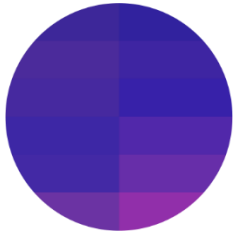
ШИФТ
Экосистема

Отбор признаков и синтетические данные

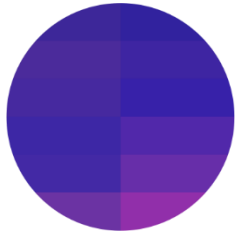
Нечман Дмитрий

Дата-аналитик

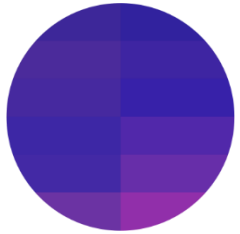
План занятия



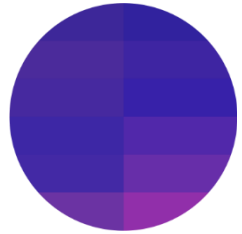
Отбор признаков, зачем он нужен и как его делать



Синтетические данные, когда и как их генерировать



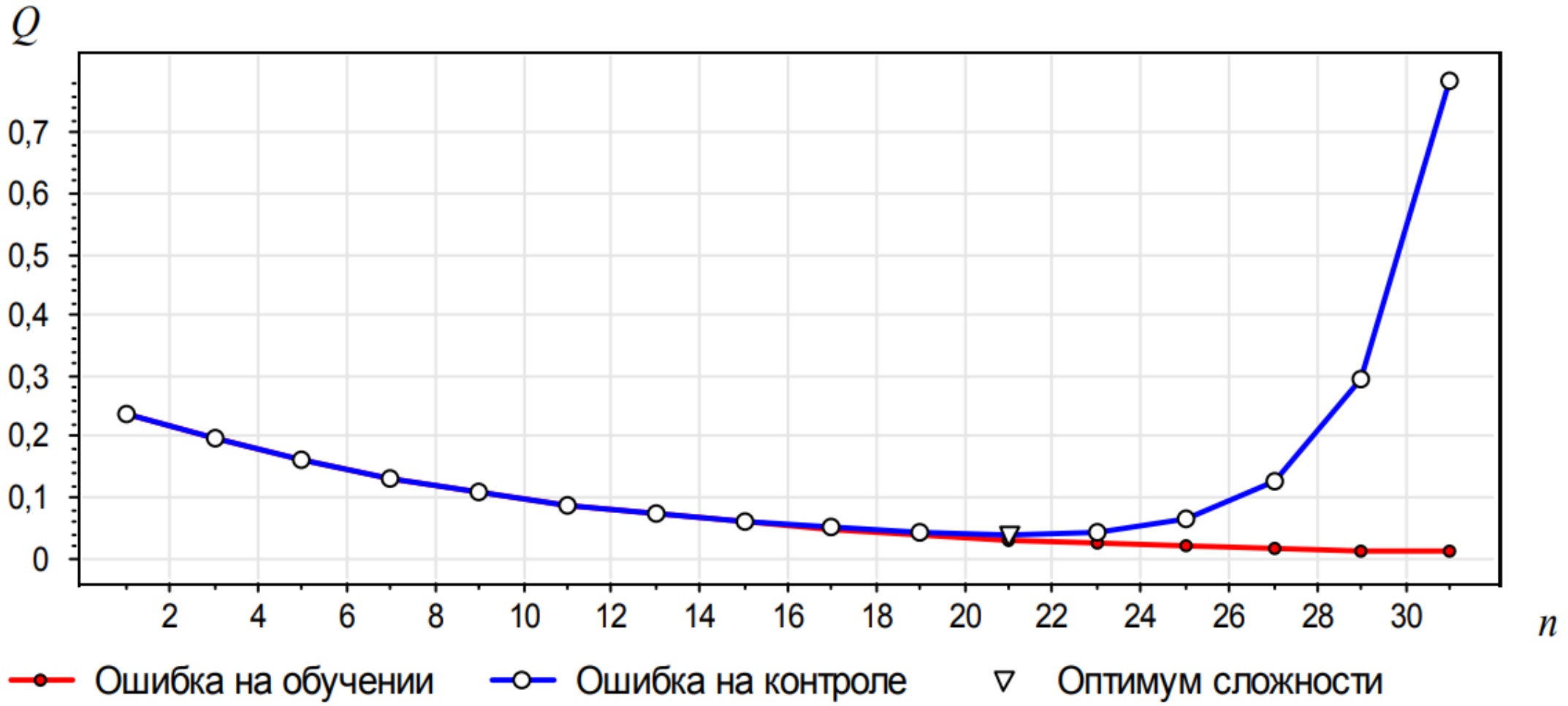
Ноутбук



Boruta

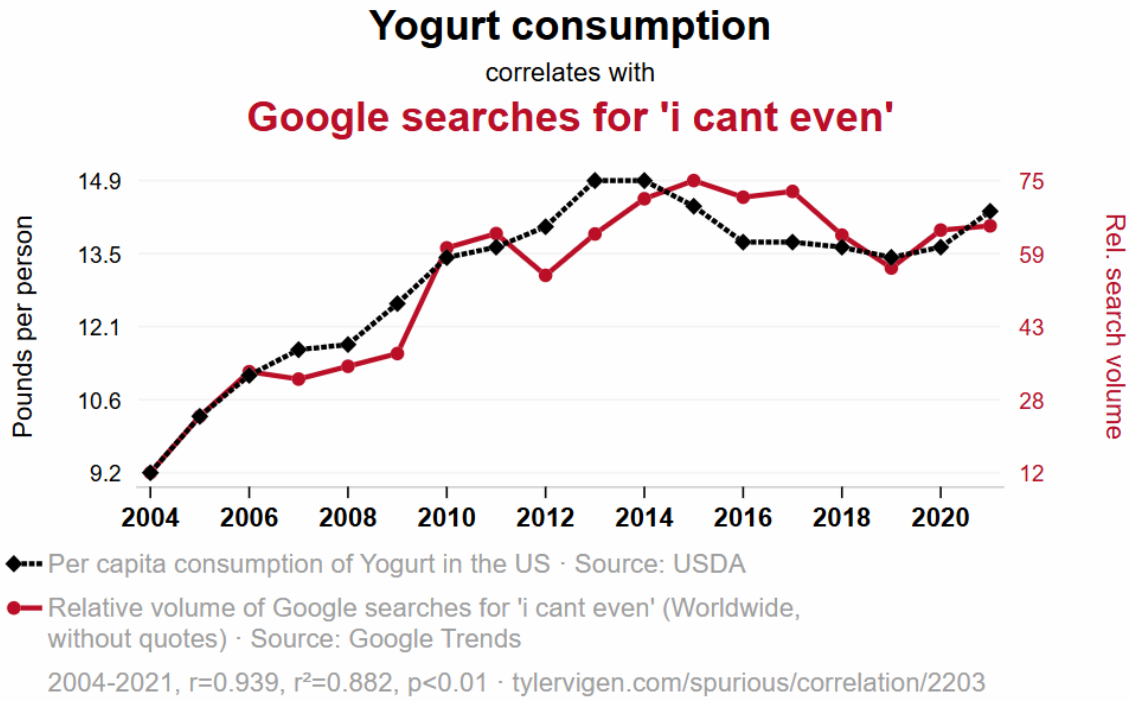
Зачем отбирать признаки?

Б о л ь ш е !=л у ч ш е



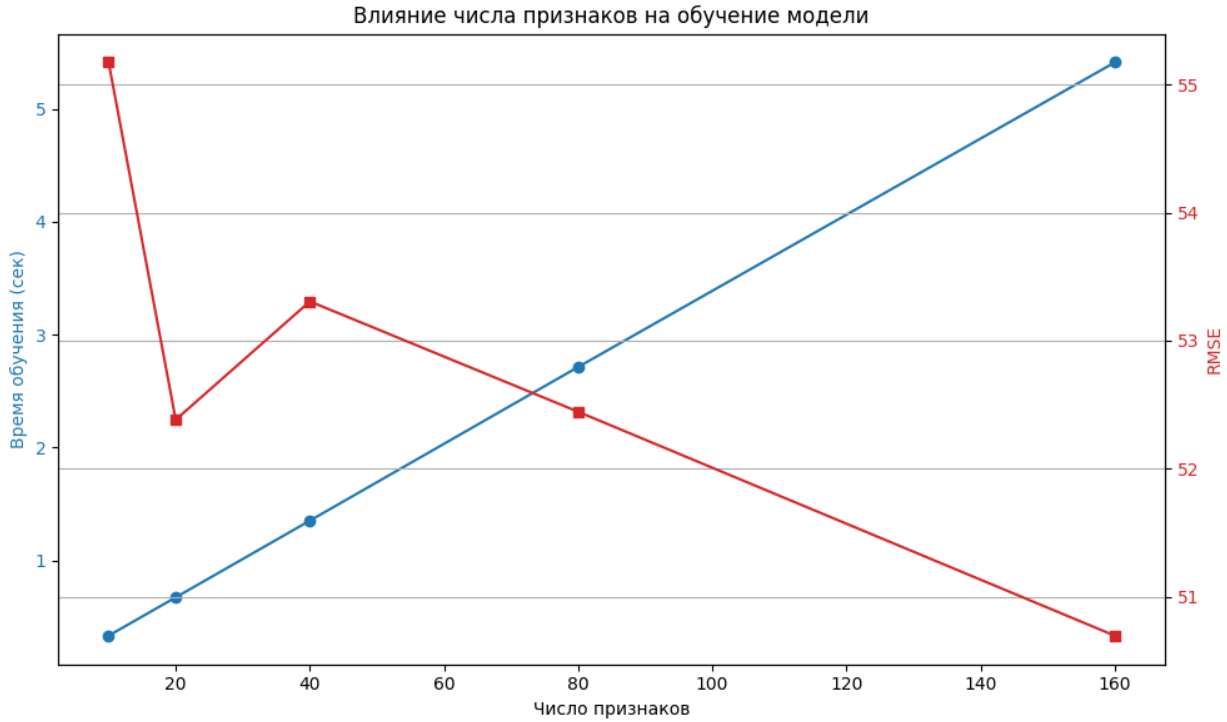
Нерелевантные и избыточные признаки

- Некоторые признаки просто не связаны с целевой переменной.
  → Такие признаки лучше исключить, чтобы избежать переобучения и ложных закономерностей (Spurious relationship).
- Другие признаки могут дублировать информацию и вызвать мультиколлинеарность, что критично для линейных моделей



Время обучения и сложность внедрения растет

- Иногда выгоднее взять меньше признаков, но обучить и применить модель быстрее.
- Чем больше признаков, тем больше рисков на проде



Проклятие размерности

- В пространстве с большим числом измерений данные становятся "разреженными" —
- каждая точка далеко от остальных.
- Модель не может эффективно находить закономерности:
- расстояния между объектами становятся неинформативны,
- плотность данных резко падает,
- всё пространство становится "пустым".
- Страдают алгоритмы, работающие с расстоянием, KNN-например.

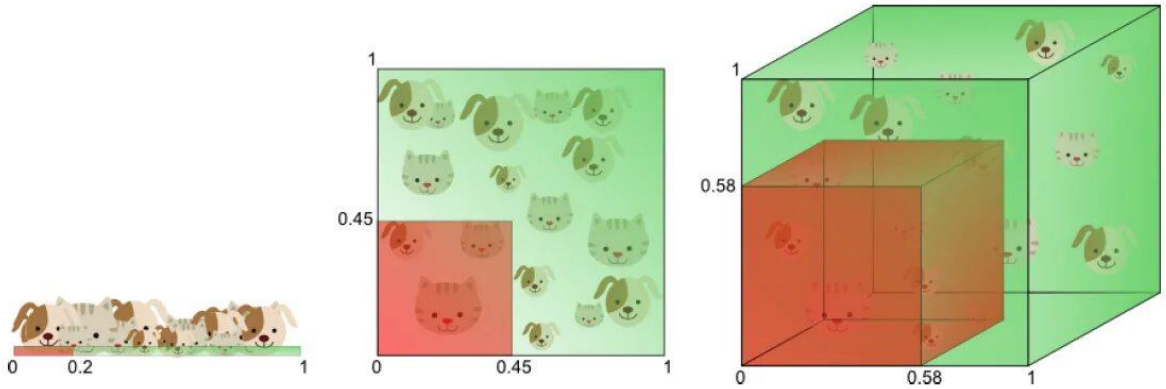
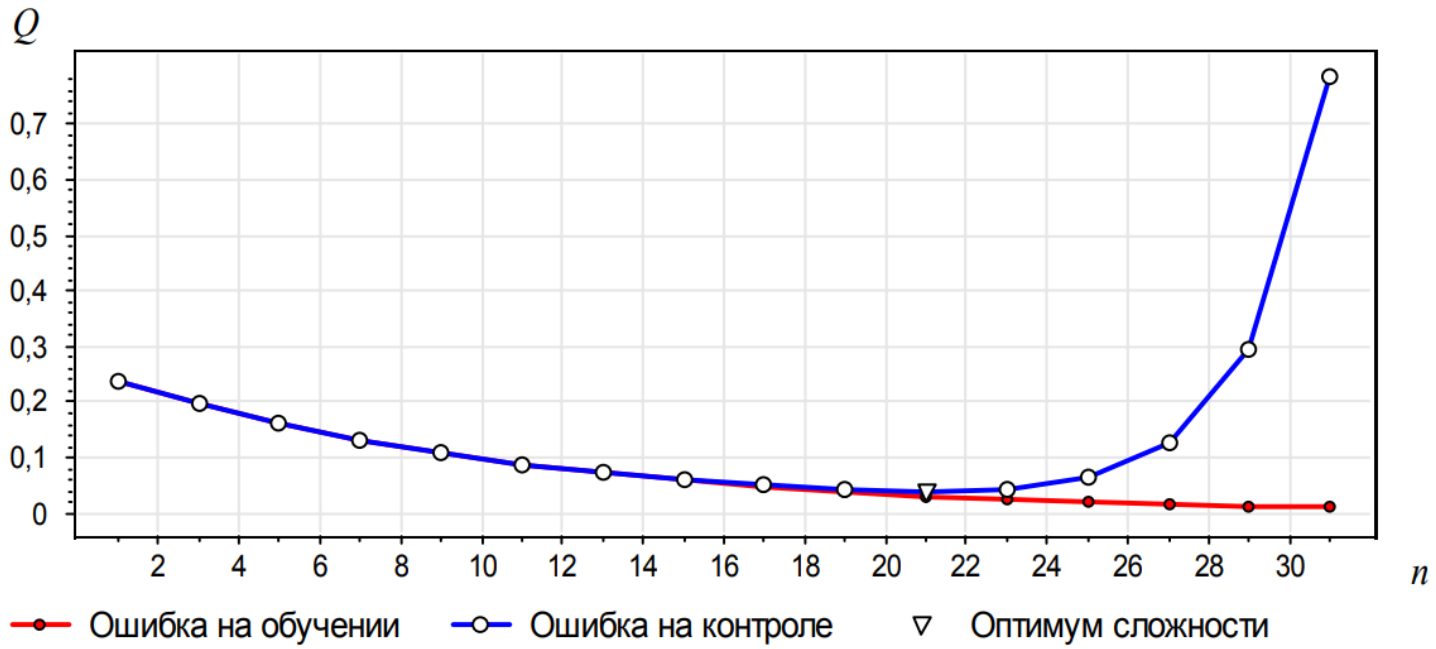


Figure 8. The amount of training data needed to cover 20% of the feature range grows exponentially with the number of dimensions.

Формальный вид задачи

$F = \{f_j : X \to D_j : j = 1, \ldots, n\}$ – множество признаков

μ_j – метод обучения использующий только признаки $J \subseteq F$



Подходы к отбору признаков

Без использования целевой переменной

Нулевая или почти нулевой дисперсия

- Если признак почти всегда принимает одно и то же значение, он ничего не сообщает модели.

	f1	f2	f3
0	0	1	0
1	0	1	0
2	0	1	1
3	0	1	1
4	0	0	1

Большое число пропущенных значений

- Если у признака слишком много пропусков, его качество под вопросом.
- Иногда лучше сразу убрать такой признак, чем пытаться его заполнять.

	f1	f2	f3
0	NaN	7.0	1
1	NaN	8.0	2
2	NaN	NaN	3
3	5.0	9.0	4
4	NaN	6.0	5

Мультиколлинеарность между признаками

- Если два признака сильно коррелируют, один можно удалить, потому что он не добавляет новой информации.
- Это уменьшает размерность и улучшает устойчивость моделей (особенно линейных).

	f1	f2	f3	f4
0	10	20	5	0
1	20	40	7	1
2	30	60	6	0
3	40	80	9	1
4	50	100	8	1

Но не всегда

Корреляция между признаками ≠ одинаковая полезность

- feature_A — уровень дохода
- feature_B — количество лет образования
- target — вероятность получения кредита

Можно удалить:

- Если признаки производные от других
- Когда используем линейные модели и высокая мультиколлинерность

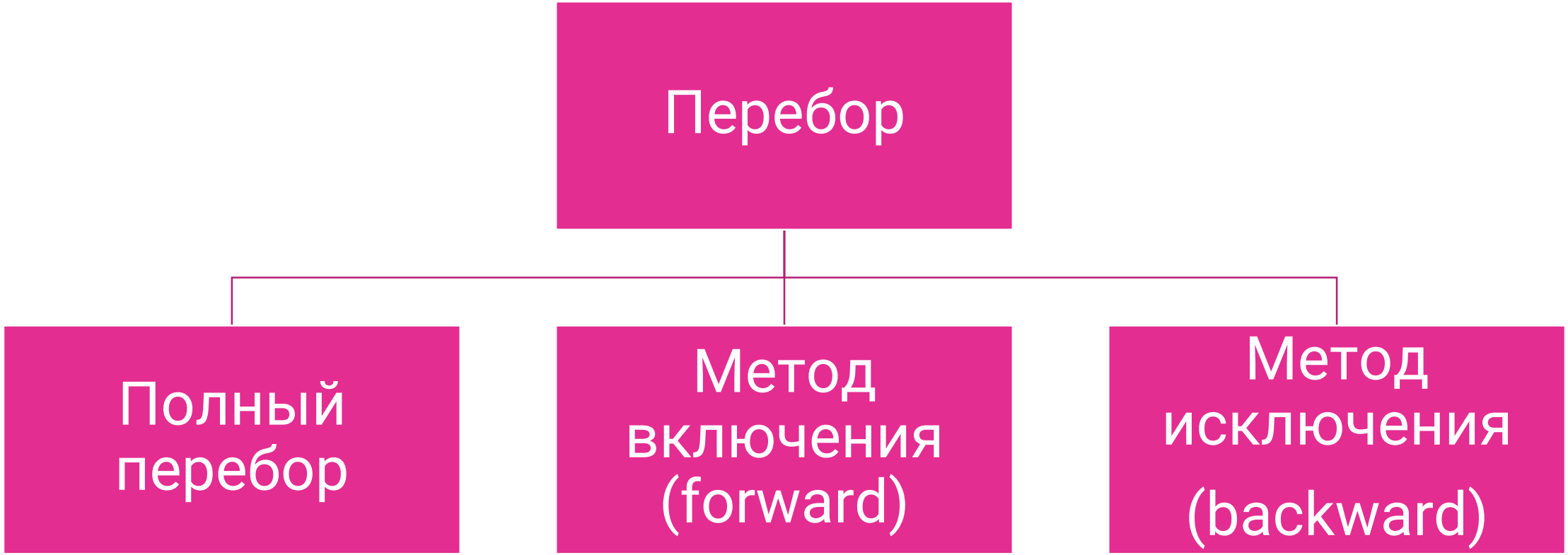
Feature selection methods

- Unsupervised
  - Drop incomplete features
  - Drop features with high multicollinearity
  - Drop features with (near-)zero variance

- Mutual info
- F-score
- Point-biserial

С использованием целевой переменной

Перебор


Перебор
Полный перебор
Метод включения (forward)
Метод исключения (backward)

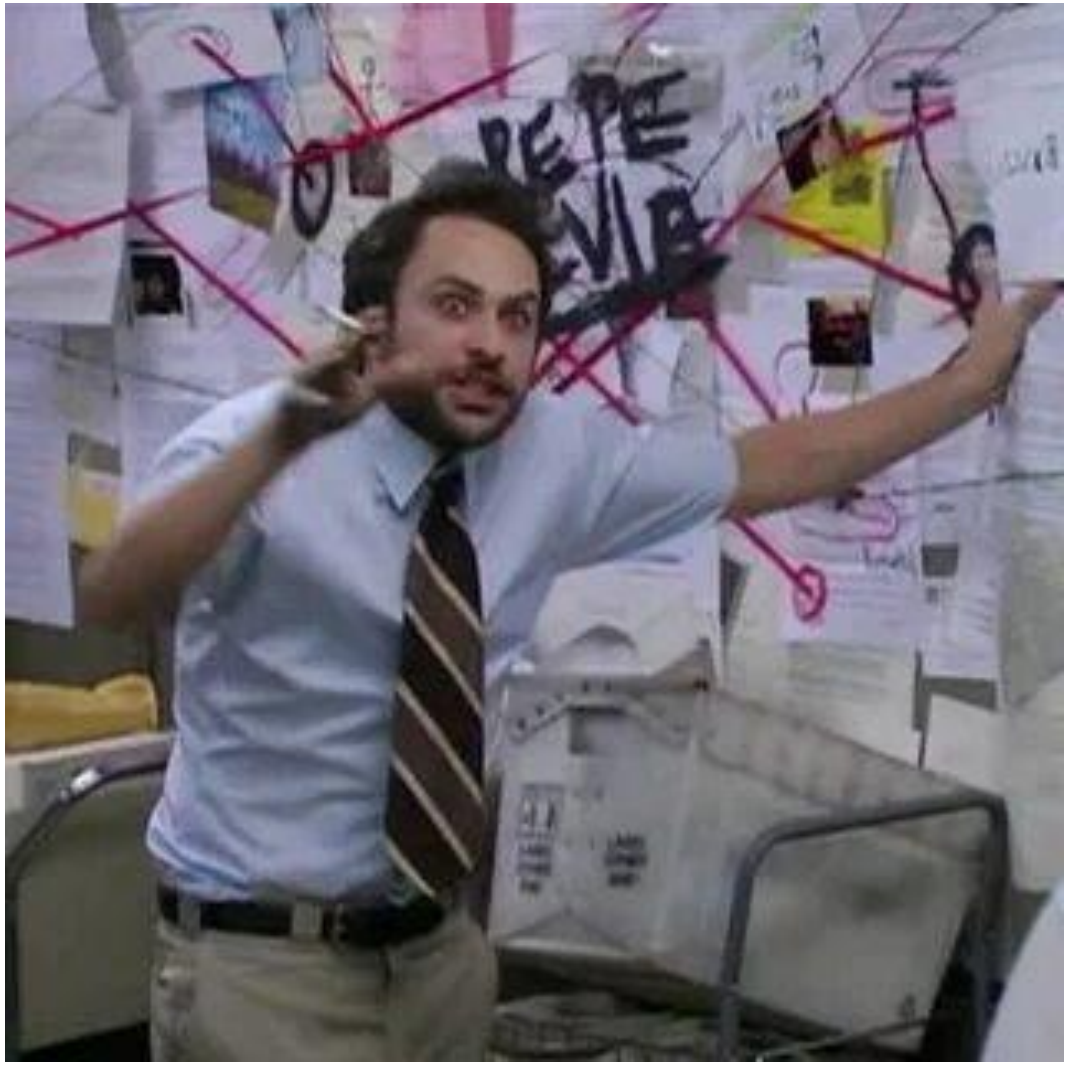
Полный перебор

Преимущества:

- простота реализации;
- гарантированный результат;
- полный перебор эффективен, когда
  - Информативных признаков не много, j<=5
  - Всего признаков немного, n<20…100

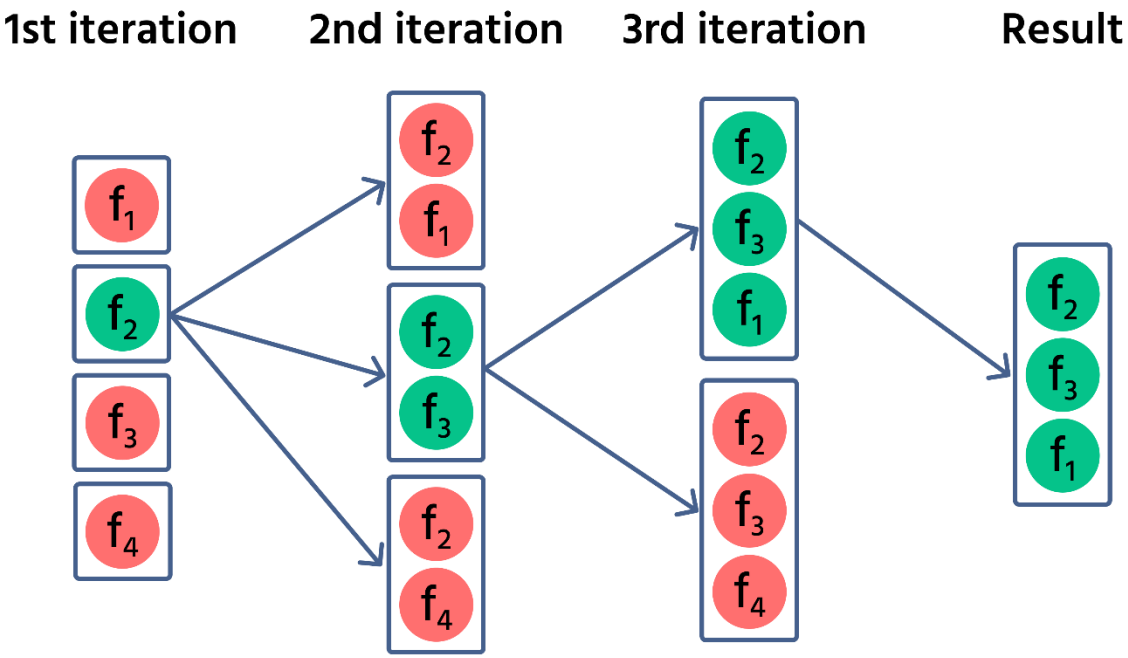
Недостатки:

- в остальных случаях оооооооочень долго - $O(2^n)$;
- чем больше перебирается вариантов, тем больше



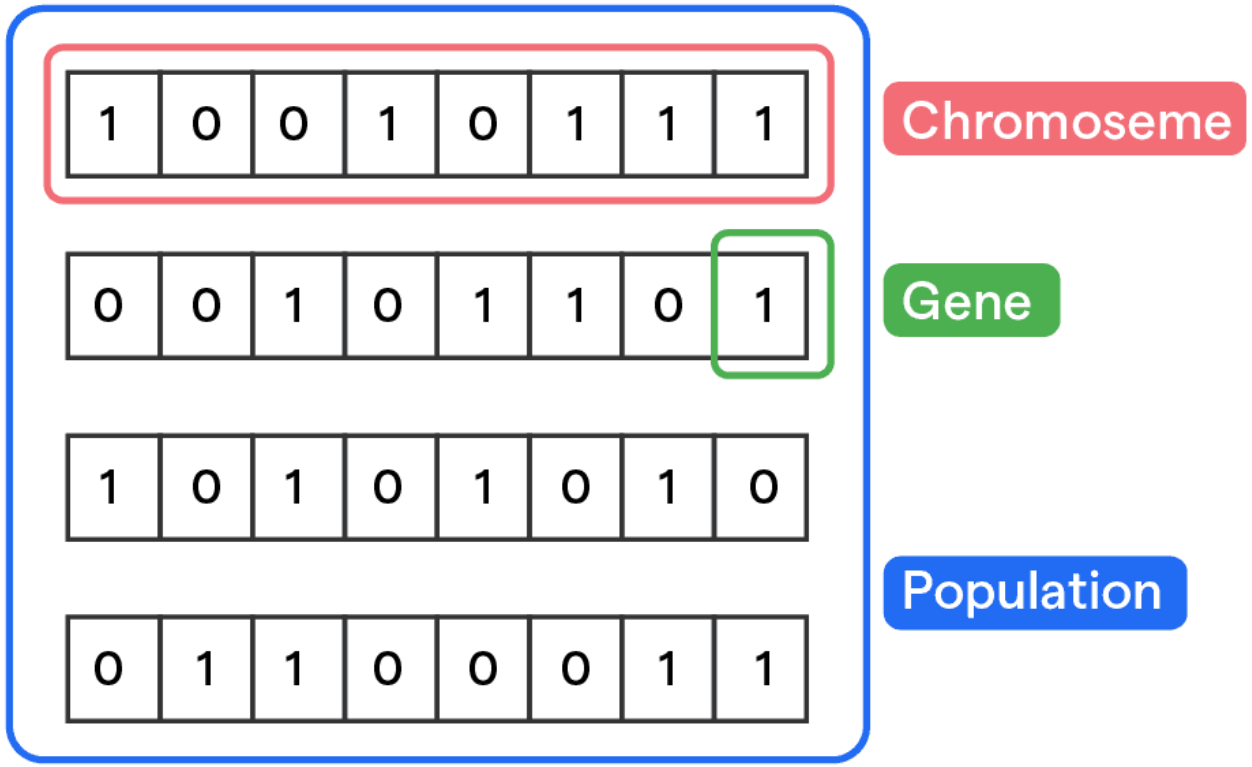
Метод включения/исключения

- Преимущество: скорость $O(n^2)$, точнее $O(nj*)$, вместо $O(2^n)$
- Недостаток: склонность включать в набор лишние признаки
- Способы устранения: Del, Add-Del(2 шага вперед, 1 назад)



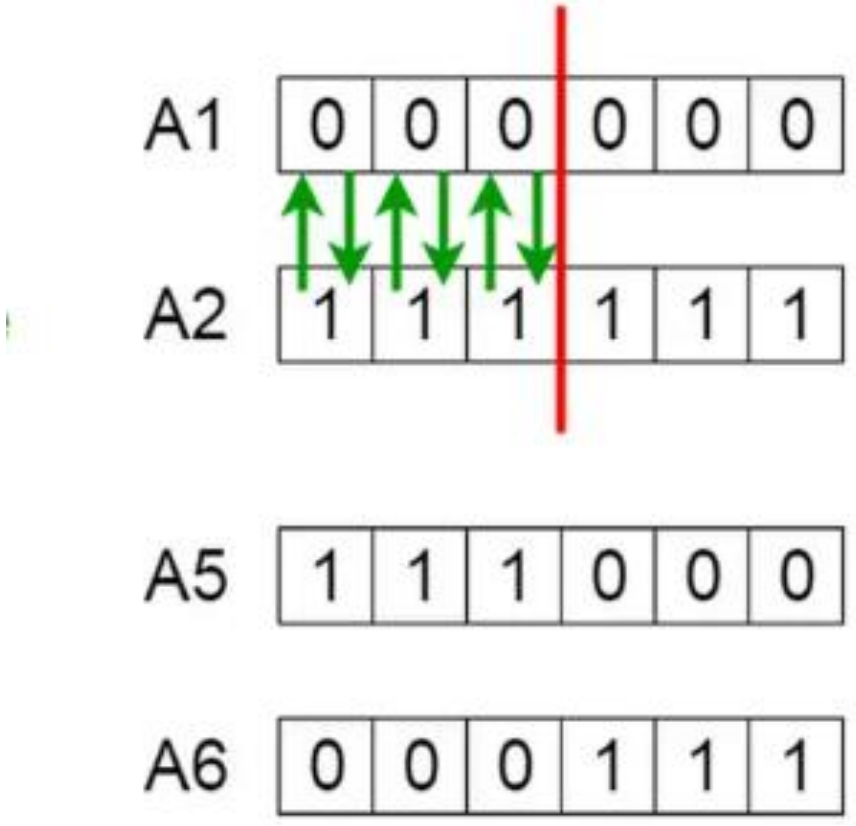
Генетический алгоритм

- Метод оптимизации, вдохновлённый принципами естественного отбора (эволюция, мутация, скрещивание).
- Набор признаков – вектор(v)=1 особь
- У каждой особи можно оценить приспособленность, f(v)



Алгоритм

- Сгенерировать случайным образом k особей, вычислить функцию приспособленности каждой особи.
- Выбрать (с некоторым вероятностным распределением) пару особей для размножения.
- С помощью смешения генотипов родителей создать двоих потомков и вычислить для потомков функции приспособленности.
- (Мутация) С вероятностью наступает следующее событие: в произвольной особи один ген мутирует.
- Две наименее приспособленные особи погибают и удаляются из популяции.



Проблемы перебора

1. Высокая вычислительная цена
2. Forward/Backward/RFE/Генетический – эвристики, не гарантируют нахождение оптимального подмножества

Статистические подходы

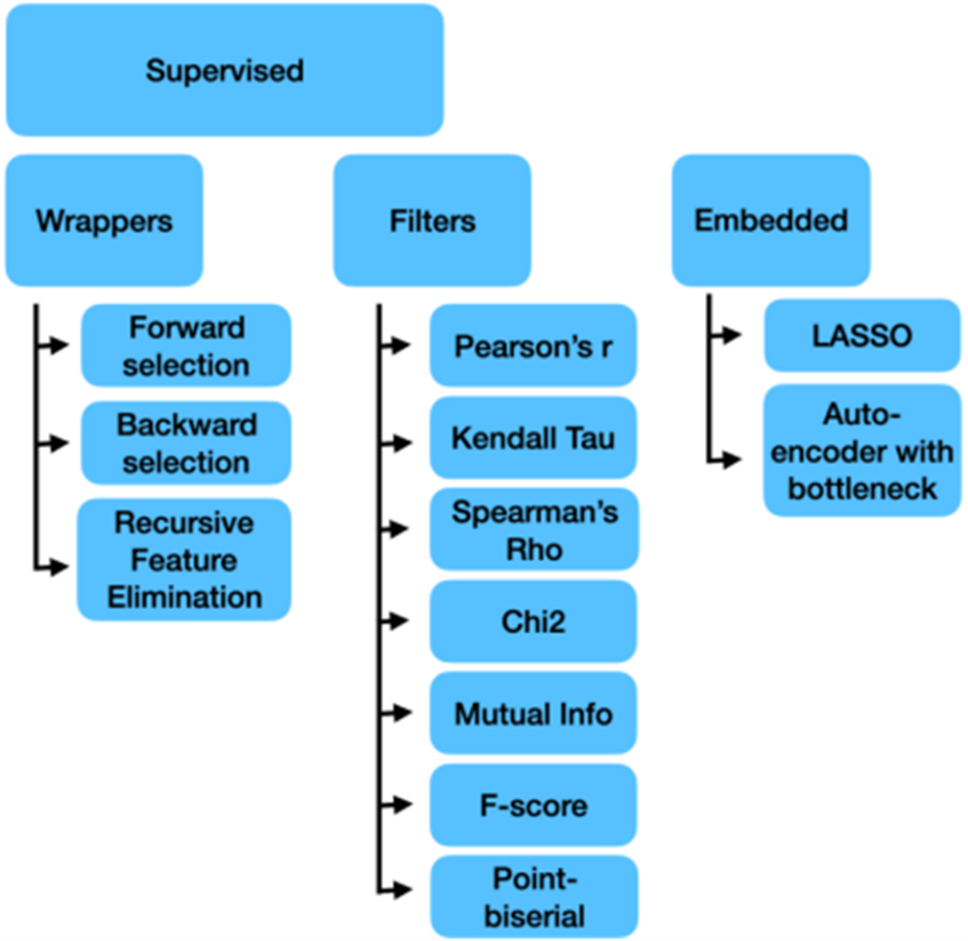
- Оценивают каждый признак по-отдельности
- Измеряют связь признака с таргетом с помощью стат. метрик

Преимущества:

- быстрые;
- Не зависят от модели;
- Просто интерпретировать

Недостатки:

- Смотрят на признак поодиночке
- Могут пропустить слабые, но в комбинации полезные признаки

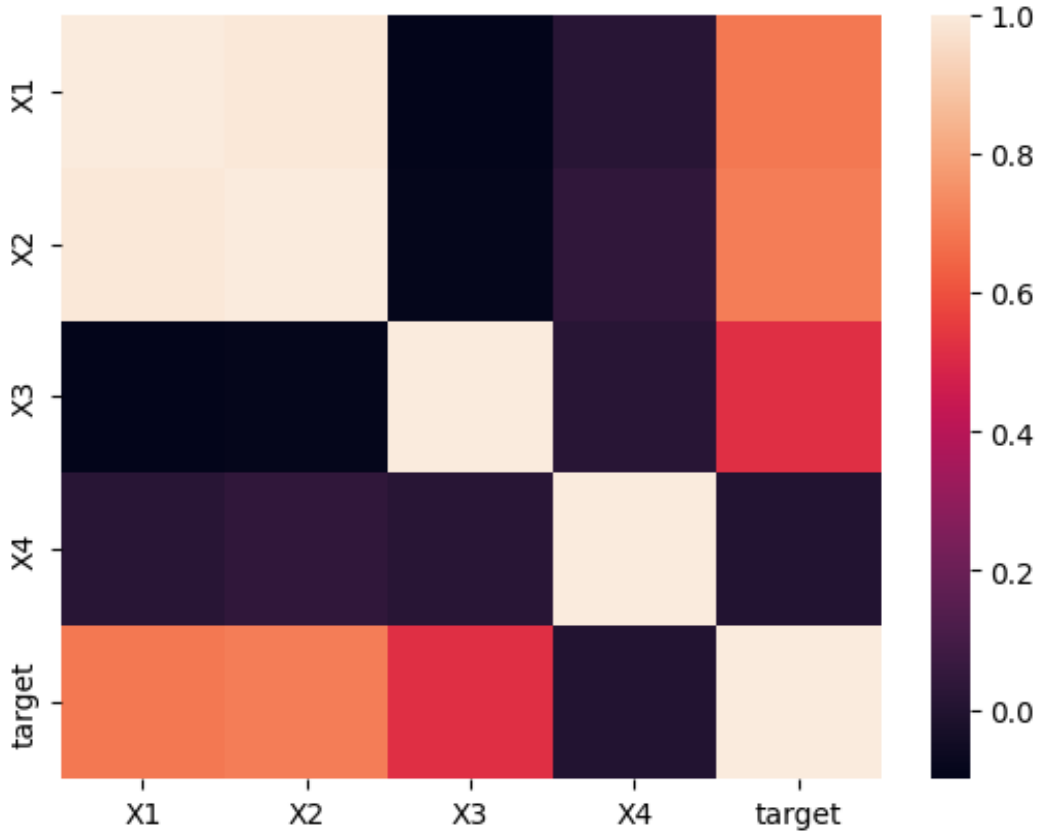


Статистические методы

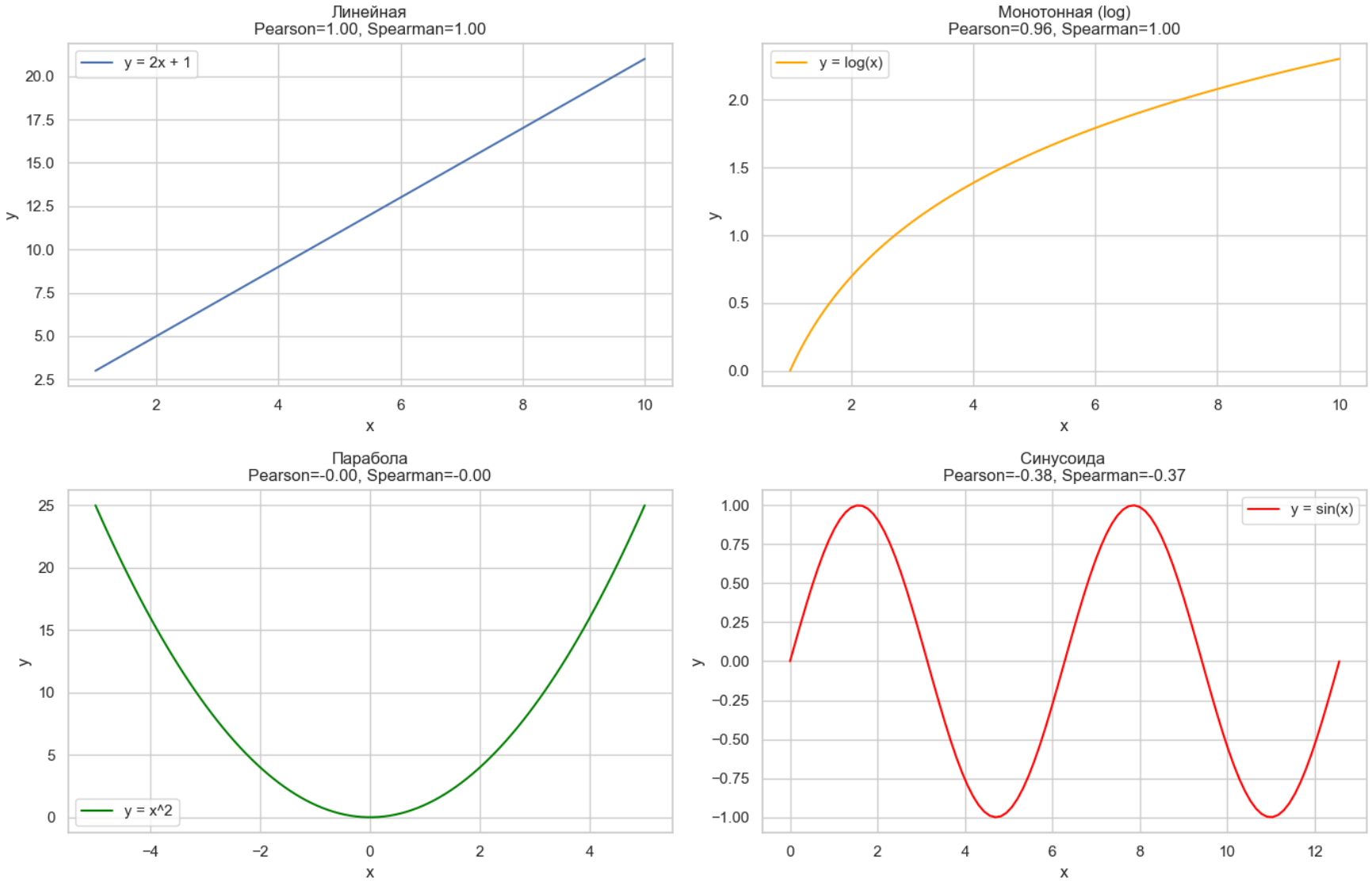
Тип признака	Тип таргета	Метод	Что проверяет	Условия применения
Числовой	Числовой	Корреляция (Pearson)	Линейная зависимость	Нормальное распределение, отсутствие выбросов
Числовой	Числовой	Корреляция (Spearman)	Монотонная зависимость	Не требует нормальности, устойчив к выбросам
Категориальный (2 категории)	Числовой	t-test	Различие средних между 2 группами	Нормальность внутри групп, равенство дисперсий
Категориальный (>2 категории)	Числовой	ANOVA (f-тест)	Различие средних между группами	То же, что и t-test + >2 групп
Числовой	Категориальный	ANOVA (обратный)	Вклад признака в классификацию	Целевой — категориальный (например, класс)
Категориальный	Категориальный	χ^2 (хи-квадрат)	Зависимость категориальных признаков	Достаточно частот в ячейках таблицы сопряжённости
Категориальный (2x2)	Категориальный	Fisher's Exact Test	Зависимость категориальных признаков	Маленькие выборки, <5 наблюдений в ячейке
Любой	Любой	Mutual Information	Обобщённая мера зависимости	Не требует распределения, устойчив к нелинейностям

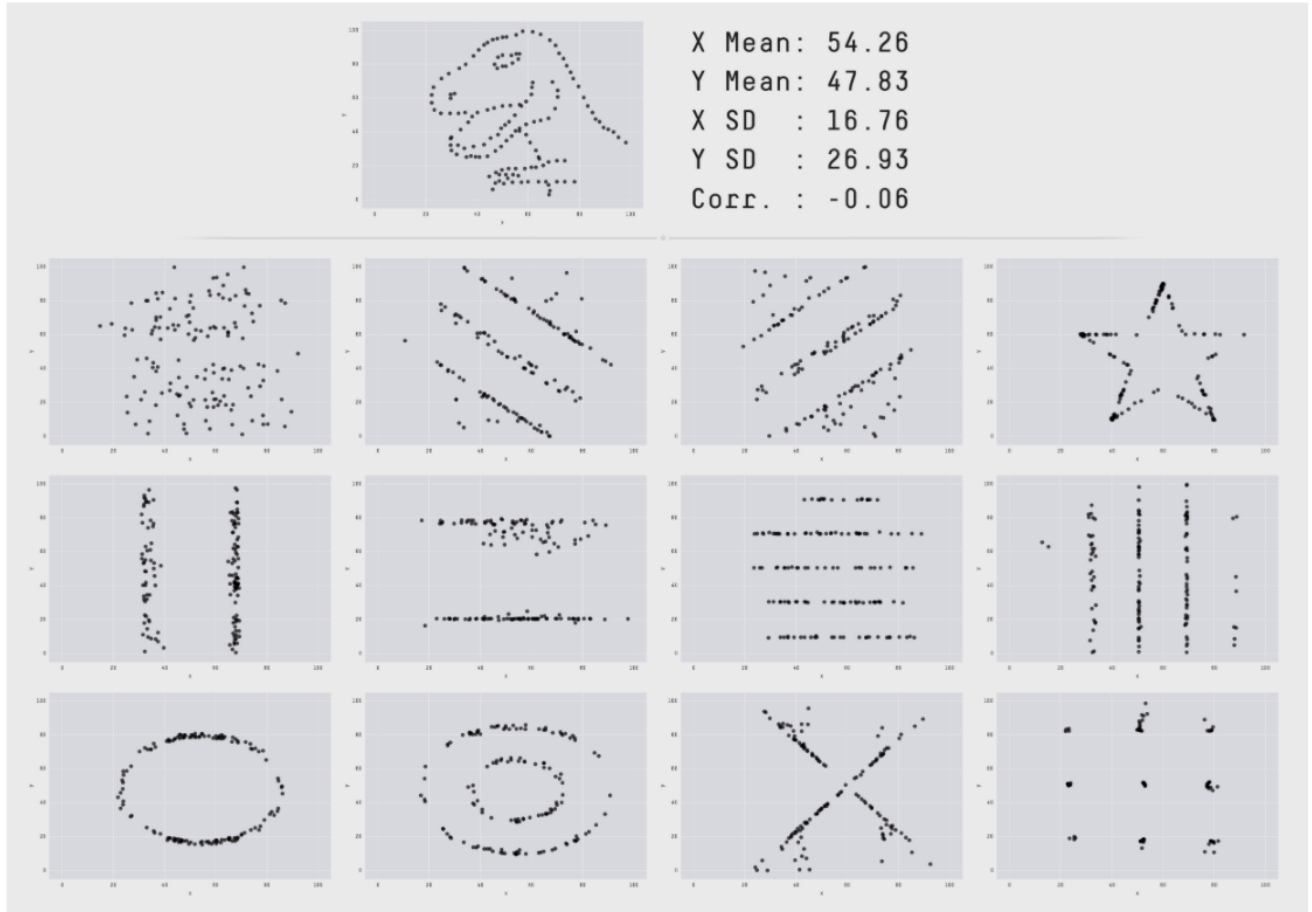
Нельзя удалять признаки с низкой корреляцией просто так

- 1. Корреляция — только про линейную связь
- Если признак связан нелинейно, корреляция может быть ≈ 0(но модель типа decision tree всё равно его использует)
- 2. Не учитывает взаимодействия между признаками
- X3 может быть слабым по отдельности, но в комбинации с X1 даёт прирост➤ такие признаки часто теряются при фильтрации



Сравнение зависимостей: линейная, монотонная и не монотонная



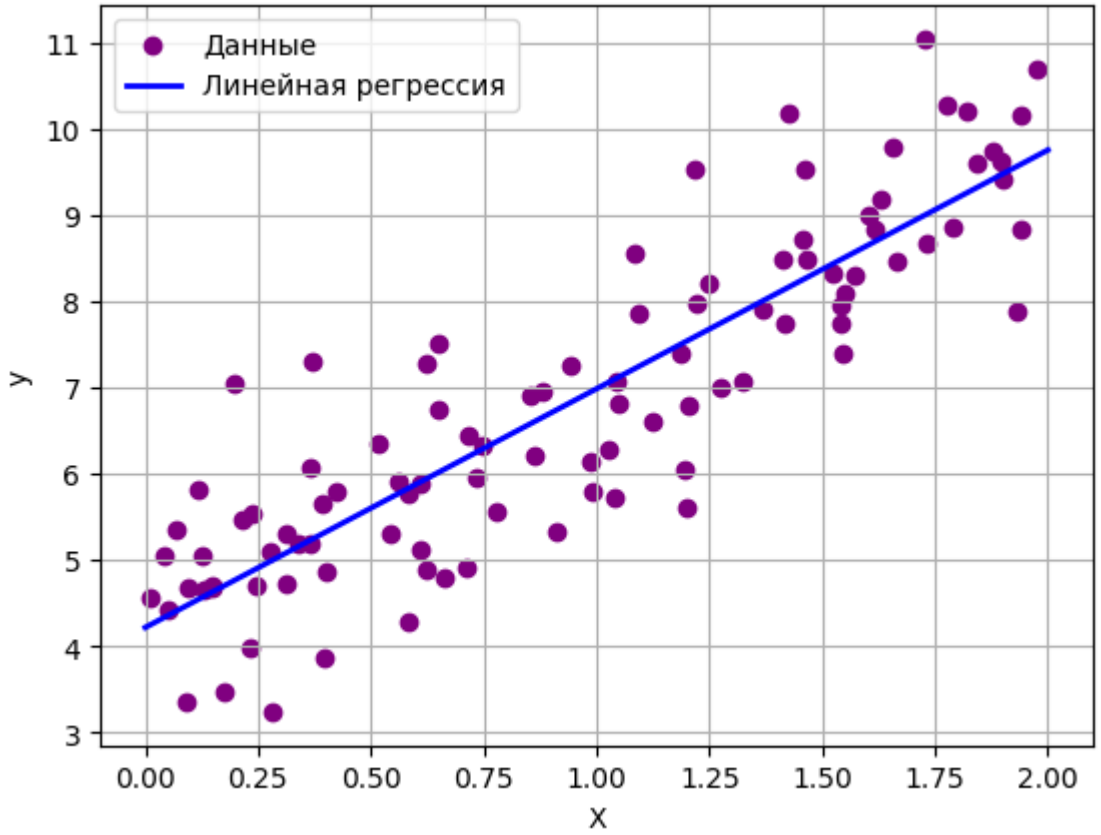

X Mean: 54.26
Y Mean: 47.83
X SD : 16.76
Y SD : 26.93
Corr. : -0.06

Отбор с использованием моделей

- Отбор признаков встроен в сам алгоритм — модель одновременно обучается и оценивает важность признаков.
- Признаки автоматически «взвешиваются» с учётом их вклада в качество модели.
- Позволяет избегать дополнительных затрат на отдельный этап отбора признаков.
- Обычно быстрее, чем wrapper-методы, особенно на больших данных.
- Работают с реальными зависимостями, а не только статистикой отдельных признаков.

Линейная регрессия

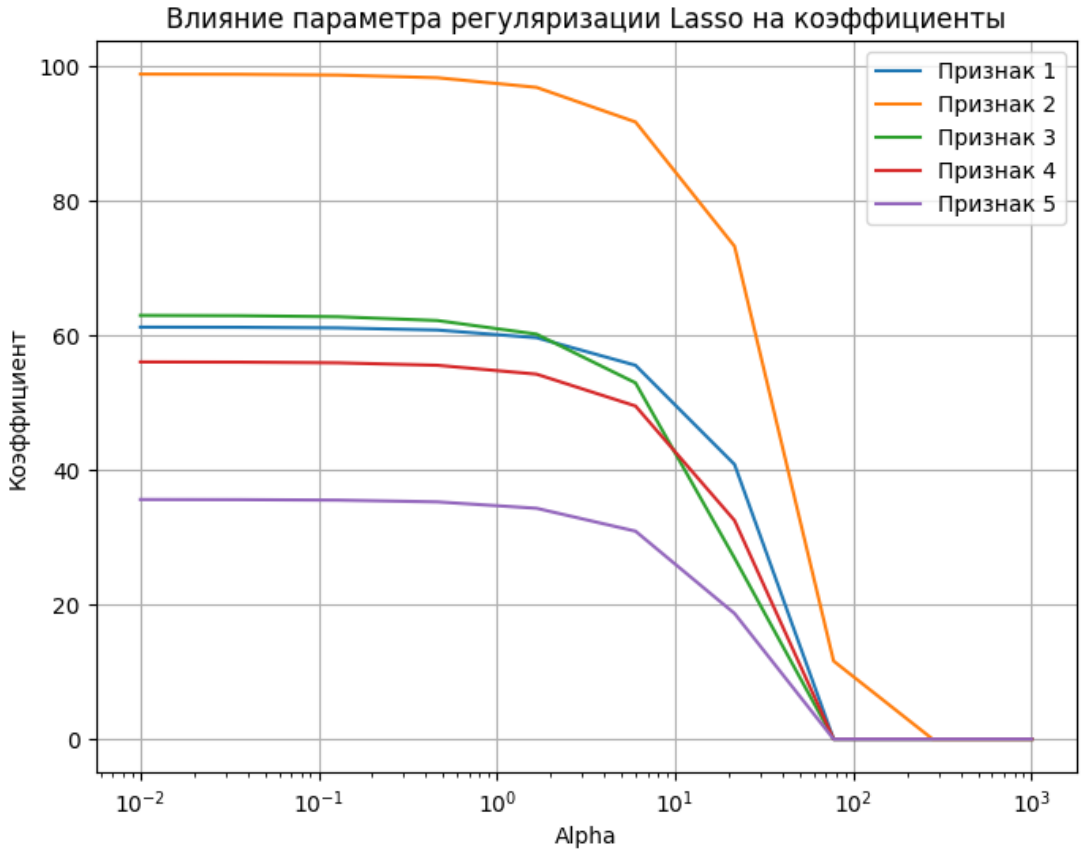
- Модель пытается найти веса, минимизируя ошибку предсказания.
- Все признаки получают коэффициенты, отражающие их влияние.
- Однако: все признаки остаются в модели, даже с очень малыми, но отличными от нуля коэффициентами.
- Может возникать переобучение, особенно при большом числе коррелирующих признаков.



$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n = w_0 + \sum_{i=1}^{n} w_i x_i$$

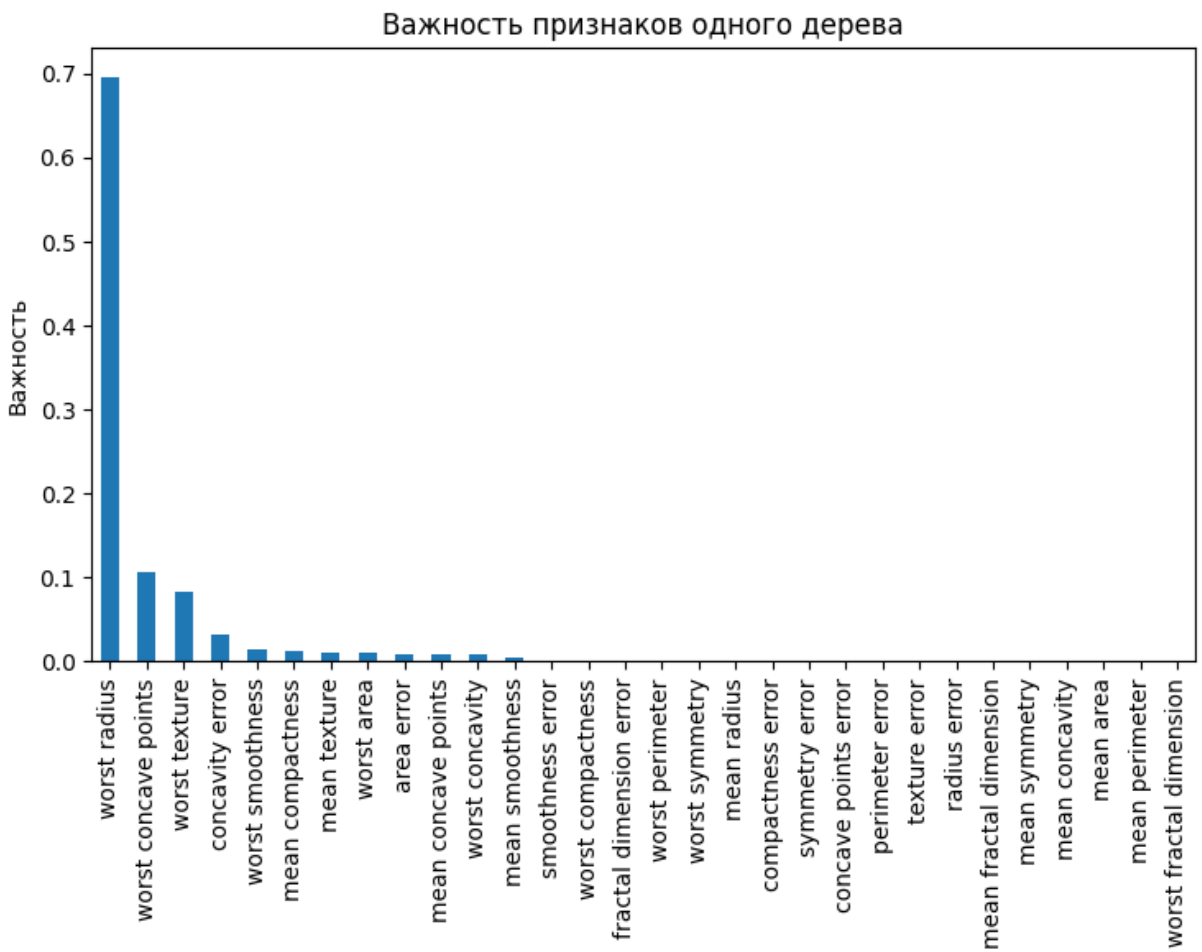
LASSO: линейная регрессия с L1-регуляризацией

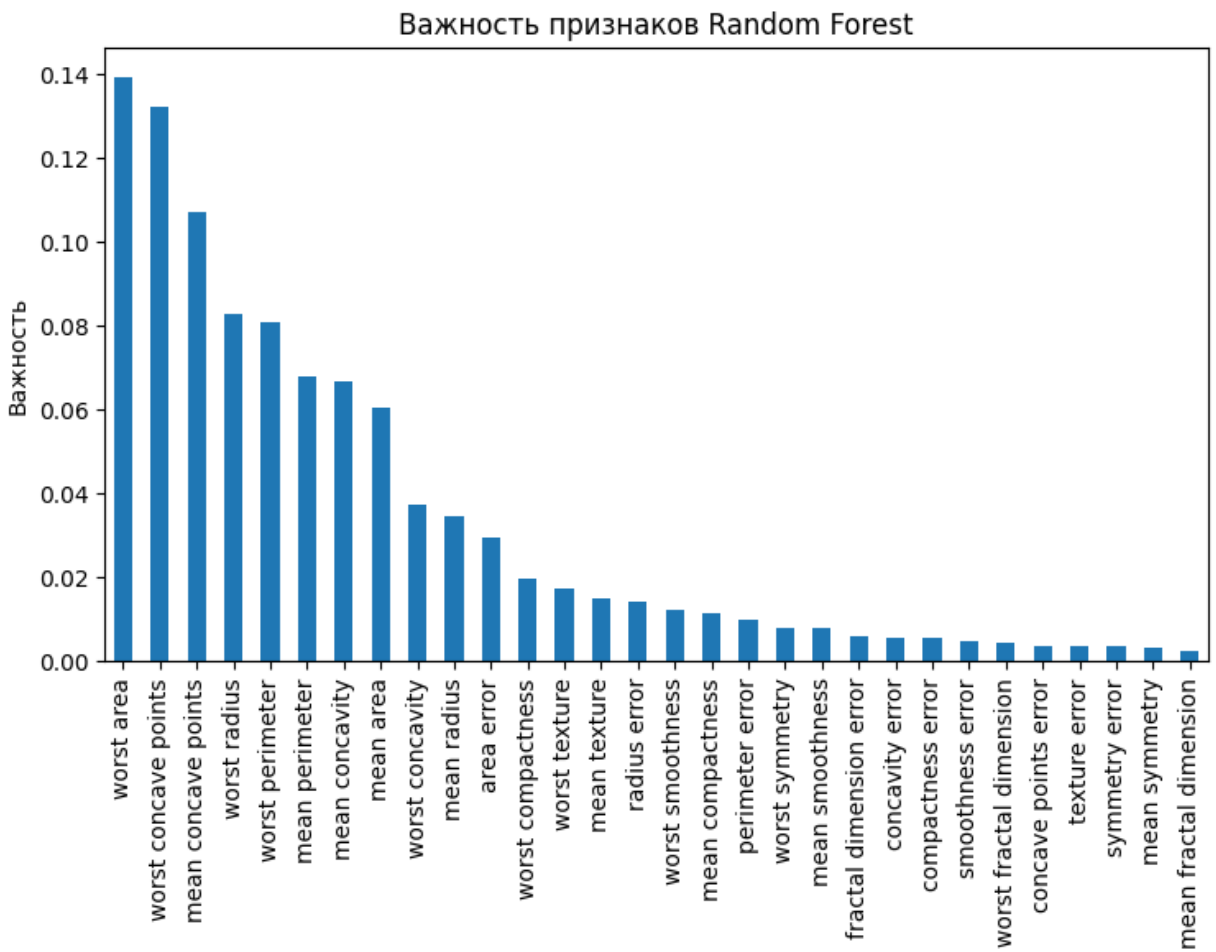
- Добавляет штраф, пропорциональный абсолютной величине весов.
- При достаточно сильном регуляризационном параметре некоторые веса могут буквально обнуляться, благодаря чему модель отбрасывает лишние признаки.



Деревья и ансамбли деревьев

- Дерево решений учится разбивать данные, выбирая признаки и пороги для максимального уменьшения ошибки.
- Важность признака — сумма уменьшений ошибки во всех узлах, где он используется. Позволяет оценить вклад признака без явной регрессии.
- Ансамбли деревьев (Random Forest, Gradient Boosting) усредняют важности по множеству деревьев, обеспечивая более стабильную и точную оценку.
- Ансамбли хорошо работают с нелинейными и сложными зависимостями, улучшая качество модели и надёжность важности признаков.


Важность признаков одного дерева
Важность
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
worst radius
worst concave points
worst texture
concavity error
worst smoothness
mean compactness
mean texture
worst area
area error
mean concave points
worst concavity
mean smoothness
smoothness error
worst compactness
fractal dimension error
worst perimeter
worst symmetry
mean radius
compactness error
symmetry error
concave points error
perimeter error
texture error
radius error
mean fractal dimension
mean symmetry
mean concavity
mean area
mean perimeter
worst fractal dimension

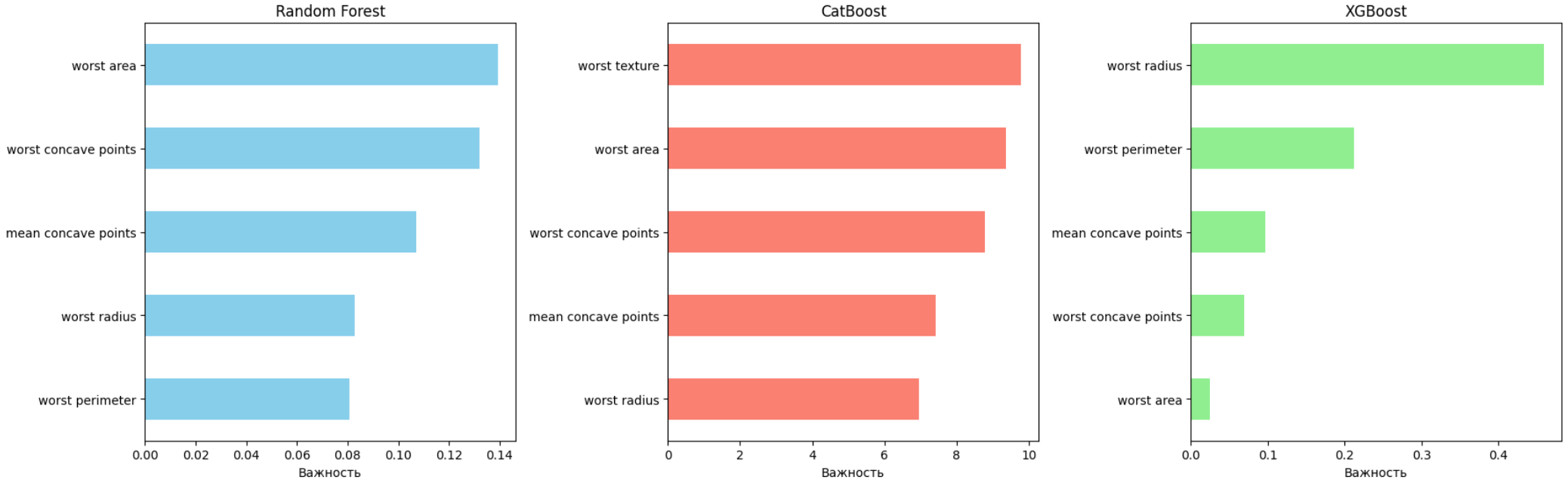

Важность признаков Random Forest
Важность
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
worst area
worst concave points
mean concave points
worst radius
worst perimeter
mean perimeter
mean concavity
mean area
worst concavity
mean radius
area error
worst compactness
worst texture
mean texture
radius error
worst smoothness
mean compactness
perimeter error
worst symmetry
mean smoothness
fractal dimension error
concavity error
compactness error
smoothness error
worst fractal dimension
concave points error
texture error
symmetry error
mean symmetry
mean fractal dimension

Преимущества Embedded методов

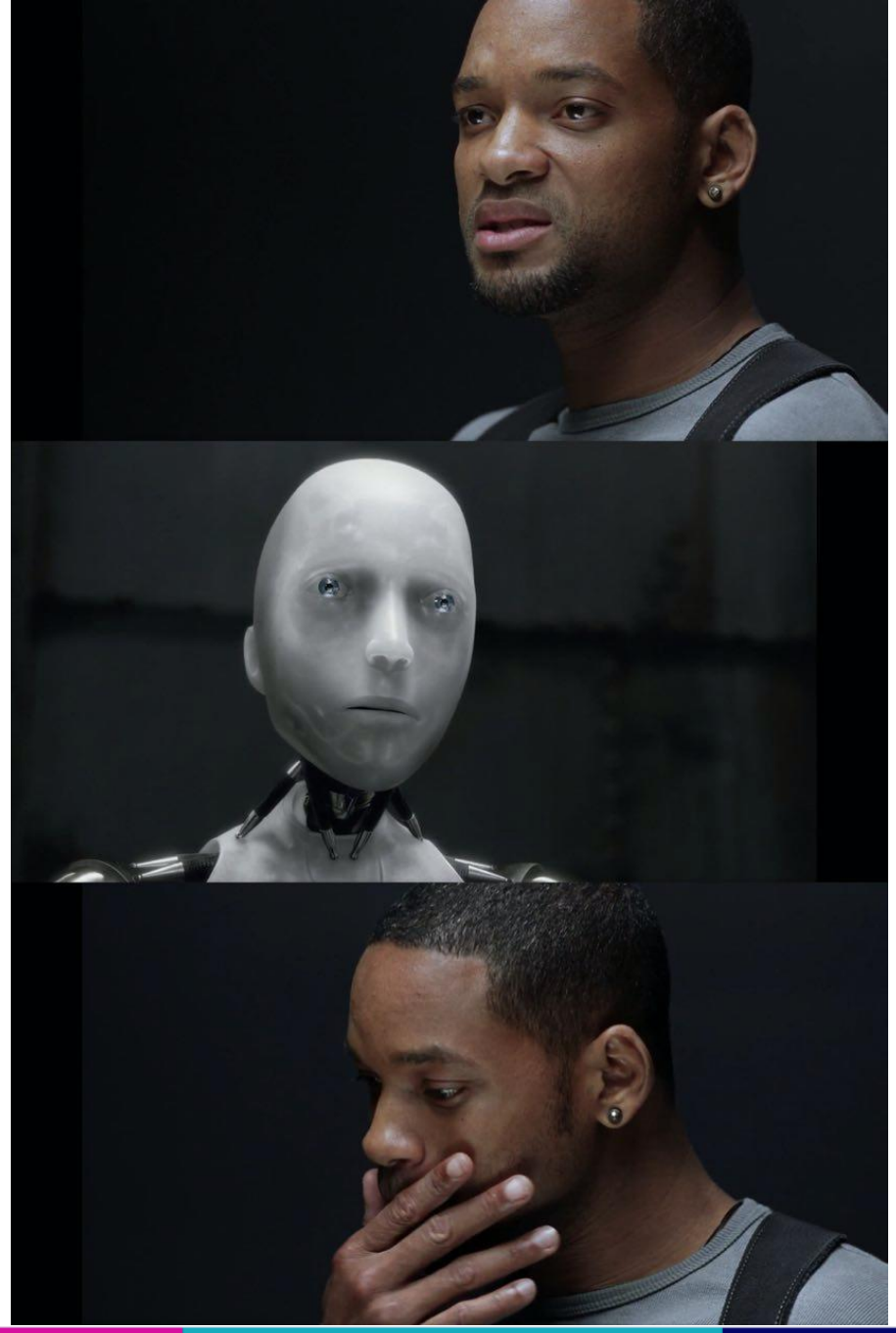
- Быстрый отбор и обучение одновременно
- Учитывают влияние признаков на итоговое качество
- Обычно дают интерпретируемые метрики важности
- Масштабируются на сотни и тысячи признаков

Ограничения и подводные камни **Embedded** методов

- Результат зависит от выбранной модели — разные модели выбирают разные признаки.
- Линейные модели не выявляют сложные взаимодействия.
- В ансамблях важности могут быть размыты из-за корреляций между признаками.
- Рекомендуется комбинировать с другими методами отбора.

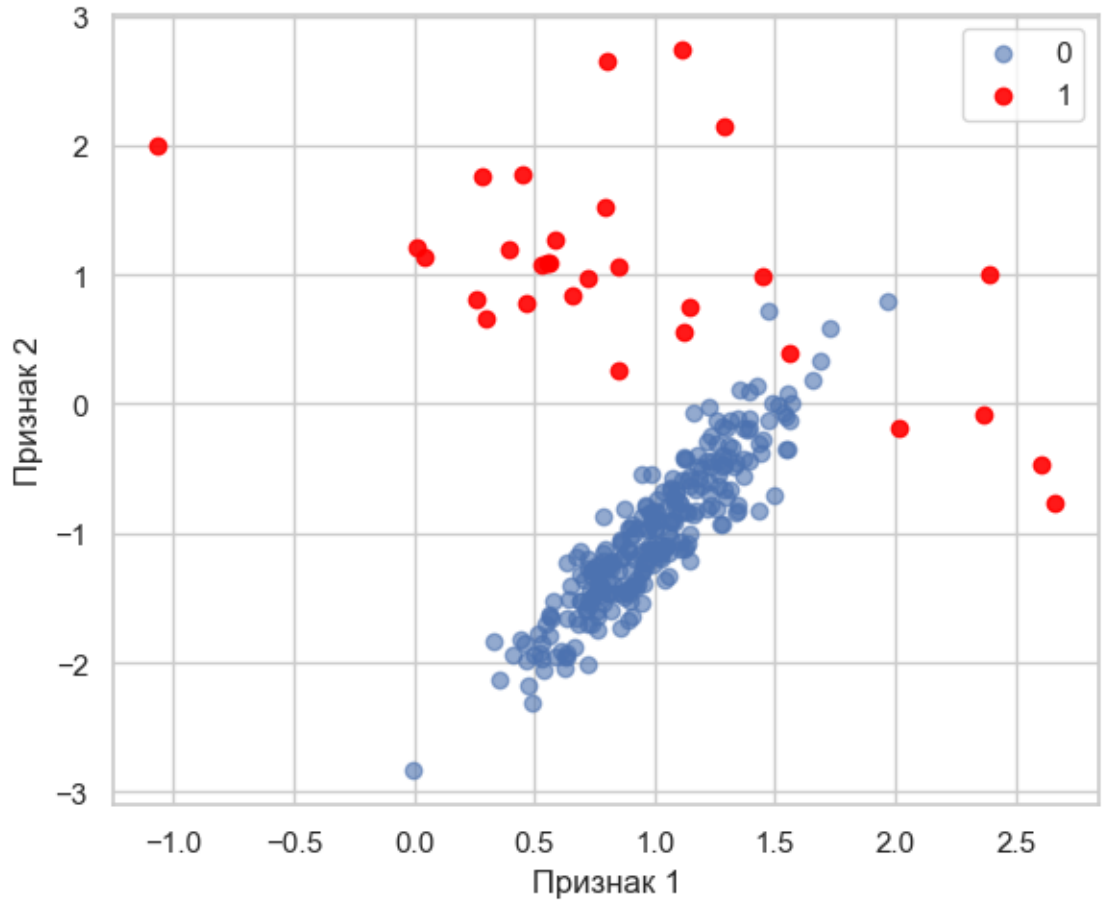


Синтетические данные



Что и зачем?

- Синтетические данные – искусственно сгенерированные данные, имитирующие реальные.
- Если тренировочная выборка объектов несбалансирована(то есть доля объектов одного класса гораздо больше доли объектов второго класса), то могут возникнуть проблемы.

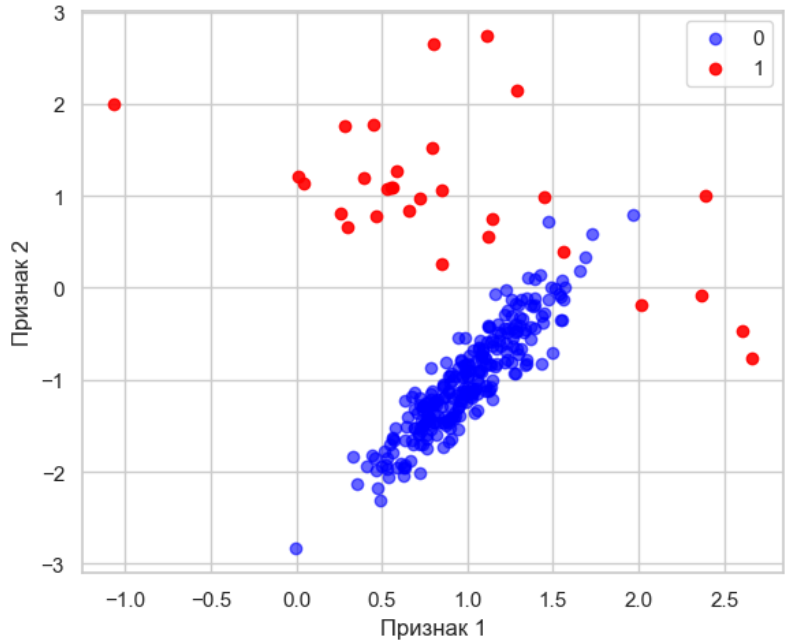


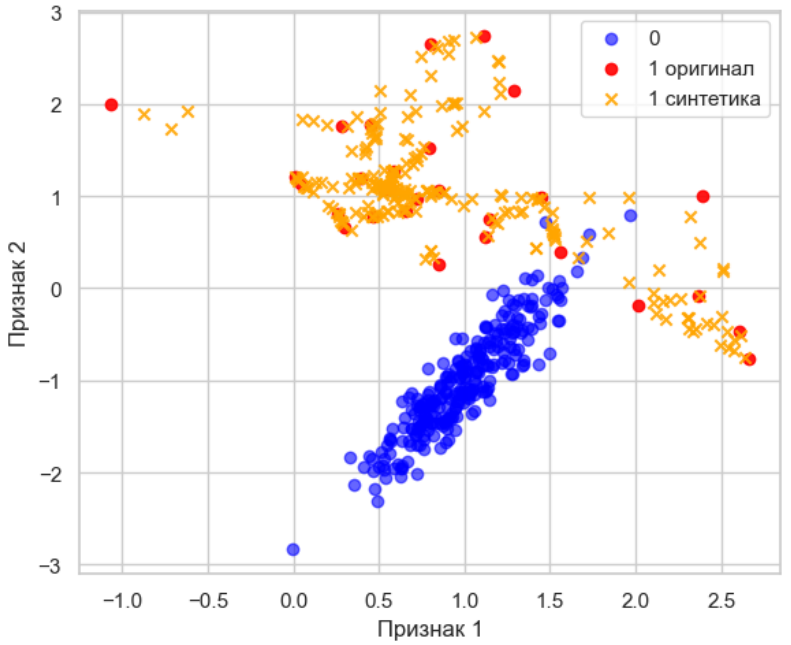
Основные методы генерации

1. Правила и шаблоны
2. Статистические модели (например, генерация по распределениям)
3. Алгоритмы машинного обучения: GAN, VAE, SMOTE

SMOTE

- SMOTE (Synthetic Minority Over-sampling Technique)
- SMOTE не копирует редкие точки, а создаёт новые, интерполируя между ближайшими соседями.
- Это помогает избежать переобучения (в отличие от простого дублирования).
- Новые точки появляются на отрезках между существующими редкими точками.





SMOTE, пример

- По паре объектов A, B можно построить синтетический объект как их линейную комбинацию

$$a \times A + (1-a) \times B$$

- где a – случайное число из отрезка [0,1].
- Например, при a=0.1 объекты

Объект	Рост	Вес	Пол
A	200	100	1
B	150	50	0
	0.1x200 + 0.9x150	0.1x100 + 0.9x50	0.1x1
C	155	55	0.1

ШИФТ
Экосистема

Теперь к ноутбуку