
4. Цель дерева решений - разделить данные таким образом, чтобы в каждой ветви или листе находились данные с максимальной однородностью.
5. Решение для нового примера данных принимается, следуя по дереву от корня до соответствующего листа, основываясь на ответах на вопросы и условиях.

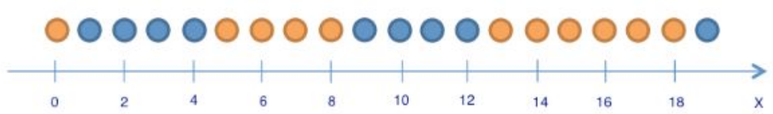
Что такое дерево решений

Энтропия Шеннона - критерий основан на понятиях теории информации, а именно — информационной энтропии.

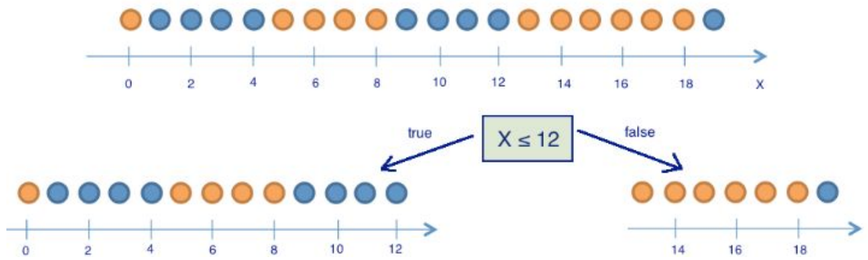
$$S = -\sum_{i=1}^{N} p_i \log_2 p_i$$

pi - вероятности нахождения системы в i-ом состоянии.

Деревья решений классификации



$$S_0 = -\frac{9}{20}\log_2\frac{9}{20} - \frac{11}{20}\log_2\frac{11}{20} \approx 1.$$



$$S_1 = -\frac{5}{13}\log_2\frac{5}{13} - \frac{8}{13}\log_2\frac{8}{13} \approx 0.96.$$

$$S_2 = -\frac{1}{7}\log_2\frac{1}{7} - \frac{6}{7}\log_2\frac{6}{7} \approx 0.6$$

Деревья решений классификации

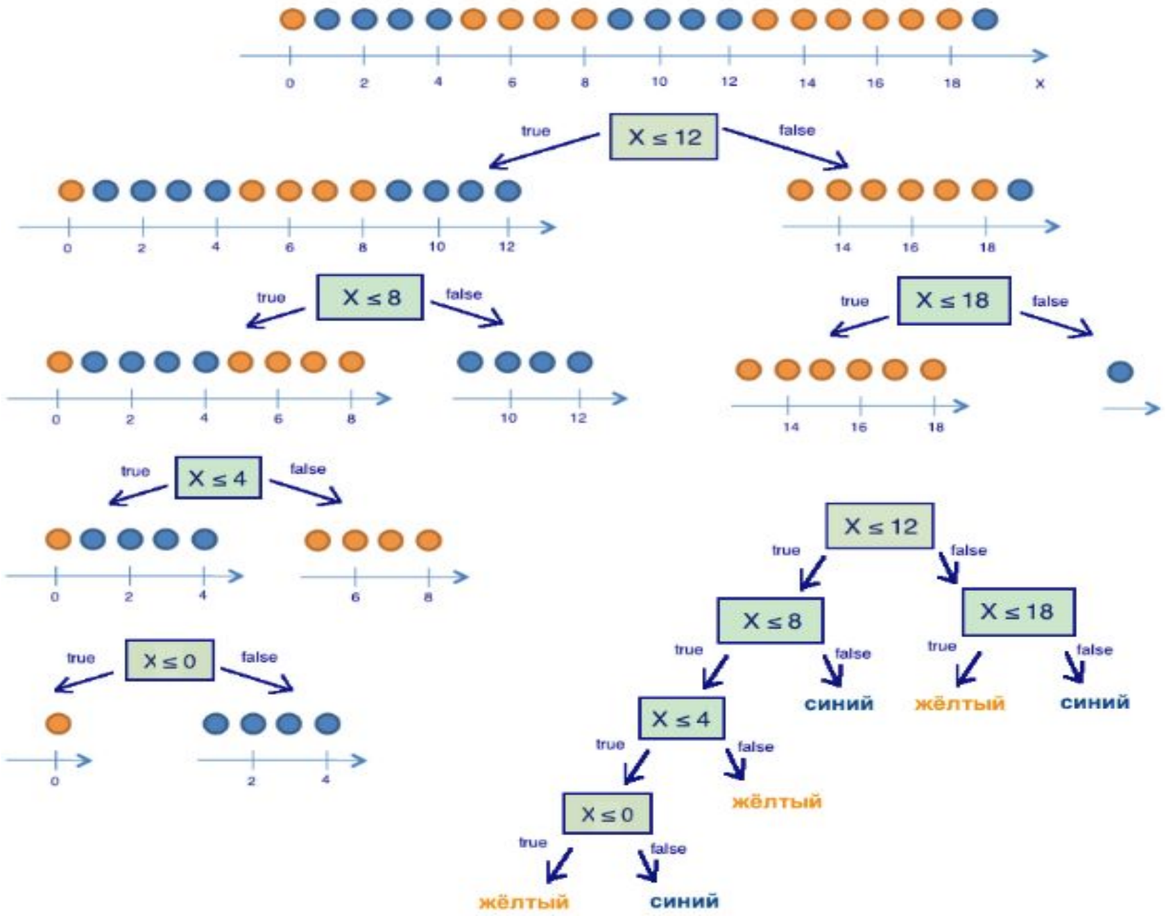
Энтропия является мерой хаоса (или неопределенности) в системе, уменьшение энтропии называется приростом информации.

$$IG(Q) = S_O - \sum_{i=1}^{q} \frac{N_i}{N} S_i,$$

где *q* – число групп после разбиения,
Ni – число элементов выборки, у которых признак *Q* имеет *i*-ое значение.

$$IG(x \leq 12) = S_0 - \frac{13}{20} S_1 - \frac{7}{20} S_2 \approx 0.16.$$

Деревья решений классификации

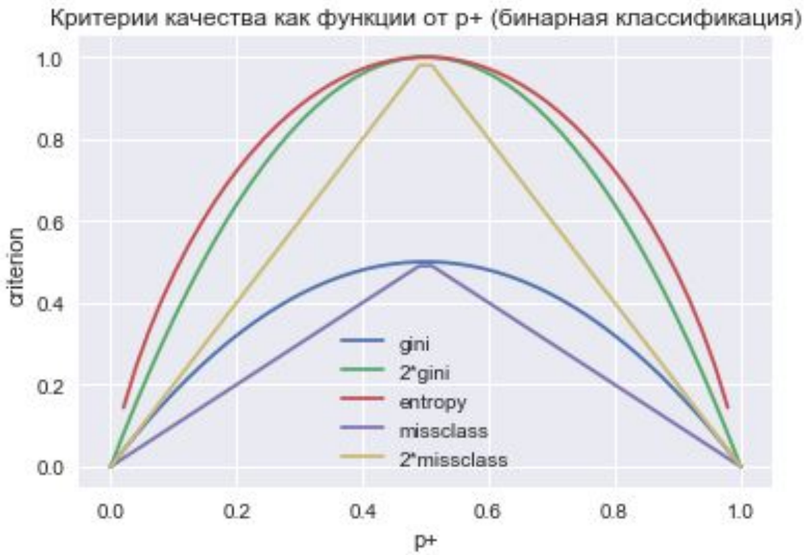


Деревья решений классификации

Статистический подход

$$H(X_m) = \min_{\sum_k c_k = 1} \frac{1}{|X_m|} \sum_{(x_i, y_i) \in X_m} \sum_{k=1}^{K} (c_k - \mathbb{I}[y_i = k])^2$$

$$H(X_m) = \sum_{k=1}^{K} p_k (1 - p_k)$$



Деревья решений регрессии

Теперь хочется в целом понять, насколько данное разбиение помогает нам уменьшить ошибку, для этого нужно ввести понятие "прирост информации" (information gain). Он считается, как

$$IG = MSE_{root} - (\frac{n_{left}}{n} MSE_{left} + \frac{n_{right}}{n} MSE_{right})$$

где left - это количество объектов в левой ветке, right - это количество объектов в правой ветке, а n - количество объектов в корневом узле.

https://colab.research.google.com/drive/1taHD-8sUH0YPUtL9jdxbCY23E5yNhW5m?usp=sharing

Обобщение

Шаги по поиску узла расщепления:

1. Рассчитываем стандартное отклонение целевой переменной.

2. Разделяем набор данных на различные атрибуты и вычисляем стандартное отклонение для каждой ветви (для целевой переменной и предиктора).

3. Вычитаем это значение из стандартного отклонения перед разделением.

4. Результатом является уменьшение стандартного отклонения.

5. В качестве узла разделения выбираем атрибут с наибольшим уменьшением стандартного отклонения.

6. Набор данных разделяем на основе значений выбранного атрибута.

7. Этот процесс выполняется рекурсивно.

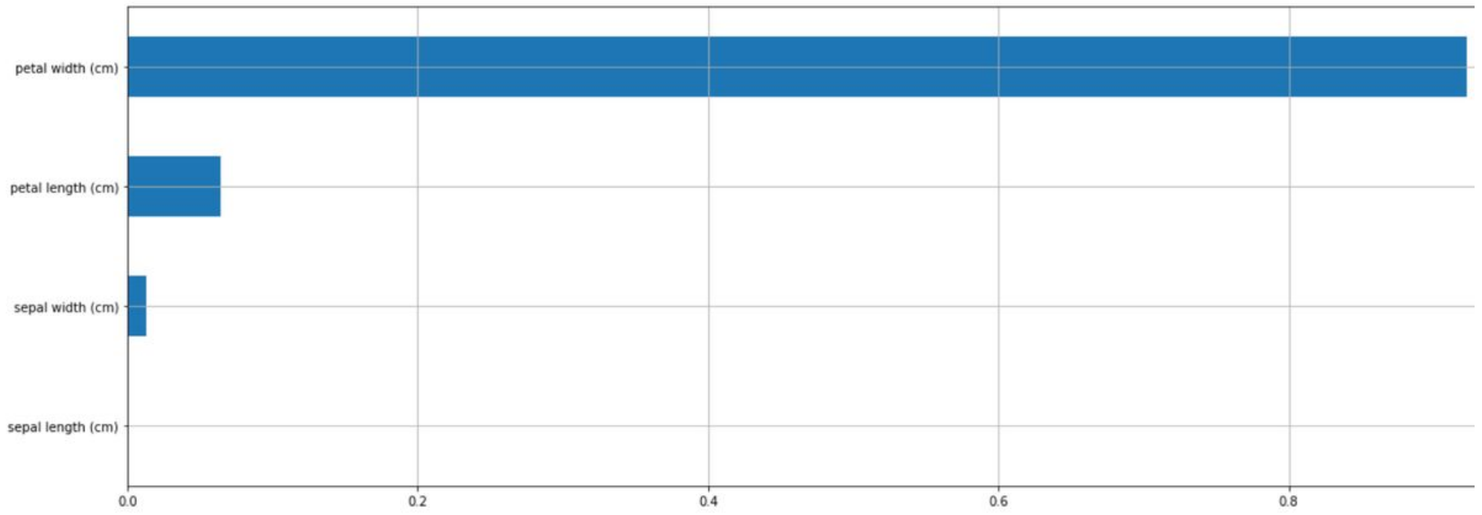
8. Чтобы избежать переобучения, используем коэффициент отклонения, который определяет, когда прекратить ветвление.

9. Наконец, каждой ветви присваивается среднее значение (в случае регрессии берется среднее значение).

Визуализация важности признаков

```
import pandas as pd
pd.Series(model.feature_importances_, index=model.feature_names_in_)\
            .sort_values(ascending=True).plot.barh(figsize=(20,7), grid=True)
```

```
<AxesSubplot:>
```



Критерий остановки алгоритма

Одним из возможных решений проблемы является принудительная остановка построения дерева до достижения переобучения.
Для этого были разработаны следующие подходы.

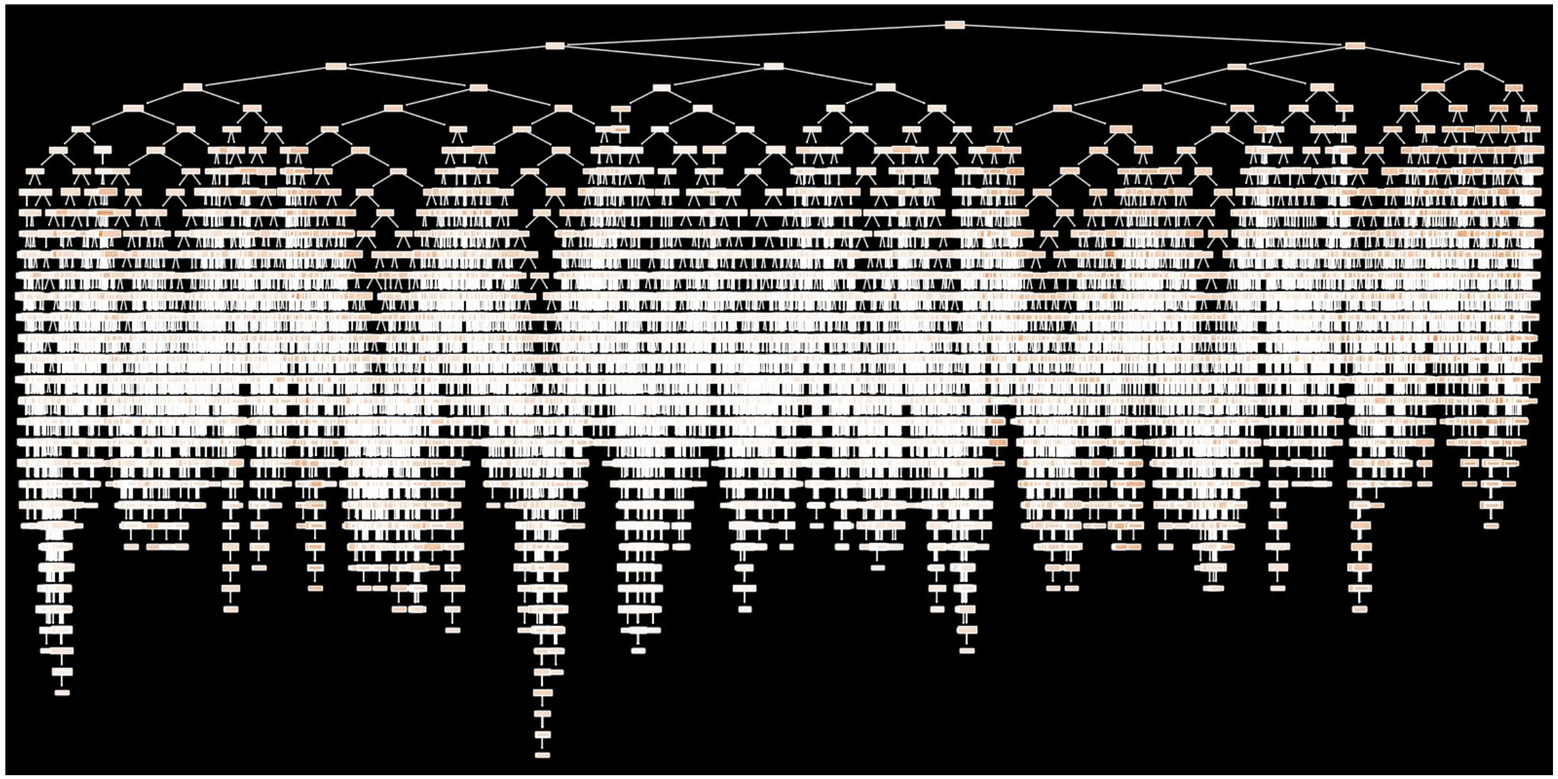
Первый подход – ранняя остановка, при которой алгоритм будет прекращать работу, как только будет достигнуто заданное значение некоторого критерия

Второй подход - ограничение глубины дерева — определение максимального количества разбиений в ветвях, при достижении которого обучение прекращается.

Третий подход - задание минимально допустимого количества примеров в узле — запрет алгоритму создавать узлы с количеством примеров меньше заданного (например, 5)

Критерий остановки алгоритма

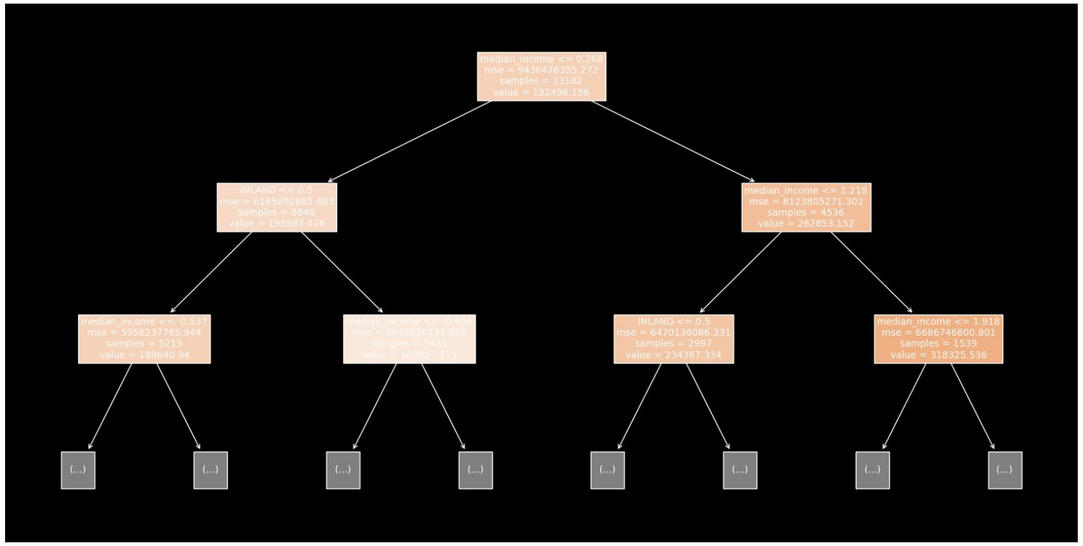
Как уже было отмечено, если не ограничить рост дерева, то получится сложное дерево с большим количеством узлов и листьев, которое будет сложно интерпретировать



Критерий остановки алгоритма

Другим способом является так называемое обрезание ветвей (pruning). Этот подход включает следующие шаги:

1. Построить полное дерево, где все листья содержат примеры одного класса.
2. Определить два показателя: относительную точность модели - отношение числа правильно распознанных примеров к общему числу примеров, и абсолютную ошибку - число неправильно классифицированных примеров.
3. Удалить из дерева листья и узлы, обрезание которых не приведет к значительному уменьшению точности модели или увеличению ошибки.



Ансамблевые методы

Ансамблевый метод — это метод машинного обучения, в котором несколько моделей обучаются для решения одной и той же проблемы, а затем объединяются для получения лучших результатов.

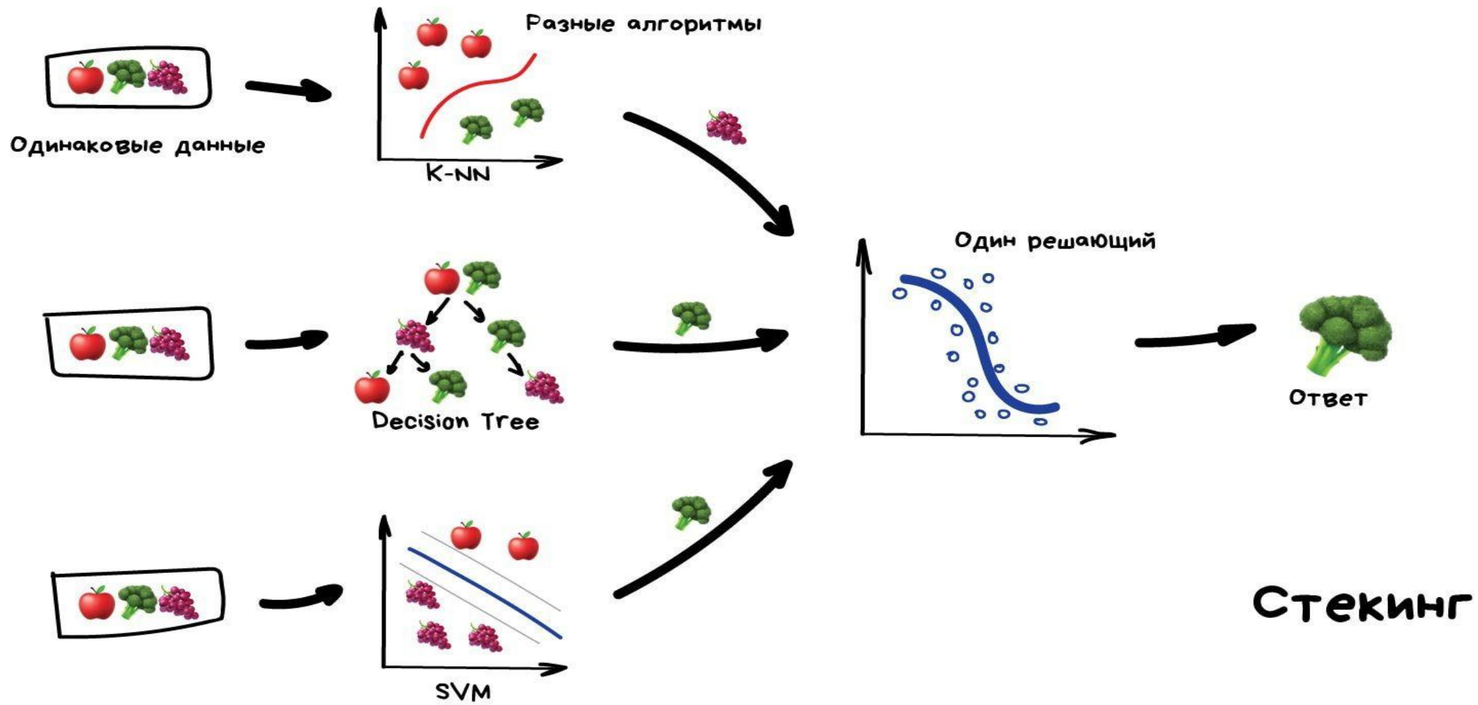
Стекинг (stacking) - комбинирование нескольких алгоритмов машинного обучения путем их последовательного применения, когда выход одной модели становится входом для следующей. Позволяет улучшить качество за счет комбинации сильных сторон разных алгоритмов.

Бэггинг (bagging) - комбинирование нескольких однотипных моделей (часто деревьев), обученных на разных подвыборках данных. Усреднение их предсказаний повышает качество и устойчивость.

Бустинг - поочередное обучение моделей с акцентом на объектах, где предыдущая модель ошибалась. Каждая модель дополняет предыдущую. Позволяет создать сильную композицию слабых моделей.

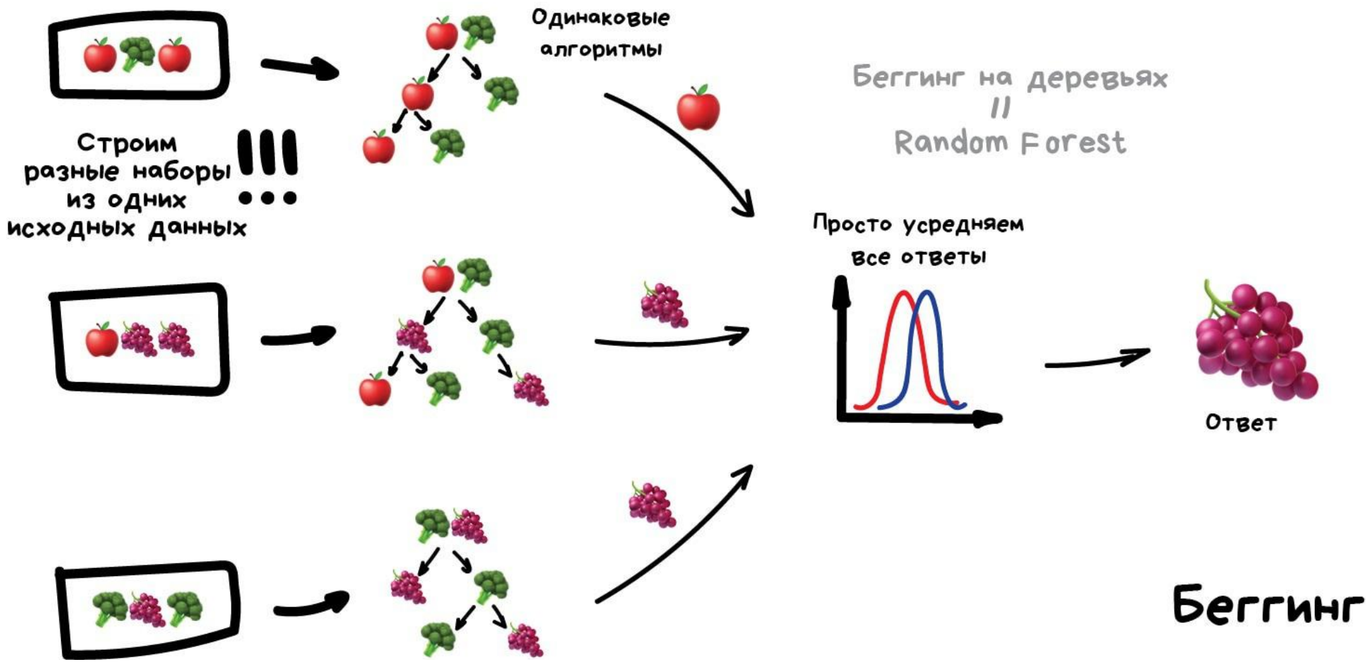
Стекинг

В этом методе все слабые прогнозаторы получают на вход обучающий набор данных, и каждый из них создает свой собственный прогноз.



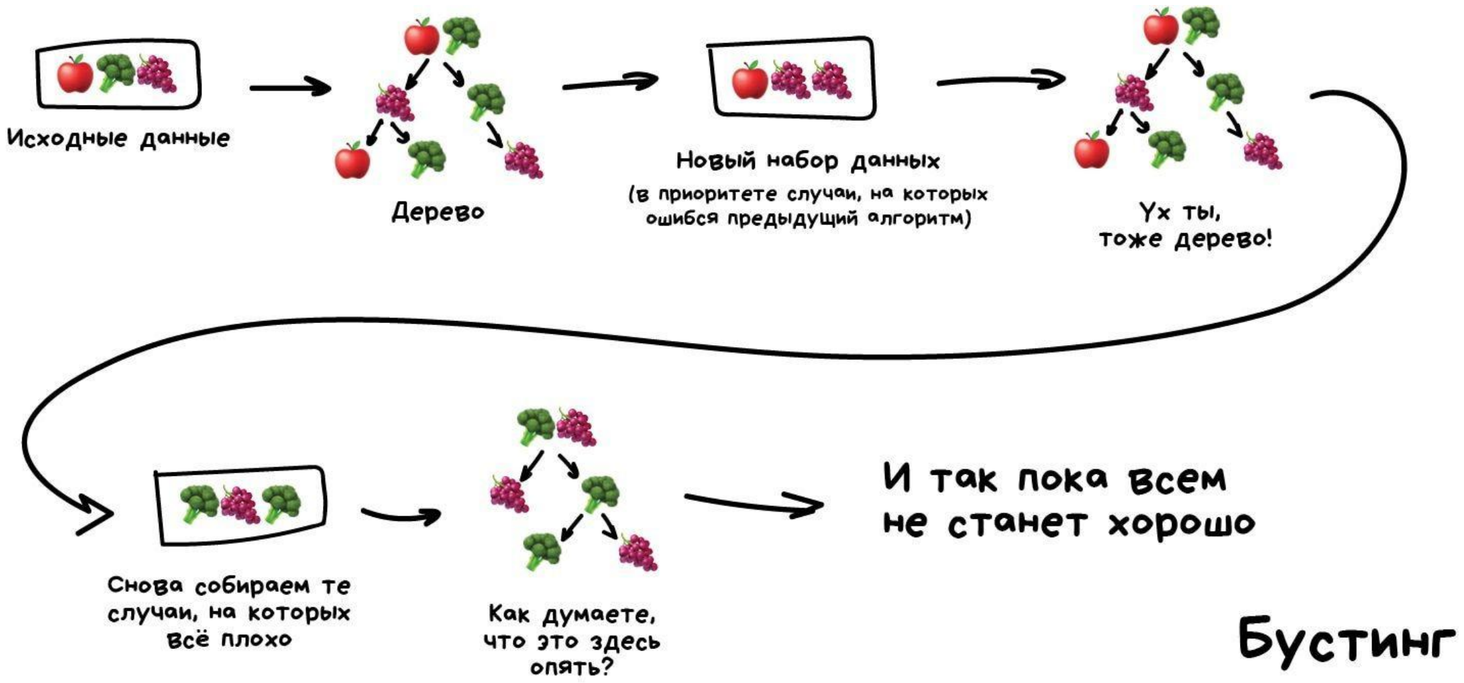
Бэггинг

Бэггинг - это метод обучения моделей, основная идея которого заключается в том, чтобы обучить несколько одинаковых моделей на различных подвыборках данных. Поскольку распределение выборки неизвестно, модели получаются разными.



Бустинг

Бустинг - метод, который отличается от метода бэггинга тем, что модели адаптируются к данным последовательно и исправляют ошибки предыдущих моделей.



Итоги

1. Деревья решений - это графическая модель, представляющая собой древовидную структуру.

2. Ансамбли деревьев - это комбинирование нескольких деревьев решений в одну модель для улучшения точности и стабильности предсказаний.

3. Практическая польза данных алгоритмов заключается в их способности моделировать сложные отношения между признаками и целевой переменной.

4. Решающие деревья и ансамбли имеют множество преимуществ, включая интуитивную понятность и легкость интерпретации. Они способны обрабатывать данные с пропущенными значениями и выбросами, а также работать с различными типами переменных. Кроме того, они могут автоматически определять важность признаков.

5. При использовании алгоритмов необходимо учитывать несколько недостатков: возможное переобучение, сложности с оптимальной структурой дерева и недостаточную устойчивость к изменениям в данных.

Спасибо за внимание