# Введение в анализ данных

## Вопросы к контрольной работе 21.04.2021

vk: vk.com/uselessofflane, tg: @fmakhnach

### 1. Что такое объекты и признаки в машинном обучении? Для чего нужен функционал ошибки? Что такое алгоритм (модель)?

**Объект (sample)** – то, для чего делается прогноз, "входные данные".

**Признак** – [числовая] характеристика объекта, которая используется в моделях при обучении и предсказании.

**Функционал ошибки** – функция, отражающая меру ошибочности ответов алгоритма. Примеры: MSE (Mean Squared Error), MAE (Mean Absolute Error).

**Алгоритм, модель** – функция, которая на вход принимает объект (в виде множества его признаков), а на выходе возвращает предсказание значения целевой переменной для этого объекта.

### 2. Чем задача классификации отличается от задачи регрессии? Приведите примеры задач классификации и регрессии.

Ответы на задачу регрессии вещественные: $\mathbb{Y} \subseteq \mathbb{R}$.
Ответы на задачу классификации целочисленные, число различных возможных ответов конечное. Зачастую классификация решает задачу отнесения объекта к какой-либо группе.

**Пример задачи классификации:**
Определение котиков на изображении (бинарная классификация: «котик»-«не котик»).

**Пример задачи регрессии:**
Определение ожидаемого дохода предприятия в следующий месяц.

### 3. Что такое вещественные (числовые), бинарные, категориальные признаки? Приведите примеры.

**Вещественные (числовые) признаки** – признаки, представляющие из себя вещественные числа, над которыми можно выполнять математические операции. Примеры: средние расходы в месяц, число проданных товаров, средняя температура.

**Бинарные признаки** – признаки, принимающие всего два значения (зачастую – «истина» или «ложь»). Например: болен ли пациент раком, шел ли в этот день дождь, открыт ли магазин.

**Категориальные признаки** – признаки, значения которых ограничены конечным множеством. Например, город проживания, марка автомобиля, исполнитель песни.

## 4. В чём заключается обобщающая способность алгоритма машинного обучения? К чему приводит её отсутствие? Что такое переобучение?

Говорят, что алгоритм машинного обучения **обладает обобщающей способностью**, если ошибка этого алгоритма на тестовой выборке невелика и сопоставима с ошибкой на обучающей выборке.
Отсутствие обобщающей способности приводит к **переобучению** алгоритма, т.е. ошибка на обучающей выборке оказывается малой, однако ошибка алгоритма на новых данных становится существенно выше. Т.е. алгоритм «подстраивается» под обучающую выборку.

## 5. Что такое гиперпараметр? Чем гиперпараметры отличаются от обычных параметров алгоритмов? Приведите примеры red параметров и гиперпараметров в линейных моделях.

**Гиперпараметр** – параметр модели, выбор которого лишь на основании обучающей выборке приведёт к потере качества. Для подбора таких параметров необходимо использовать дополнительные данные. В методе $kNN$ количество соседей $k$ является гиперпараметром.

Параметры нужны, чтобы подогнать модель под данные.

Гиперпараметры, чтобы контроллировать сложность модели. Позволяют бороться с переобучением, подбираются кросс-валидацией.

## 6. Что такое отложенная выборка? Что такое кросс-валидация (скользящий контроль)? Как ими пользоваться для выбора гиперпараметров?

**Отложенная выборка** – метод проверки переобученности модели путём разбиения обучающей выборки на 2 части. Первая часть используется для обучения, вторая – для проверки качества обученной модели. Таким образом проверяется как ведёт себя модель на новых данных.

**Кросс-валидация** – метод проверки переобученности модели. Обучающая выборка разбивается на $n$ частей, после чего модель обучают $n$ раз, используя одну из $n$ частей в качестве тестовой выборки, а остальные объекты – в качестве обучающей. Затем для оценки всей модели используют оценки качества модели в каждом эксперименте (можно взять среднее / наибольшее / наименьшее / что-то другое).

Эти методы можно использовать для подбора гиперпараметров: рассчитывать качество модели при различных значениях гиперпараметра и выбирать то значение, которое приводит к наилучшей метрике качества.

## 7. Как метод k ближайших соседей определяет класс для нового объекта?

Модель хранит все объекты, ответы на которые уже известны. При рассчёте предсказания на новый объект рассчитывается «расстояние» (метрика) от этого объекта до каждого объекта из обучающей выборки, затем эти объекты упорядочиваются по возрастанию расстояния,

после чего выбирается $k$ «ближайших» объектов. Наиболее часто встречающийся класс среди выбранных объектов и выбирается в качестве предсказания.

## 8. Опишите метод k ближайших соседей с парзеновским окном. Какие в нём есть параметры?

$kNN$ с парзеновским окном – дополнение стандартного $kNN$, учитывающее расстояние до объектов (чем ближе объект, тем больший вклад он вносит). Предсказание вычисляется по следующей формуле:

$$a(x) = arg\max_{y \in \mathbb{Y}} \sum_{i=1}^{k} w_i[y_{(i)} = y], \;\; w_i = K\left(\frac{\rho(x, x_{(i)})}{h}\right)$$

где $K$ – ядро (некоторая функция, напр $\frac{1}{x+1}$), $h$ – «ширина окна»: регулирует, насколько нам важны дальние объекты.

## 9. Запишите формулу метода kNN для регрессии.

kNN для регрессии выдаёт среднее значение целевой переменной среди $k$ ближайших объектов:

$$a(x) = \frac{1}{k} \sum_{i=1}^{k} y_{(i)}$$

## 10. Что такое градиент? Какое его свойство используется в машинном обучении?

**Градиент** – функция, сопоставляющая каждой точке $(x_1, ..., x_n)$ функции $f(x_1, ..., x_n)$ вектор, направление которого указывает в наибольшее **возрастание** функции в точке. Этот вектор, компоненты которого равны частным производным функции $f(x_1, ..., x_n)$ по всем её аргументам.

**Антиградиент** – вектор-градиент, направленый противоположную сторону (градиент со знаком «минус»). Указывает в сторону наибольшего **убывания** функции в точке.

**Градиентный спуск** – метод нахождения минимума функции в случае, когда аналитическое решение невозможно / слишком затратное. Использует **антиградиент** для нахождения ближайшей точки *локального* минимума. Градиентный спуск используется для минимизации функции потерь при поиске коэффициентов в линейной регрессии.

## 11. Опишите алгоритм градиентного спуска.

Некоторым образом (например, случайно) выбирается начальный набор коэффициентов (начальное приближение) линейной регрессии. Затем выполняются некоторое количество итераций, на каждой из которых

1. Рассчитывается градиент функции потерь от текущих коэффициентов. Если он достаточно близок к нулю (достигли минимума), вычисление завершается. Альтернативно, можно остановить вычисление, когда разница между значениями, полученными на этом и предыдущем шагах ($||w^t|| - ||w^{t-1}||$) мала;

2. К текущим коэффициентам прибавляется антиградиент, умноженный на заданный параметр $\eta$ (длина шага) – делаем «шаг вниз» по функции.

$$w^t = w^{t-1} - \eta \nabla Q(w^{t-1})$$

Полученные на последней итерации коэффициенты применяем для предсказаний модели.

## 12. Что такое стохастический градиентный спуск? В чём его отличия от обычного градиентного спуска? Какие у него плюсы и минусы?

**Стохастический градиентный спуск** – оптимизация градиентного спуска, заключающаяся в том, что вместо использования всех объектов обучающей выборки для вычисления градиента на каждом шаге используется лишь часть объектов.

Например, если имеется обучающая выборка на 100000 объектов, для вычисления точного значения градиента потребуется вычислить градиент функционала ошибки для каждого объекта. Чтобы этого избежать, на каждом шаге выбирается случайная подвыборка размера (например) 10, по которой и строится градиент. Это позволяет сократить затраты ресурсов для вычисления шага, однако шаг становится менее точным. Тем не менее, достаточное число таких «не совсем точных» шагов всё равно, как правило, приводят к довольно точному результату.

Стохастический = случайный (зато слово пафосное).

## 13. Как обучается линейная регрессия?

Линейная регрессия обучается путём минимизации функционала ошибки. То есть стараемся «подобрать» такие коэффициенты, чтобы функционал ошибки был минимальным. Этого можно достигнуть, например, с помощью аналитической формулы (сложно, не всегда применимо) или с помощью градиентного спуска (менее затратно, но результат может быть хуже).

P.S. Если между признаками существует линейная зависимость, то у задачи нахождения минимума будет несколько решений. В этом случае аналитическое решение не сработает, а градиентный спуск даст какой-то результат (один из минимумов).

## 14. Что такое регуляризация? Как она помогает бороться с переобучением?

Существует эмпирическое наблюдение – большие коэффициенты линейной модели свидетельствуют о **переобучении**. Вывод: большие коэффициенты – плохо. Следующий вывод: можем препятствовать возникновению больших весов и таким образом улучшать качество модели.

Будем штрафовать за большие веса с помощью **регуляризатора** – функции от вектора весов, которая возрастает при возрастании модулей значений весов. Её значение добавляется к значению функционала ошибки, например

$$\frac{1}{k}\sum_{i=1}^{k}(\langle w, x_i\rangle - y_i)^2 + \lambda||w||^2 \to min$$

$\lambda$ – коэффициент регуляризации (чем больше, тем важнее регуляризация).

P.S. $w_0$ не входит в регуляризацию (!) – не хотим штрафовать за свободный коэффициент.

## 15. Чем L1-регуляризация отличается от L2-регуляризации?

$L_1$-норма: $||z||_1 = \sum_{j=1}^{d} |z_j|$

$L_2$-норма: $||z||_2 = \sqrt{\sum_{j=1}^{d} z_j^2}$

$L_1$ регуляризация способствует появлению нулевых весов, $L_2$ регуляризация препятствует появлению больших по модулю весов. Помимо этого, у $L_1$ из-за модуля есть проблемы с дифференциированием.

## 16. Что такое масштабирование признаков? Как его проводить? Зачем это нужно?

**Масштабирование признаков** – процедура приведения признаков к одному масштабу.
Пример: StandardScaler

$$x_i^j := \frac{x_i^j - \mu_j}{\sigma_j}$$

$$\mu_j = \frac{1}{k}\sum_{i=1}^{k} x_i^j, \quad \sigma_j = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(x_i^j - \mu_j)^2}$$

После проведения масштабирования признаков можно делать выводы о важности признаков по весам.

P.S. Последнее возможно только в случае **не**переобученности модели, а значит следует также применить регуляризацию.

## 17. Чем функционал MSE отличается от MAE? Что такое функция потерь Хубера и для чего она нужна?

**MSE** – Mean Squared Error – среднеквадратичная ошибка, вычисляется по формуле

$$Q(a, X) = \frac{1}{l}\sum_{i=1}^{l}(a(x_i) - y_i)^2$$

**MAE** – Mean Absolute Error – средняя абсолютная ошибка, вычисляется по формуле

$$Q(a, X) = \frac{1}{l}\sum_{i=1}^{l}|a(x_i) - y_i|$$

MSE довольно сильно меняется из-за выбросов, в отличие от MAE. В свою очередь MAE не дифференциируема.

**Функция потерь Хубера** – функционал ошибки, являющийся комбинацией MSE и MAE. На достаточно близких к нулю значениях она ведёт себя как MSE, а на остальных – как MAE. Это позволяет избавиться проблемы с выбросами у MSE и проблемой с недифференциируемостью MAE. Формула:

$$L(y, a) = \begin{cases} \frac{1}{2}(y-a)^2, & |y-a| < \delta, \\ \delta\left(|y-a| - \frac{1}{2}\delta\right), & |y-a| \geq \delta \end{cases}$$

## 18. Как выглядит модель линейной классификации в случае двух классов?

Прогноз принадлежности объекта $x$ некоторому классу строится следующим образом:

$$a(x) = \text{sign}\left(w_0 + \sum_{j=1}^{d} w_j x_j\right) = \text{sign}\langle w, x\rangle$$

Здесь значения 1 и -1 будут как-то соответствовать двум классам.
Вообще, $\langle w, x\rangle = 0$ задаёт гиперплоскость. Можем интерпретировать процесс обучения как процесс подбора гиперплоскости, которая наилучшим образом будет разделять пространство объектов на 2 части по принадлежности классу.

## 19. Что такое отступ? Для чего он нужен?

**Отступ** (margin) – величина модели, характеризующая для каждого объекта

1. права ли модель в своём предсказании на данном объекте;
2. насколько она «уверена» в своём ответе.

Вычисляется по формуле

$$M_i = y_i\langle w, x_i\rangle$$

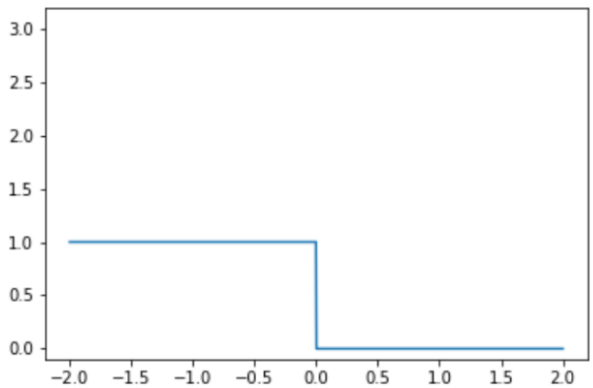
$M_i > 0 \Rightarrow$ классификатор даёт верный ответ,
$M_i < 0 \Rightarrow$ классификатор ошибается,
$|M_i|$ характеризует степень уверенности модели (чем больше величина, тем более уверена).
С помощью отступа можно определять выбросы – искать объекты, у которых отступ отрицательный и большой по модулю.

## 20. Как обучаются линейные классификаторы (общая схема с верхними оценками)?

Функционал ошибки для линейного классификатора выглядит так:

$$Q(w, X) = \frac{1}{k}\sum_{i=1}^{k}[\underbrace{y_i\langle w, x_i\rangle}_{M_i} < 0]$$

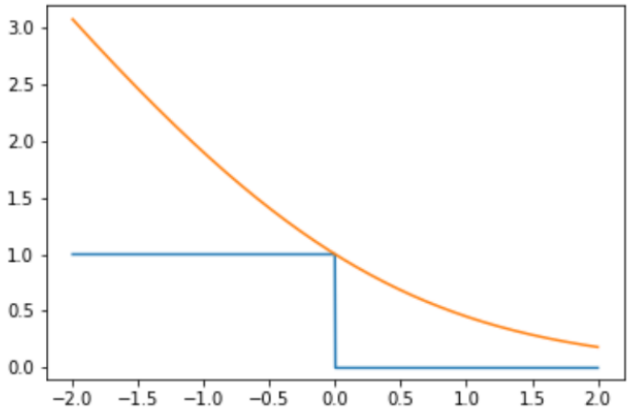
График такой:



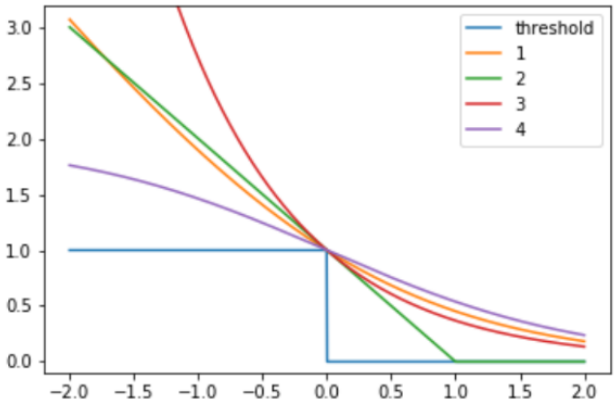
Данный функционал плох тем, что

1. не дифференциируем;
2. не зависит от отступа – степени уверенности модели.

Для избавления от этих недостатков можем оценить функционал сверху дифференциируемой функцией:




Существуют разные варианты верхних оценок функции, обладающие разными свойствами:




1. $\tilde{L}(M) = \log(1 + e^{-M})$ – логистическая;
2. $\tilde{L}(M) = \max(0, 1 - M)$ – кусочно-линейная;
3. $\tilde{L}(M) = e^{-M}$ – экспоненциальная;
4. $\tilde{L}(M) = \frac{2}{1+e^{M}}$ – сигмоидная;

Выбрав подходящую верхнюю оценку, обучаем модель минимизацией этого функционала. Например, с помощью градиентного спуска. Можем применить регуляризацию.

Помимо этого, можем подбирать **порог** $t$ для нахождения оптимального значения:

$$a(x) = \text{sign}(\langle w, x\rangle - t) = 2[\langle w, x\rangle > t] - 1$$

Высокий порог $\Rightarrow$ мало объектов относим к $+1$ $\Rightarrow$ точность выше, полнота ниже.
Низкий порог $\Rightarrow$ много объектов относим к $+1$ $\Rightarrow$ точность ниже, полнота выше.

## 21. Для чего может понадобиться оценивать вероятности классов?

Кредитный скоринг (с какой вероятностью клиент вернёт кредит), предсказание вероятности клика на баннерную рекламу, медицинская диагностика, поисковое ранжирование.

## 22. Как обучается логистическая регрессия? Запишите функционал и объясните, откуда он берётся.

Функция потерь называется **log-loss**:

$$L(y, z) = -[y = 1] \log z - [y = -1] \log (1 - z)$$

Функционал потерь получается следующий:

$$\sum_{i=1}^{k} L(y, \sigma(\langle w, x_i \rangle))$$

где $\sigma(z) = \frac{1}{1+\exp(-z)}$. Если немного пошаманить, эта функционал можно привести к виду:

$$\sum_{i=1}^{k} \log (1 + \exp (-y_i \langle w, x_i \rangle))$$

Логистическая регрессия обучается минимизацией вот этого функционала.

## 23. Как в логистической регрессии строится прогноз для нового объекта?

Прогноз (он же – вероятность принадлежности $x$ положительному классу):

$$b(x) = \sigma(\langle w, x \rangle)$$

Здесь $w$ – коэффициенты, найденные в результате обучения, $\sigma(z) = \frac{1}{1+e^{-z}}$

## 24. Что такое метод опорных векторов? Опишите его основную идею.

**Проблема**: при обучении линейного классификатора можем получить несколько различных решений (гиперплоскостей) – как выбрать наилучшее?

**Решение**: будем максимизировать отступ классификатора – расстояние от гиперплоскости классификатора до ближайшего объекта. Т.е. хотим провести прямую максимально далеко от всех объектов. При этом не стоит забывать о минимизации числа ошибок.

Ищем минимальное расстояние так:

$$\rho_{\min} = \min_{i=1\ldots k} \frac{|\langle w, x_i \rangle + w_0|}{||w||}$$

Наблюдение: можем поделить все веса на $\min_{i=1\ldots k} |\langle w, x_i \rangle + w_0|$, тогда $\rho_{\min} = \frac{1}{||w||}$.
Итого: максимизируем $\frac{1}{||w||}$ = минимизируем $||w||^2$ при условии $y_i(\langle w, x_i \rangle + w_0) \geq 1$

## 25. Как устроены метрики accuracy, precision, recall? Что такое F-мера? Чем она лучше арифметического среднего точности и полноты?

Введём следующие обозначения:
ALL – число всех объектов;
TP («True Positive») – число «истинных» объектов, на которых модель выдала +1;
TN («True Negative») – число «ложных» объектов, на которых модель выдала -1;
FP («False Positive») – число «ложных» объектов, на которых модель выдала +1;
FN («False Negative») – число «истинных» объектов, на которых модель выдала -1;
Т.е. первая буква – верное ли предсказание, вторая – что было предсказано.

Перейдём к метрикам.

**Accuracy** – доля правильных ответов алгоритма

$$accuracy = \frac{TP + TN}{ALL}$$

Почти не используется, т.к. бесполезна в задачах с неравными классами.

**Precision** (точность) – доля верных ответов среди положительных ответов.

$$precision = \frac{TP}{TP + FP}$$

**Recall** (полнота) – доля найденных положительных ответов среди всех положительных объектов.

$$precision = \frac{TP}{TP + FN}$$

**F-мера** – объединение precision и recall в одну метрику. Равна среднему гармоническому между precision и recall:

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{\beta^2 \cdot precision + recall}$$

$\beta$ – вес точности в метрике.
F-мера, в отличие от среднего арифметического, близка к нулю, если один из аргументов близок к нулю. Т.е. эта метрика сильно штрафует за хотя бы 1 низкую метрику из двух (precision и recall), а мы как раз и хотим максимизировать обе одновременно.

## 26. Для чего нужны ROC- и PR-кривые? Как они строятся? Что такое AUC-ROC и AUC-PRC?

**Проблема**: качество линейного классификатора зависит от порога, однако порог может меняться со временем в зависимости от наших задач в моменте. Хотим оценить обобщённое качество модели для всех порогов.

**PR-кривая** – кривая, отображающая значения precision и recall в зависимости от порога. Для всевозможных порогов, которые дают разные разбиения, на графике с осями precision-recall отмечаются значения precision и recall для данного порога.

**ROC-кривая** – кривая, отображающая false positive rate и true positive rate в зависимости от порога. ROC = Receiver Operating Characteristic.

$$FPR = \frac{FP}{FP + TN}, \quad TPR = \frac{TP}{TP + FN}$$

FPR = «ошибочно отнесённые к +1» / «все объекты -1»,
TPR = «корректно отнесённые к +1» / «все объекты +1» = полнота.

**AUC-PR** – площадь под PR-кривой. Тем больше, чем лучше качество модели.

**AUC-ROC** – площадь под ROC-кривой. Тем больше, чем лучше качество модели. Для идеального алгоритма AUC-ROC = 1, для худшего – AUC-ROC $\approx 0.5$

P.S. AUC = Area Under Curve

## 27. Как можно свести задачу многоклассовой классификации к серии задач бинарной классификации?

Для каждого класса $C$ проводим бинарную классификацию
«Объект принадлежит классу $C$»-«Объект НЕ принадлежит классу $C$».

## 28. Что такое решающее дерево? Как оно строит прогноз для объекта? Как обучаются решающие деревья в задачах классификации и регрессии (и что такое критерии информативности)?

**Решающее дерево** – бинарное дерево поиска, у которого внутренние вершины (те, что с листьями) содержат в себе предикаты типа $[x_j < t]$ («j-ый признак объекта $x$ меньше, чем $t$»), а листья содержат прогнозы $c \in \mathbb{Y}$.

**Как строится прогноз**: стартуем с корня дерева, а дальше

1. Если текущая вершина – лист, то берём из неё прогноз и возвращаем;
2. Если текущая вершина – не лист, то применяем предикат к объекту;
3. Если предикат выдал положительный ответ, идём «влево» по дереву, иначе – «вправо» (или наоборот).

**Построение:**

Если вкратце – хотим минимизировать **критерий информативности** в вершинах. **Критерий информативности** – функция, отражающая качество распределения объектов среди множества $R$ (чем меньше, тем однороднее – то, чего и добиваемся). Примеры критериев информативности: энтропия, критерий Джини.

**Энтропия, impurity** (в данном случае) – величина, которая тем больше, чем ближе к равномерному распределению распределение объектов, попадающих в вершину (напр. по 1 объекту от каждого класса). Соответственно, минимальна она будет тогда, когда в вершину попадают объекты лишь одного класса – именно то, чего и добиваемся. Энтропия в вершине обозначается как $H(R)$. Если $p_1, ..., p_k$ – доля объектов каждого класса в вершине, то энтропия в вершине есть

$$H(p_1, ..., p_k) = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

Дальше максимизируем такую штуку:

$$Q(R, j, t) = H(R) - \frac{|R_l|}{|R|} H(R_l) - \frac{|R_r|}{|R|} H(R_r) \to \max_{j,t}$$

или минимизируем такую:

$$Q(R, j, t) = \frac{|R_l|}{|R|} H(R_l) + \frac{|R_r|}{|R|} H(R_r) \to \min_{j,t}$$

$H$ – критерий информативности, $R$ – объекты, попавшие в родительскую вершину, $R_l$ – объекты, попавшие в левое поддерево после разбиения, $R_r$ – объекты, попавшие в правое поддерево после разбиения.

Сами варианты подбора предикатов находим путём жадного разбиения: по каждому признаку пытаемся разделить объекты на 2 группы всевозможными (качественно различными) способами. Ну и среди этих всех вариантов выбираем тот, у которого критерий информативности наименьший. И так на каждом шаге.

## 29. Какие вы знаете критерии останова и способы выбора значений в листьях? Какие гиперпараметры имеются у деревьев?

**Критерии останова:**

1. Ограничение максимальной глубины дерева;
2. Ограничение минимального числа объектов в листе;
3. Ограничение максимального количества листьев в дереве;
4. Останов в случае, если все объекты в листе относятся к одному классу.
5. Требование, что функционал качества при дроблении улучшался как минимум на $s$ процентов.

**Способы выбора значений в листьях:**

1. Для регрессии: среднее значение объектов обучающей выборки, попавших в этот лист;
2. Для классификации: преобладающий класс среди объектов обучающей выборки, попавших в этот лист;

**Гиперпараметры (???):**

1. Вид предикатов в вершинах;
2. Критерий останова;
3. Функционал качества.