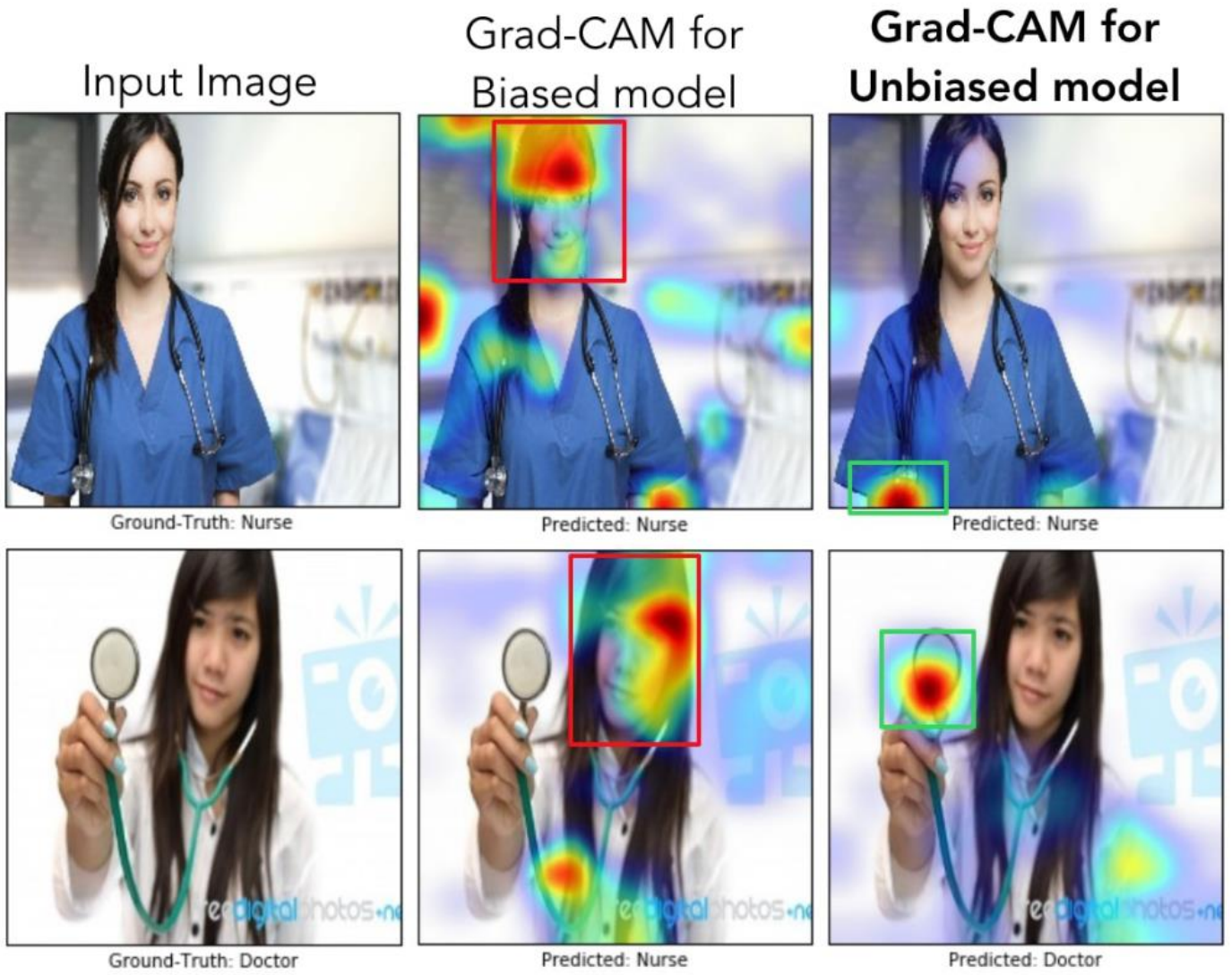
Интерпретируемость нейронных сетей

Сендерович Александра, БПМИ181

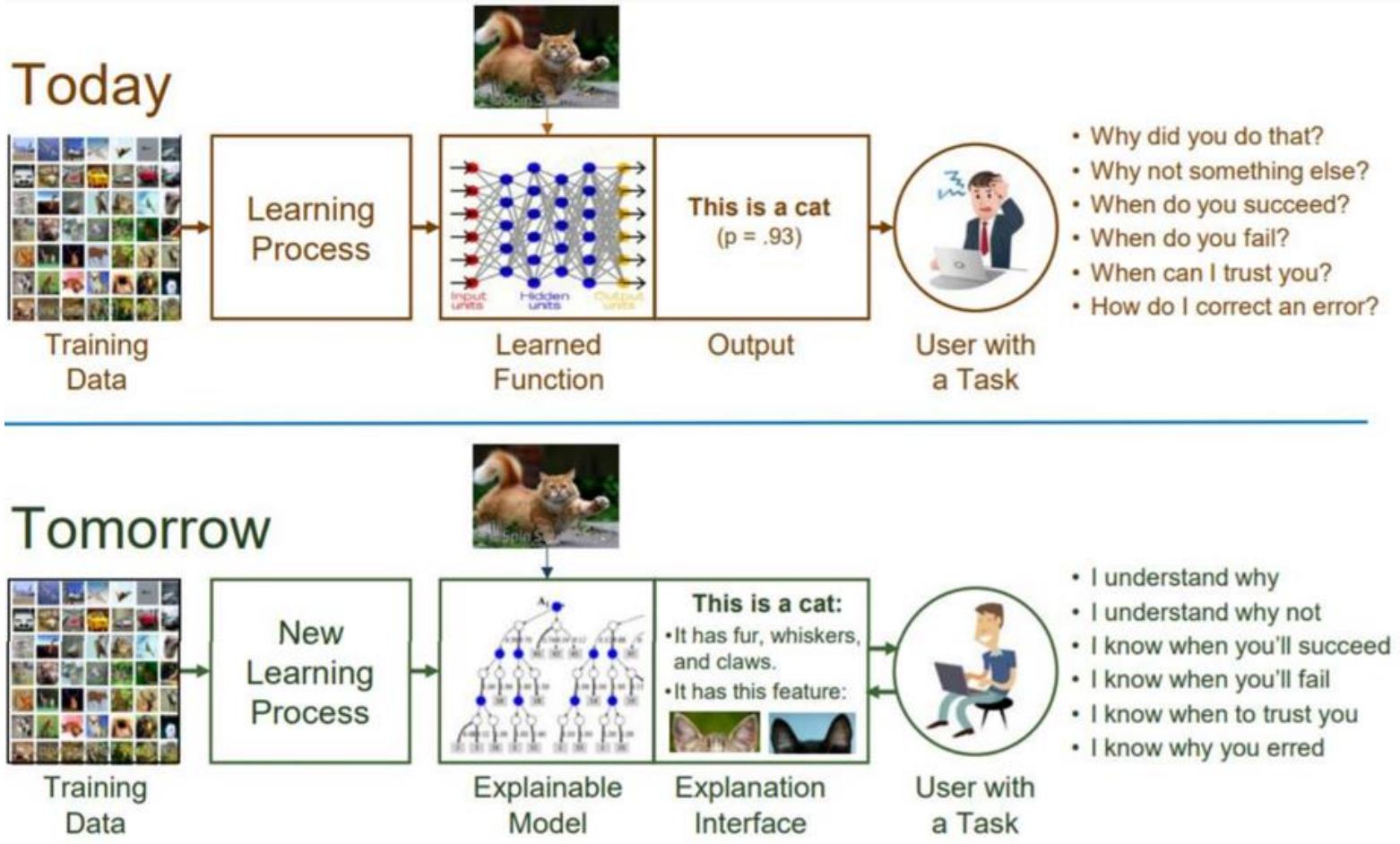
Зачем интерпретировать?

1) Если модель слабая: почему ошибается?



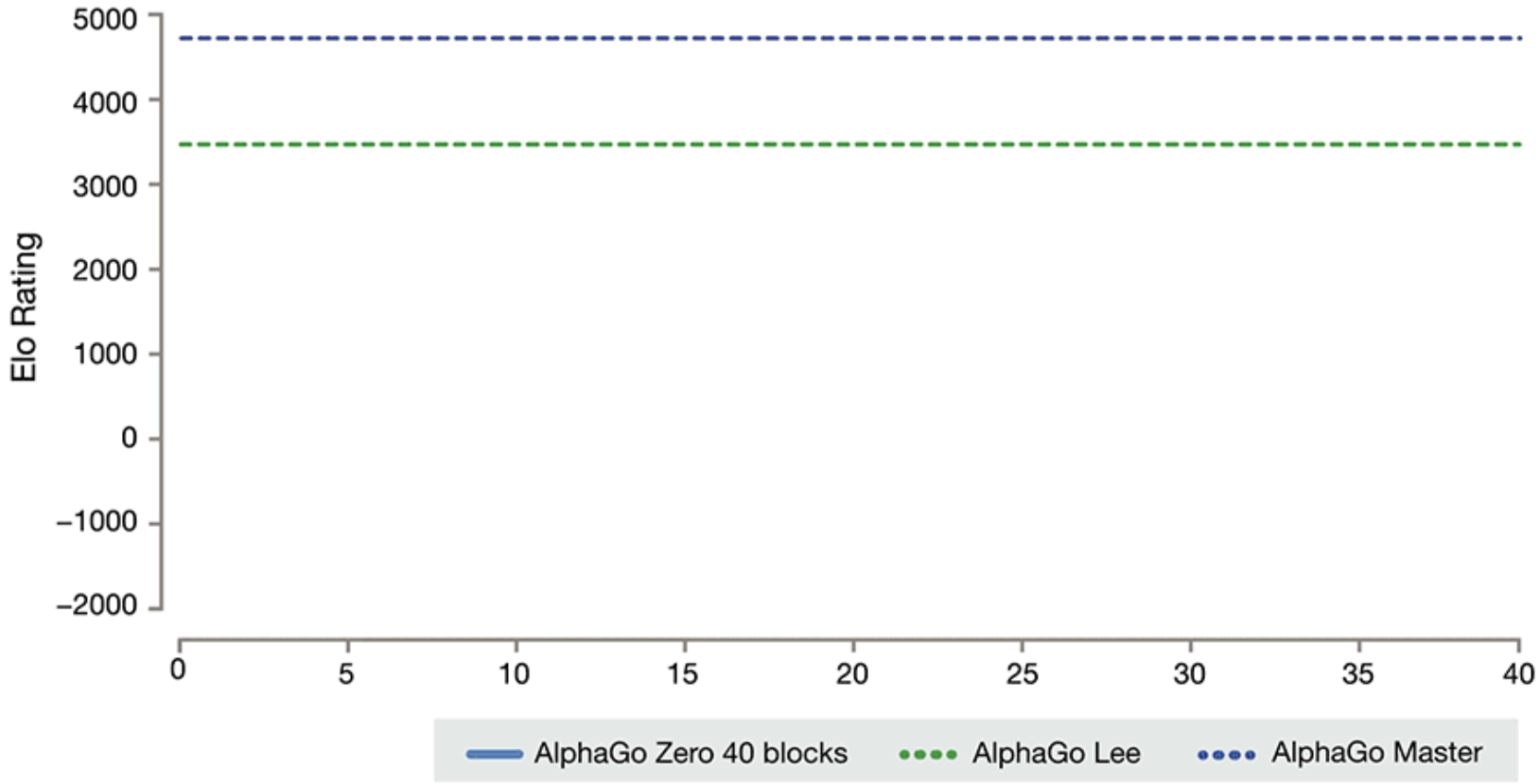
Зачем интерпретировать?

2) Если модель часто применяется: почему такие ответы? Можно ли им доверять?

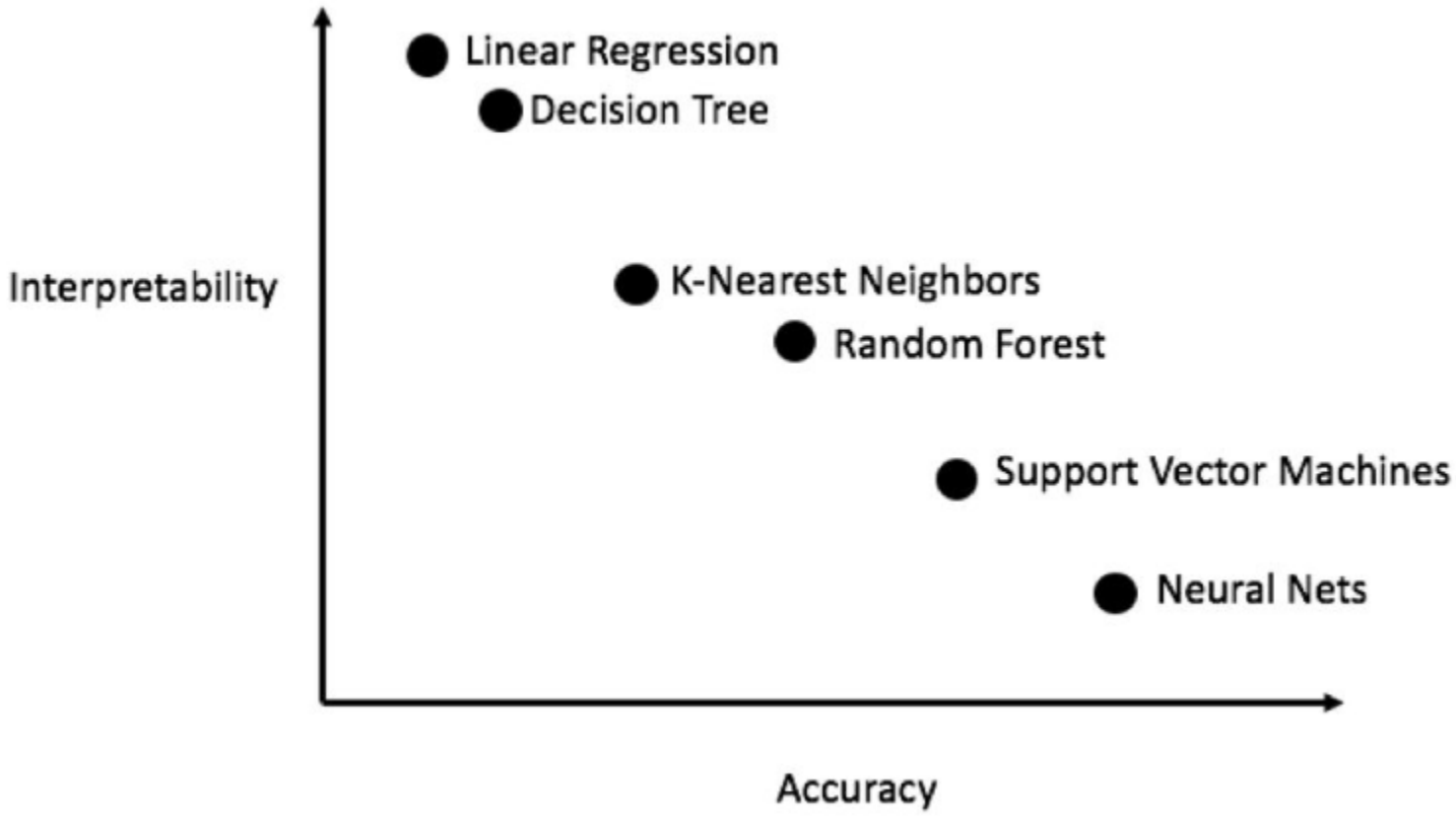


Зачем интерпретировать?

3) Если модель превзошла человека: может, можно чему-то научиться?



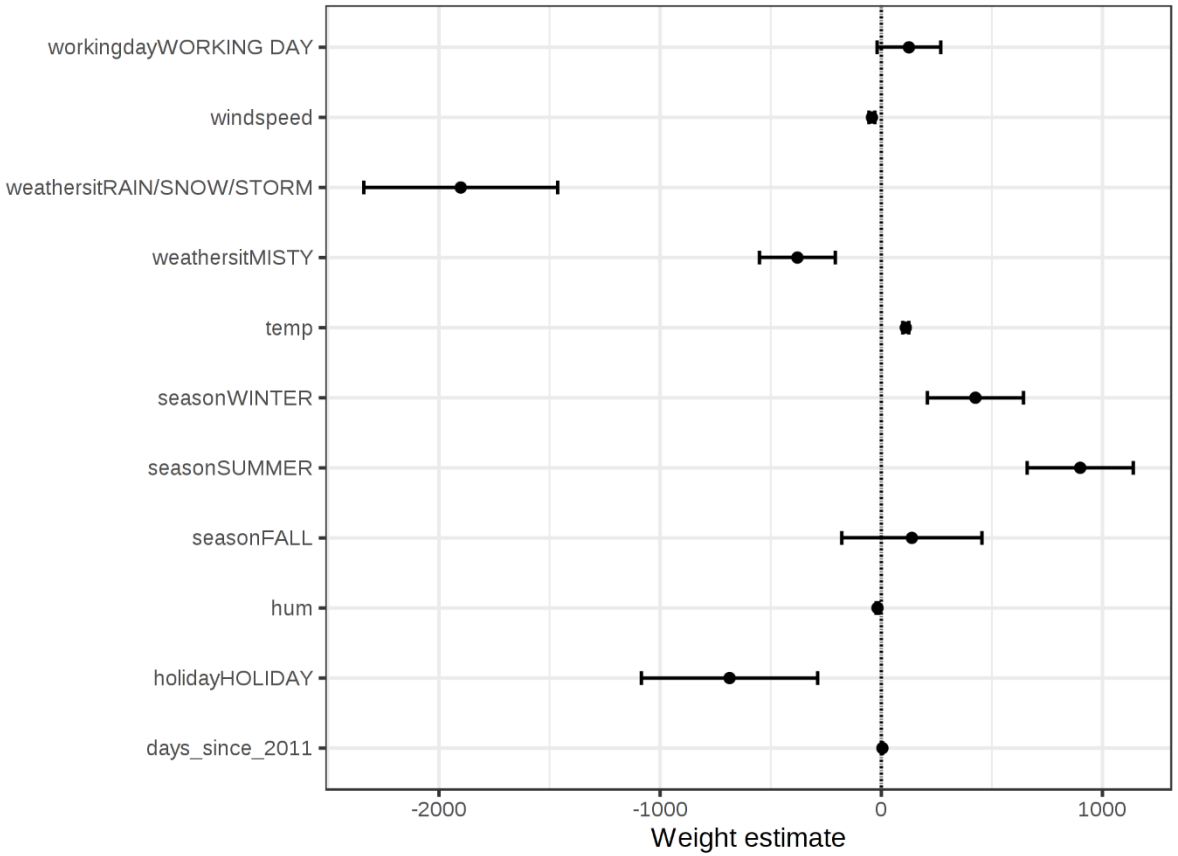
Что просто интерпретировать?



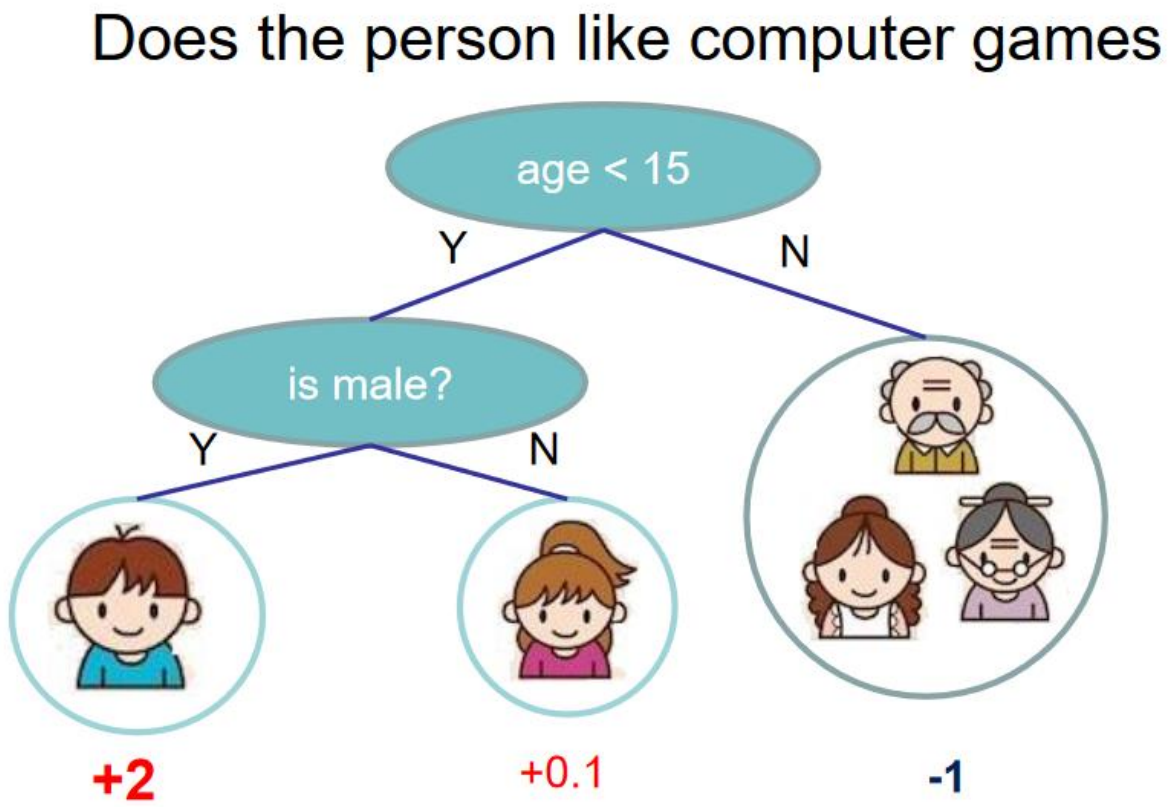
Что просто интерпретировать?

Линейная регрессия

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

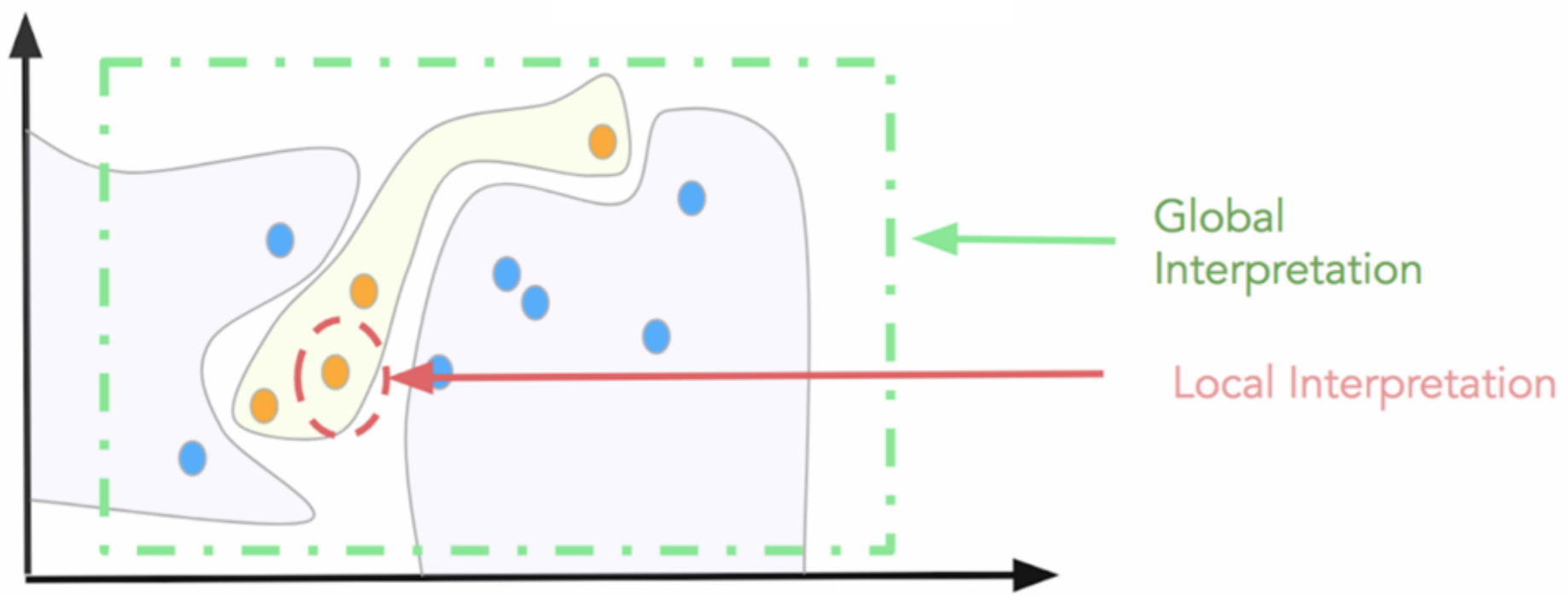


Решающие деревья



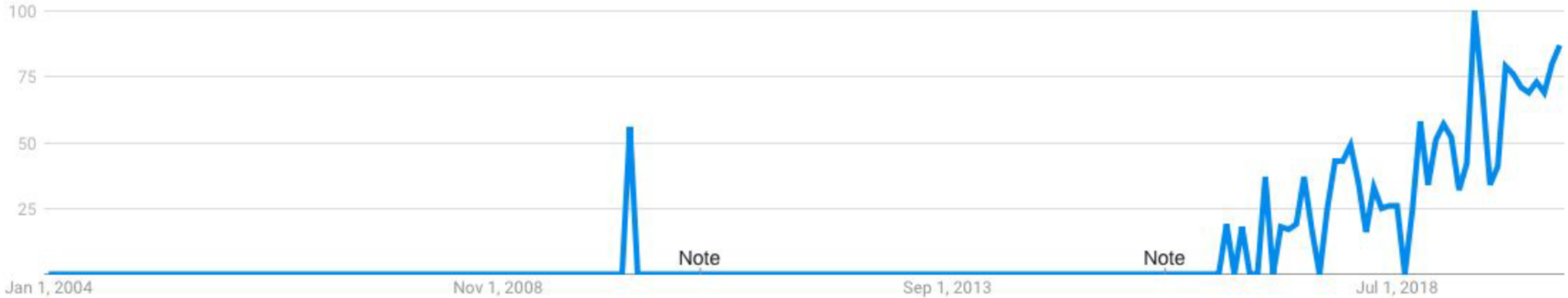
Виды алгоритмов интерпретации

- Локальные vs глобальные



- Моделезависимые (model-specific) vs моделенезависимые (model-agnostic)

Interpretability is becoming more popular



Google trends result for ‘explainable AI’

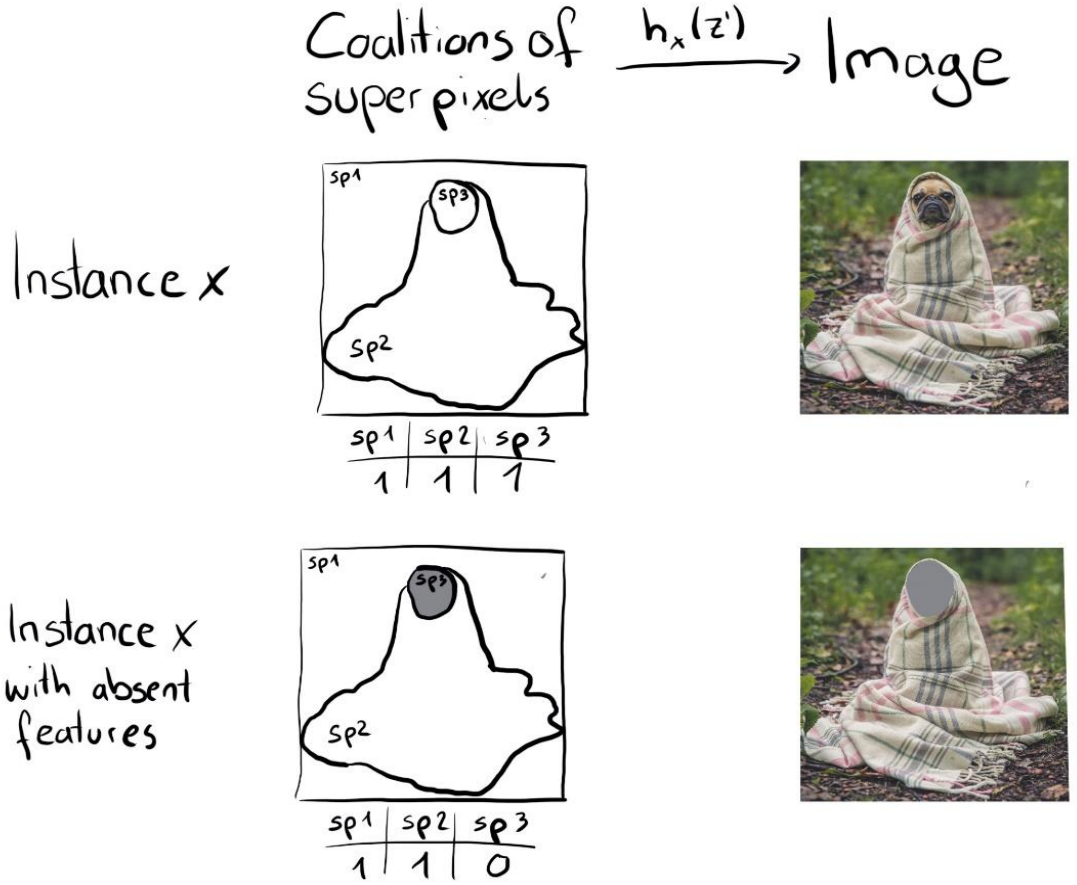
LIME (Local Surrogate, 2016)

Local Interpretable Model-agnostic Explanation

- Создаём простую модель, приближающую предсказания сложной в окрестности какого-то объекта
- Обладает local fidelity – локально точно воспроизводит сложную модель

Интерпретируемое представление данных

- Текст: бинарный вектор, обозначающий наличие или отсутствие того или иного слова
- Картинка: такой же вектор для суперпикселей
- Далее надо будет по интерпретируемому представлению воссоздавать исходный обычное



LIME

Генерируем новую обучающую выборку:

- Для текста и картинок: заменяем случайным образом единицы на нули в бинарном представлении точки x
- Для табличных данных: генерируем новые данные случайно.
- Считаем для них предсказания сложной модели

LIME

Решаем такую задачу:

$$\text{explanation}(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

Функция потерь:

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z) \left(f(z) - g(z')\right)^2$$

Ядро (веса функции потерь):

$$\pi_x(z) = exp(-D(x, z)^2 / \sigma^2)$$

LIME

$\Omega(x)$: как получить модель с не более, чем K признаками?

- L1-регуляризация: подобрать коэффициент регуляризации
- Жадно выбирать признаки

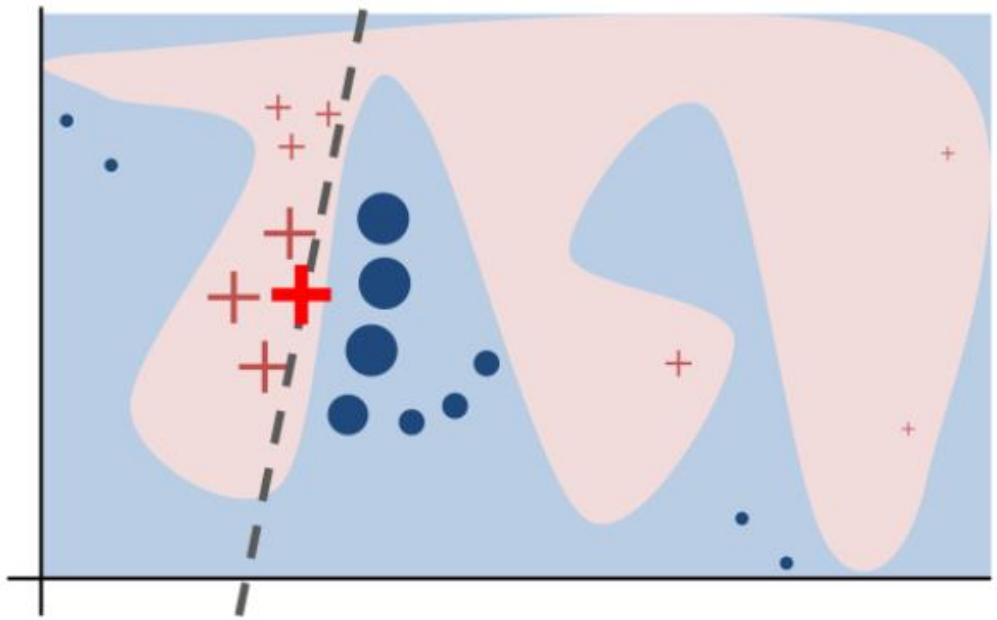
Чем больше признаков, тем сложнее интерпретировать, но тем лучше приближается сложная модель.

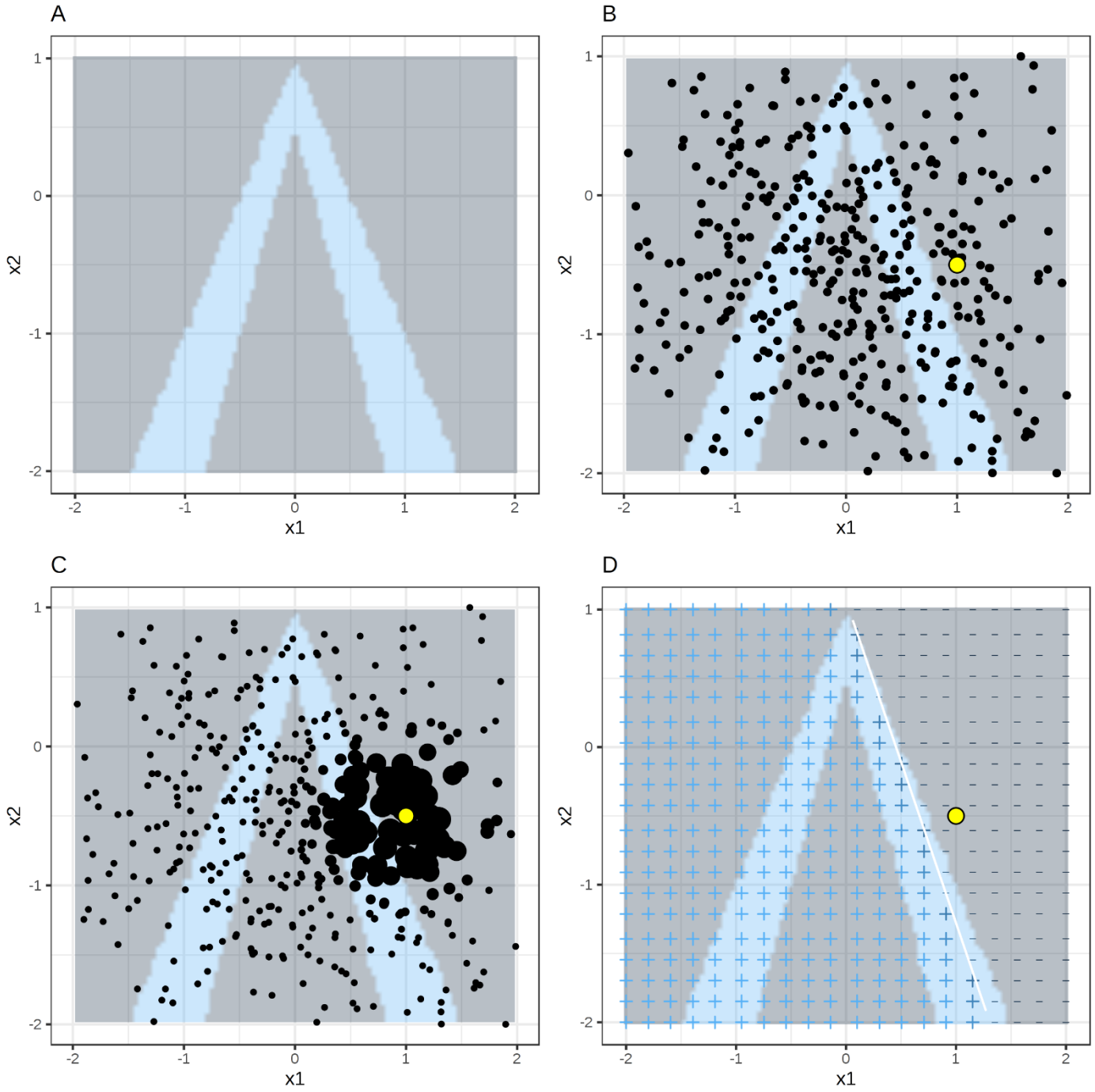
LIME

Алгоритм:

- Выбрать точку, для которой хотим объяснить предсказание
- Получить новую обучающую выборку
- Каждому объекту присвоить вес – посчитать $\pi_x(z)$.
- Обучить интерпретируемую модель
- Объяснить предсказание

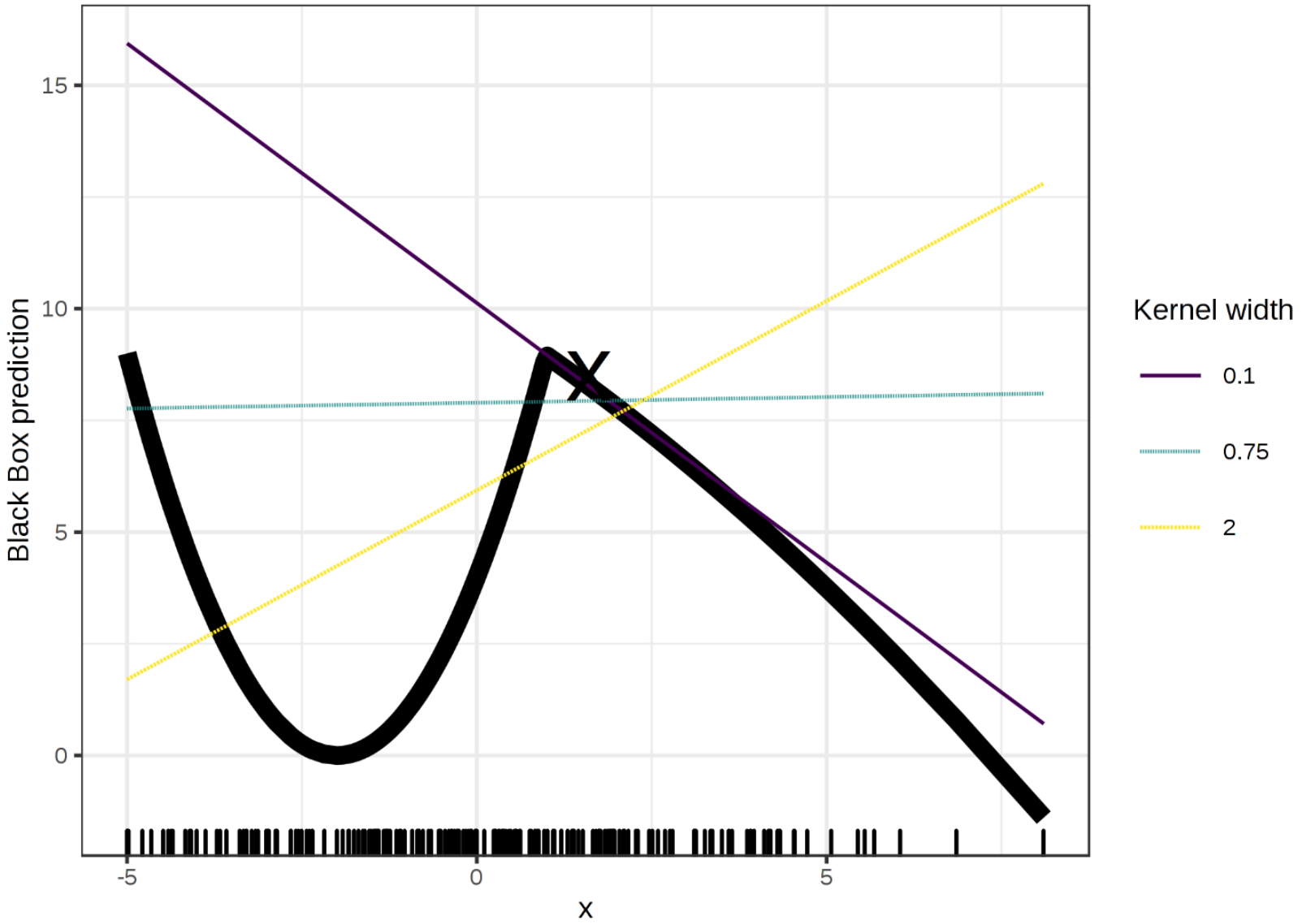
LIME





LIME

$$\pi_x(z) = exp(-D(x,z)^2/\sigma^2)$$



LIME

	CONTENT	CLASS
267	PSY is a good guy	0
173	For Christmas Song visit my channel! ;)	1

	For	Christmas	Song	visit	my	channel!	;)	prob	weight
2	1	0	1	1	0	0	1	0.17	0.57
3	0	1	1	1	1	0	1	0.17	0.71
4	1	0	0	1	1	1	1	0.99	0.71
5	1	0	1	1	1	1	1	0.99	0.86
6	0	1	1	1	0	0	1	0.17	0.57

LIME

case	label_prob	feature	feature_weight
1	0.1701170	is	0.000000
1	0.1701170	good	0.000000
1	0.1701170	a	0.000000
2	0.9939024	channel!	6.180747
2	0.9939024	Christmas	0.000000
2	0.9939024	Song	0.000000

LIME

(a) Original Image

(b) Explaining *Electric guitar*

(c) Explaining *Acoustic guitar*

(d) Explaining *Labrador*

LIME

Преимущества:

- Можно регулировать сложность итоговой модели
- Работает со всем видами данных

Недостатки:

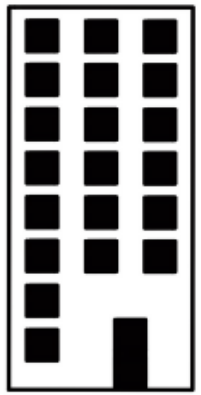
- Сложно выбрать хорошее интерпретируемое представление
- Не всегда можно приблизить простой моделью сложную
- Метод не очень стабильный (например, меняется при выборе σ)

Вектор Шепли (Shapley values, 1951, 2012)

- Метод из кооперативной теории игр
- Признак – игрок, разность текущего предсказания и среднего – выигрыш
- Честно распределяем выигрыш по игрокам (в зависимости от их вклада)

Вектор Шепли

- Рассмотрим на примере предсказания цен на квартиру



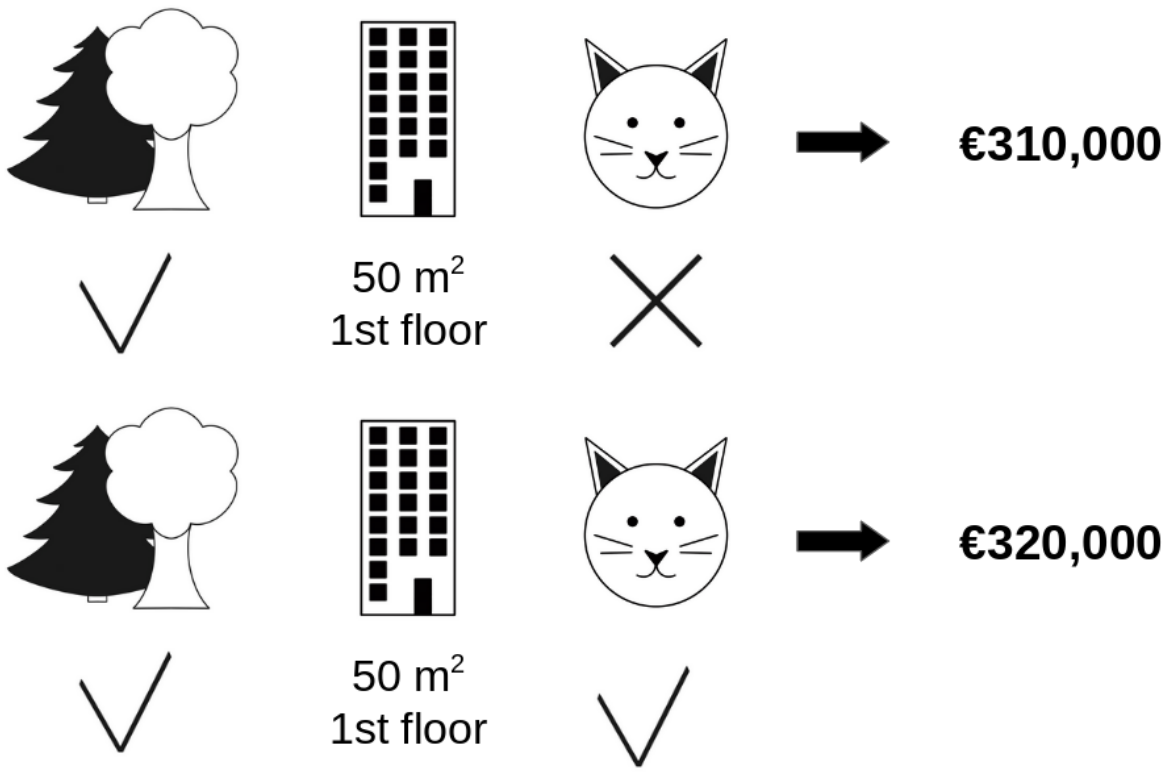
50 m²
2nd floor

€300,000

Средняя стоимость квартиры – 310 000 евро, значит выигрыш = -10 000

Вектор Шепли

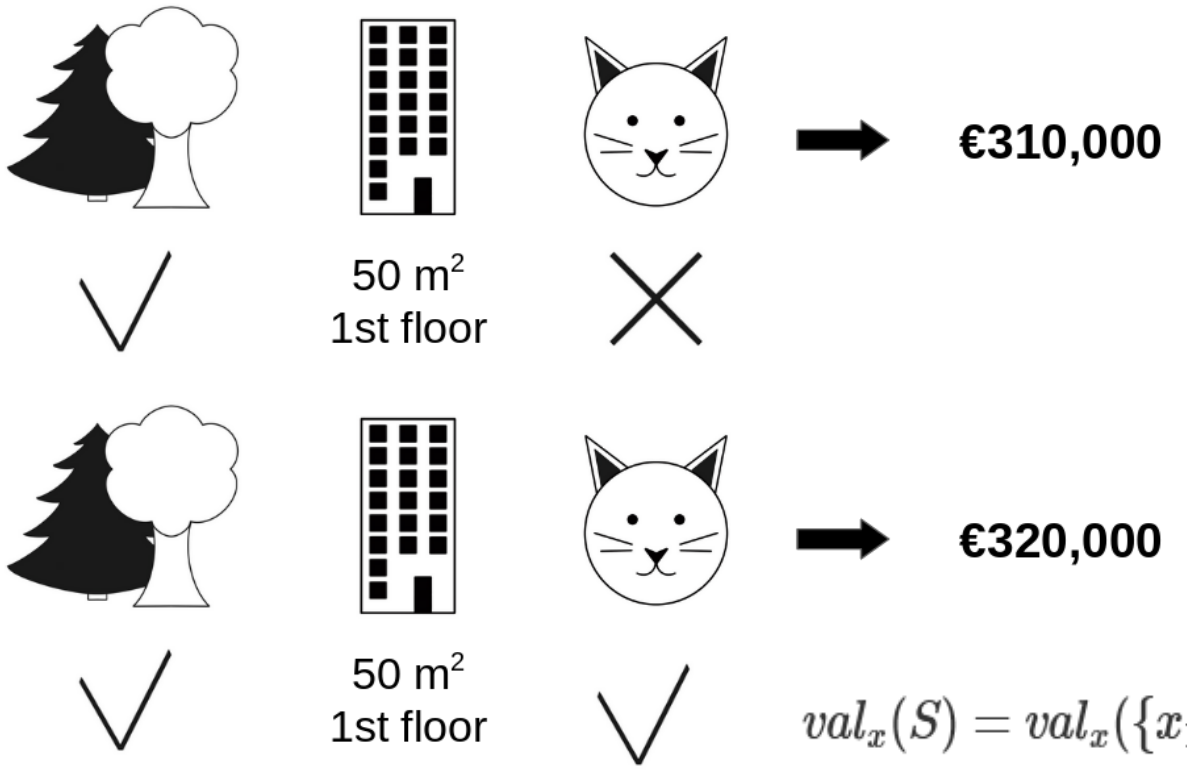
Узнаем для признака «запрещены ли кошки?» его вклад в цену.



- Выберем какое-то подмножество признаков, не включающее этот (парк и площадь)
- Для других признаков возьмём случайное значение из данных

Вектор Шепли

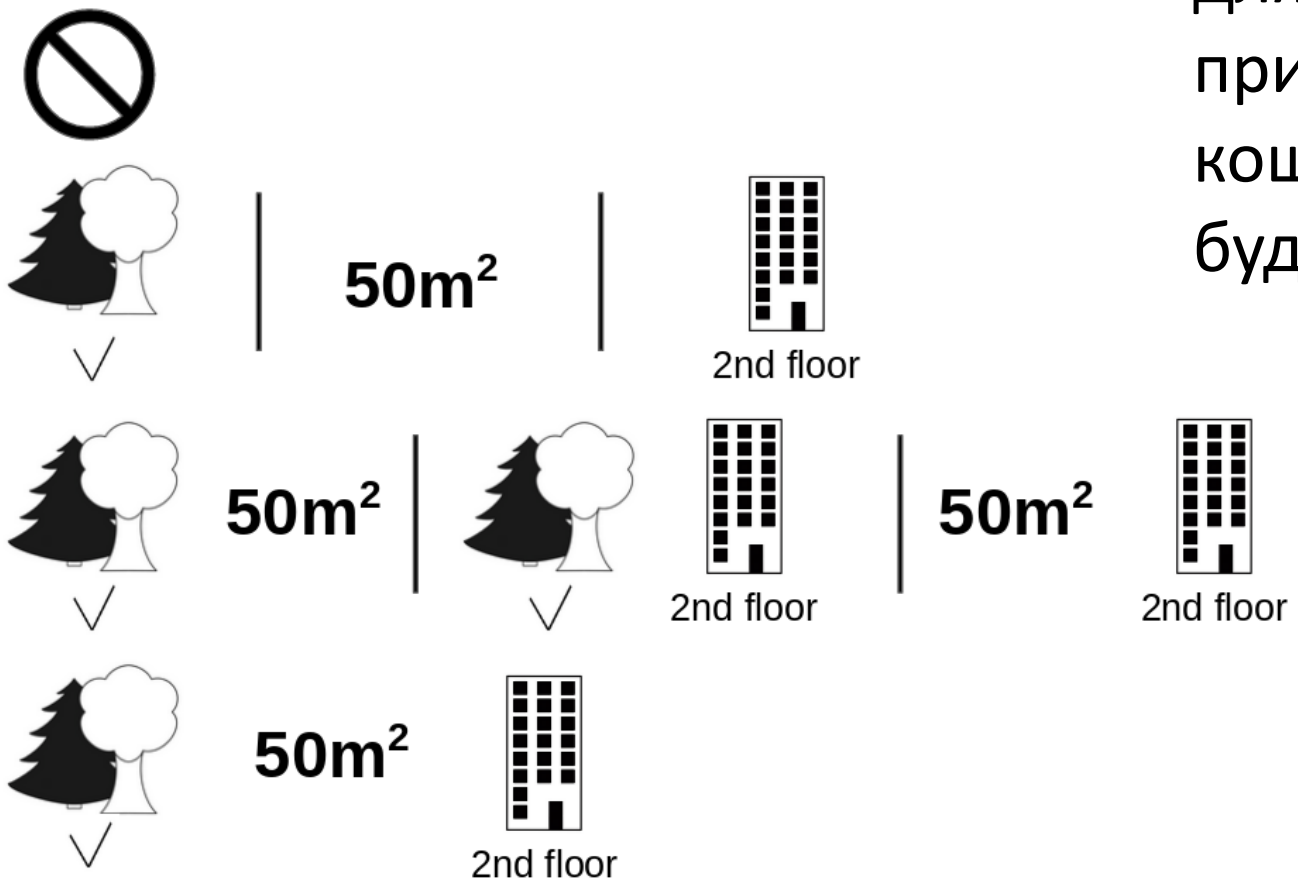
Узнаем для признака «запрещены ли кошки?» его вклад в цену.



- Выберем какое-то подмножество признаков, не включающее этот (парк и площадь)
- Для других признаков возьмём случайное значение из данных

$$val_x(S) = val_x(\{x_1, x_3\}) = \int_{\mathbb{R}} \int_{\mathbb{R}} \hat{f}\,(x_1, X_2, x_3, X_4) d\mathbb{P}_{X_2 X_4} - E_X(\hat{f}\,(X))$$

Вектор Шепли



Эту операцию надо повторить для всех подмножеств наших признаков без признака про кошек (в нашем случае их будет 8)

Вектор Шепли

Математическая формула вектора Шепли:

$$\Phi(v)_i = \sum_{K \ni i} \frac{(k-1)!(n-k)!}{n!}(v(K) - v(K \setminus i)),$$

Его также иногда вводят по-другому:

- зафиксируем порядок над игроками
- будем добавлять их по одному в этом порядке в команду
- при добавлении будем считать вклад нового игрока
- для каждого игрока усредним вклад по всем порядкам

Вектор Шепли

Свойства:

1) **Эффективность:** $\sum_{j=1}^{p} \phi_j = \hat{f}(x) - E_X(\hat{f}(X))$
2) **Симметричность:** одинаковые игроки получают одинаковый выигрыш
3) **Аксиома болвана:** если игрок никогда не приносит вклада, он получает 0
4) **Линейность:** $\Phi(v+w) = \Phi(v) + \Phi(w) \qquad \Phi(\alpha v) = \alpha\Phi(v)$

Доказано, что вектор Шепли – единственные значения, удовлетворяющие этим свойствам.

Вектор Шепли

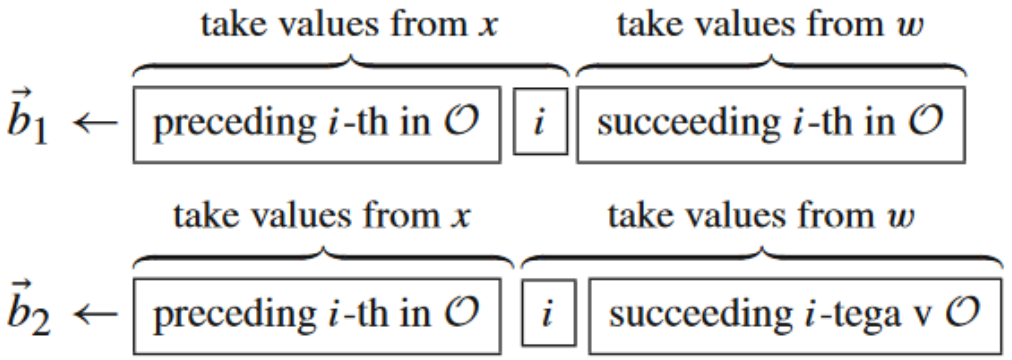
Приближённый алгоритм вычисления величин: усредним по M перестановкам.

Для j-ого признака повторим M раз:

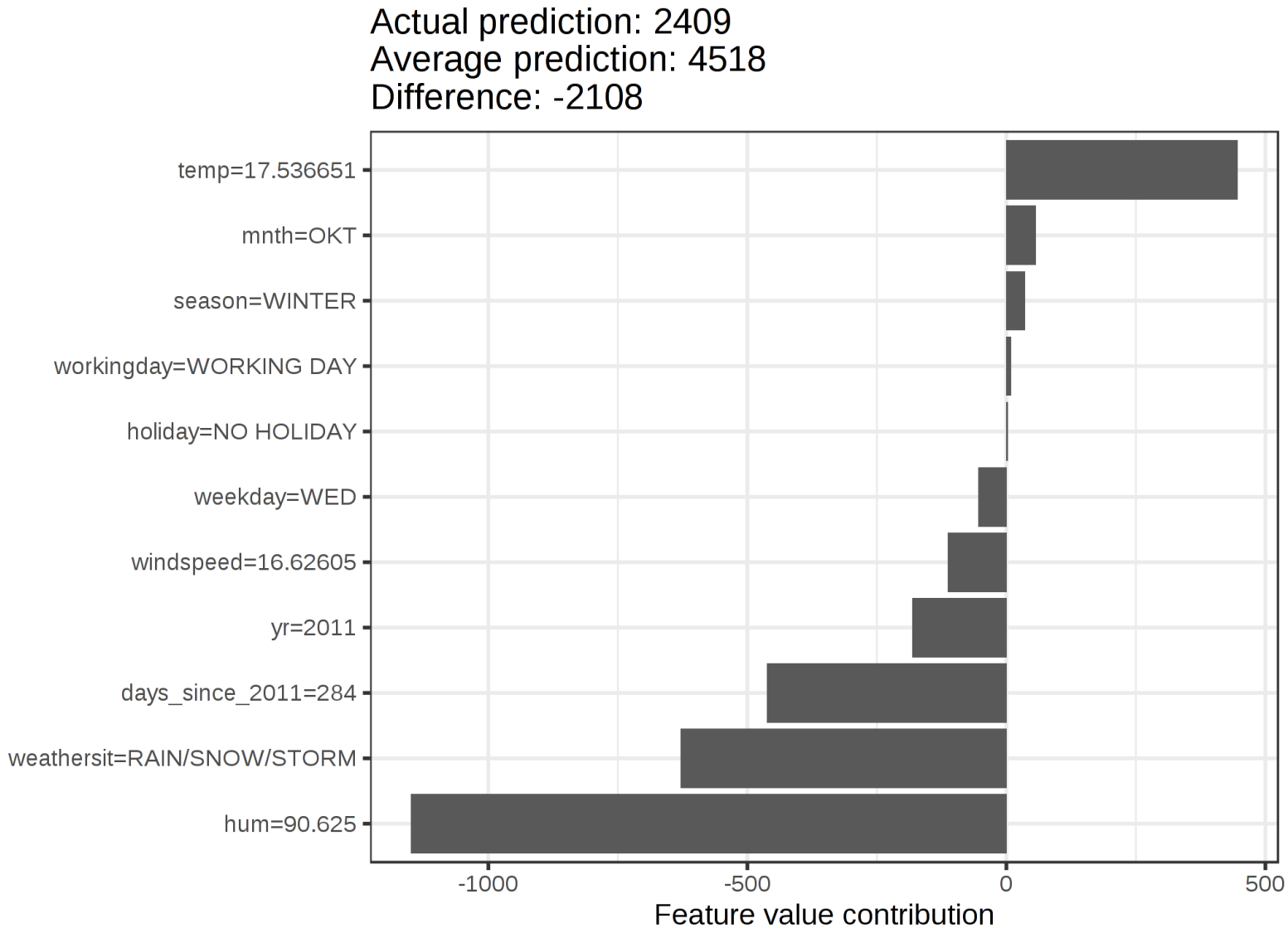
- Выберем произвольный объект w
- Сгенерируем случайную перестановку и переставим признаки в w и x
- Посчитаем $f(b_1) - f(b_2)$

В конце усредним эти разности.

Оценка несмещённая и состоятельная.



Вектор Шепли



Вектор Шепли

Преимущества:

- Гарантирует честное, математически обоснованное распределение важности признаков
- Можно сравнивать предсказание со средним по какому-нибудь подмножеству

Недостатки:

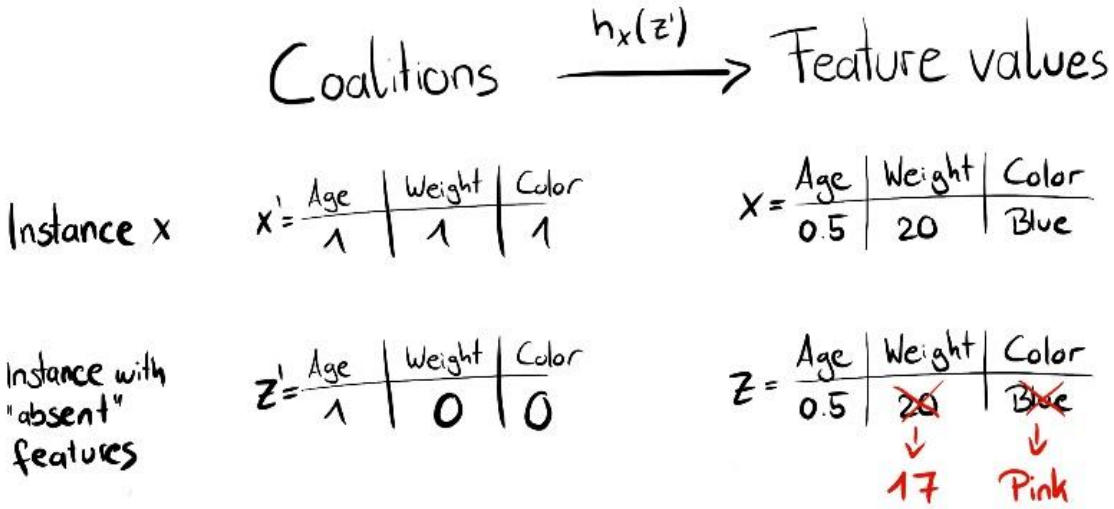
- Экспоненциальное время работы
- Не даёт формулы для локального поведения модели
- Нужны обучающие данные
- Возникает проблема, если признаки зависимы

SHAP (SHapley Additive exPlanations, 2017)

- Объединяет в себе два предыдущих метода
- Ищем простую модель в такой форме:

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j$$

- Для табличных данных:



SHAP

Свойства:

1) Local accuracy: $f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i \qquad \phi_0 = f(h_x(0))$

2) Missingness: $x'_i = 0 \implies \phi_i = 0$

3) Consistency (Согласованность):

$$f'_x(z') - f'_x(z' \setminus i) \geq f_x(z') - f_x(z' \setminus i)$$

$$\Downarrow$$

$$\phi_i(f', x) \geq \phi_i(f, x)$$

Существуют единственные коэффициенты, удовлетворяющие этим свойствам – значения Шепли.

SHAP

Задача, которую решает LIME:

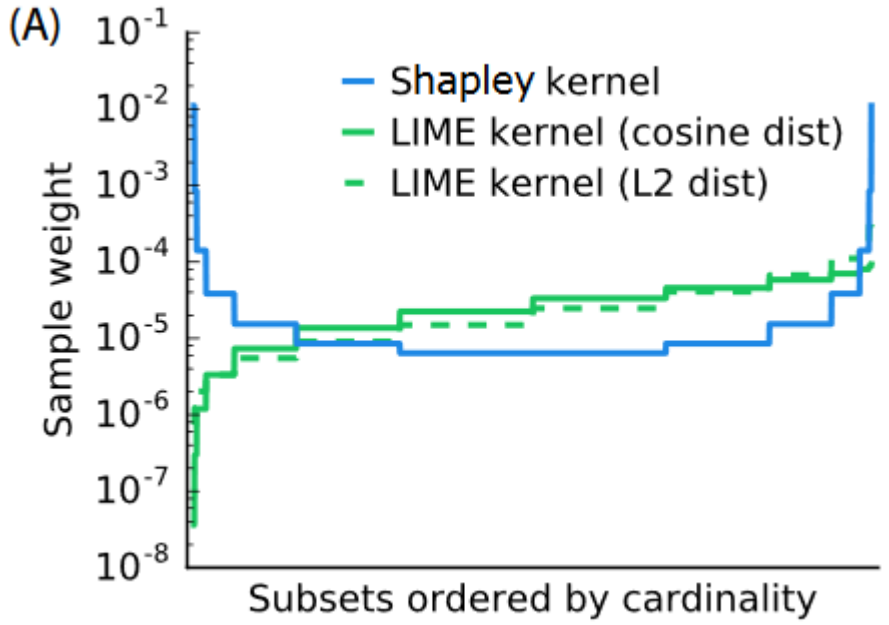
$$\text{explanation}(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

Для нахождения значений Шепли выберем такие функции:

$$\Omega(g) = 0,$$

$$\pi_{x'}(z') = \frac{(M-1)}{(M \; choose \; |z'|)|z'|(M-|z'|)},$$

$$L(f, g, \pi_{x'}) = \sum_{z' \in Z} \left[f(h_x(z')) - g(z')\right]^2 \pi_{x'}(z'),$$



SHAP

- Значения Шепли можно вычислить так:

$$\phi = (X^T W X)^{-1} X^T W y$$

X – матрица с 2^M строк и M столбцов

W – матрица с $\pi_x(\text{строка } X)$ на диагонали

y – значения сложной функции от строк X

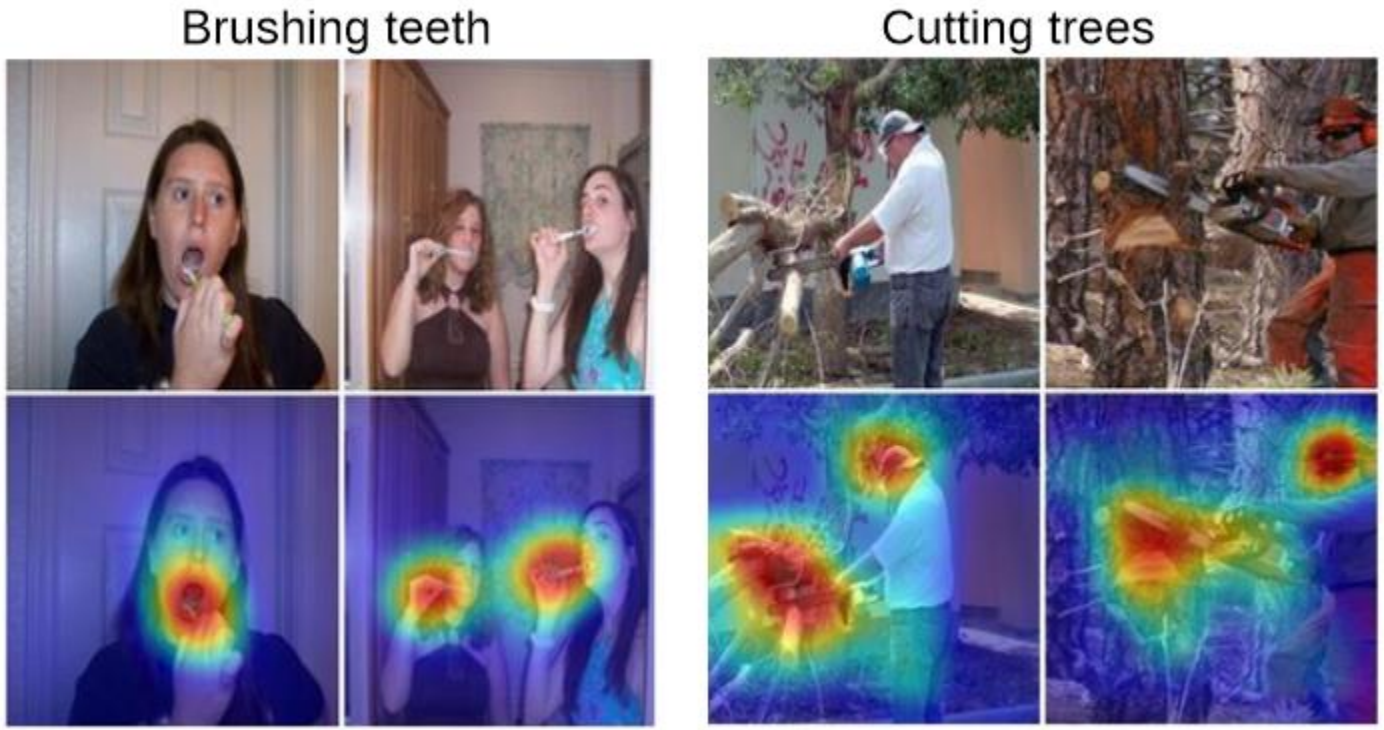
- Асимптотика: $O(M2^M)$
- Если сложная модель – дерево: $O(LD^2)$, где L – максимальное число листьев, D – глубина дерева.

SHAP

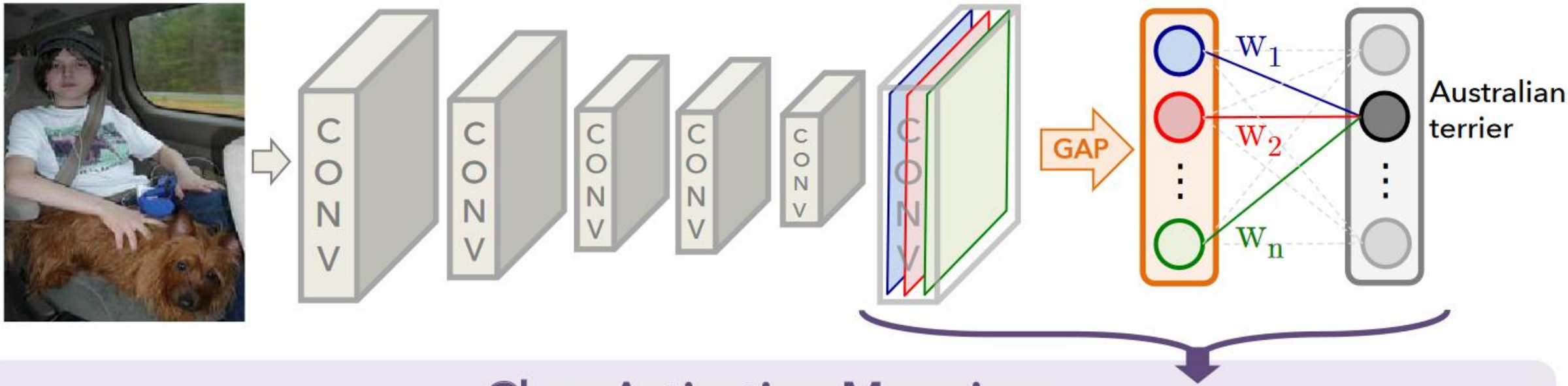

higher ⇄ lower
base value
output value
0.1979
0.2979
0.3979
0.4979
0.5979
0.70
0.7979
0.8979
0.9979
Passes = 399
Goals in PSO = 3
Saves = 1
Free Kicks = 18
Fouls Committed = 25
Corners = 6
On-Target = 7
Goal Scored = 2
Ball Possession % = 38
Offsides = 1
Blocked = 2
Distance Covered (Kms) = 148
sample order by similarity
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
52.34
47.34
42.34
37.34
32.34
27.34
22.34
17.34
12.34
7.34
model output

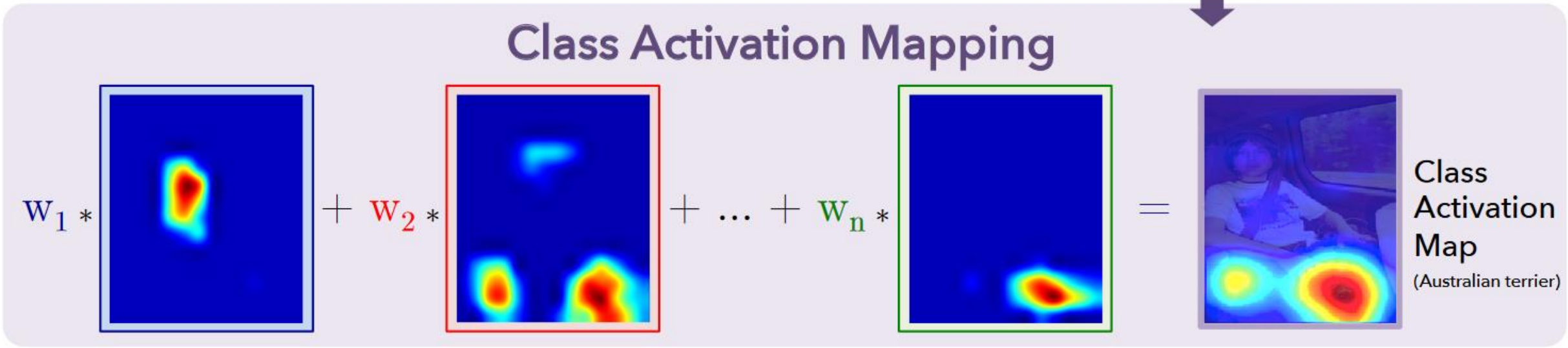
CAM (Class Activation Mapping, 2015)

Предлагает визуальные объяснения для решений, принятых CNN



CAM

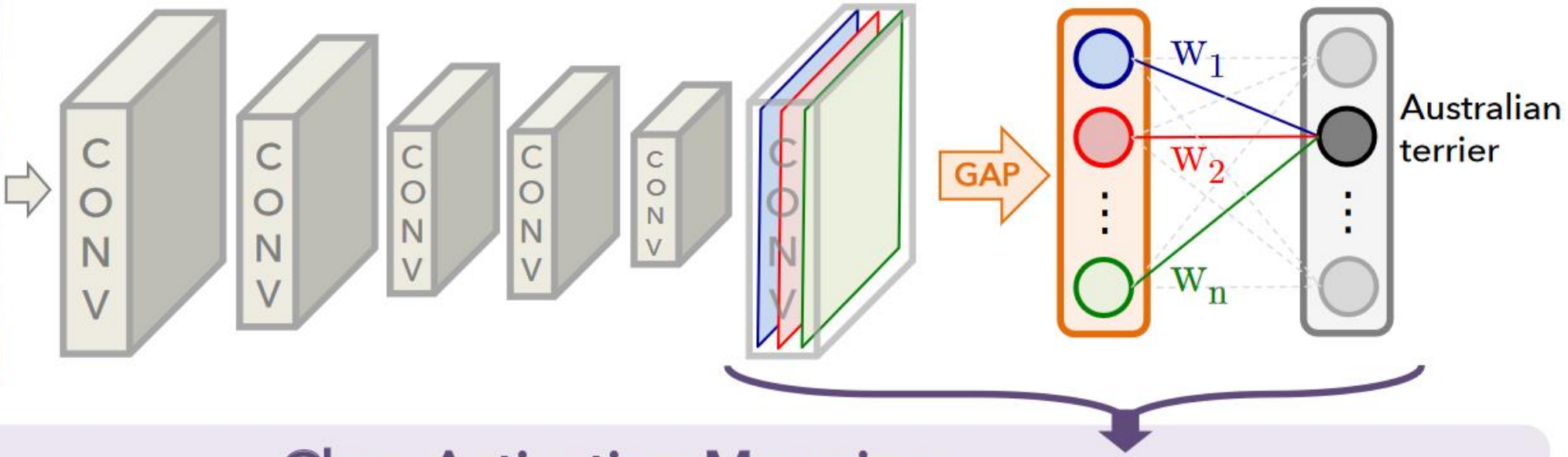

CONV
CONV
CONV
CONV
CONV
CONV
GAP
w_1
w_2
w_n
Australian terrier


Class Activation Mapping
w_1 *
+ w_2 *
+ ... + w_n *
=
Class Activation Map
(Australian terrier)

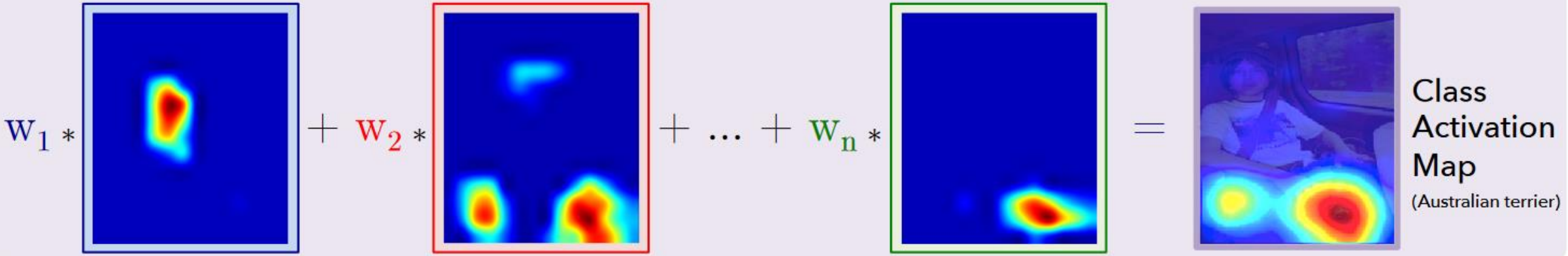
CAM

$$y^c = \sum_k \underbrace{w_k^c}_{\text{class feature weights}} \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{A_{ij}^k}_{\text{feature map}}$$

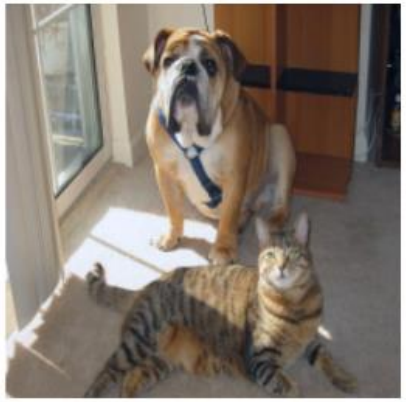
$$L_{\text{CAM}}^c = \underbrace{\sum_k w_k^c A^k}_{\text{linear combination}} .$$



Class Activation Mapping

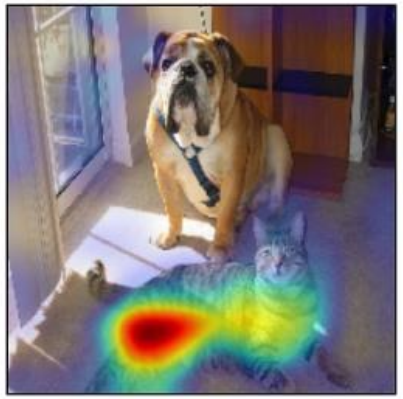


Grad-CAM (Gradient-weighted CAM, 2016)

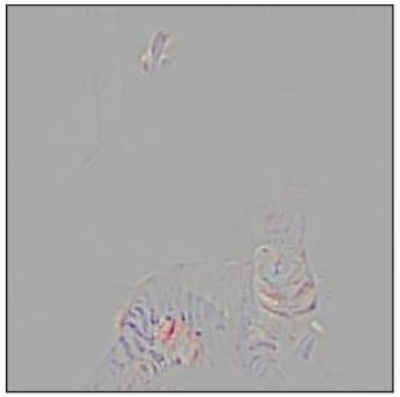


(a) Original Image

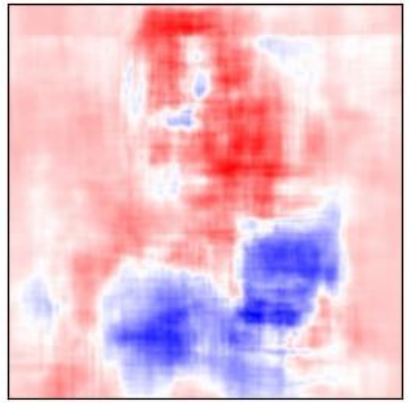
(b) Guided Backprop 'Cat'



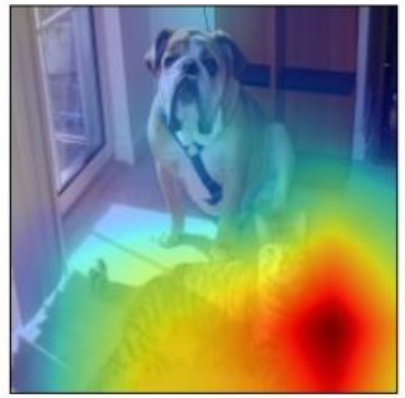
(c) Grad-CAM 'Cat'



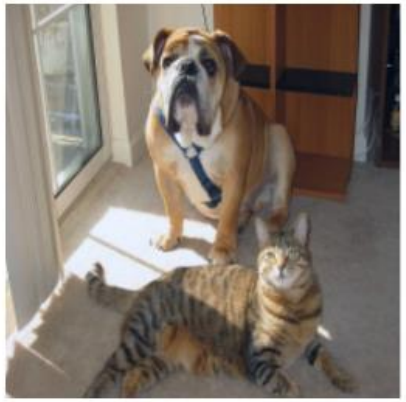
(d)Guided Grad-CAM 'Cat'



(e) Occlusion map 'Cat'



(f) ResNet Grad-CAM 'Cat'

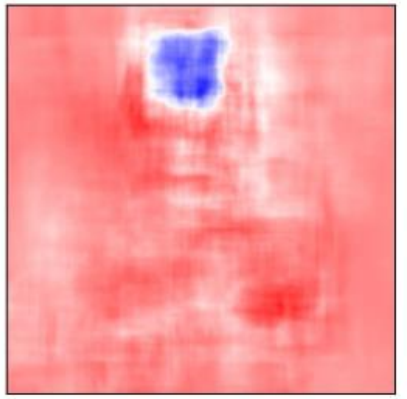


(g) Original Image

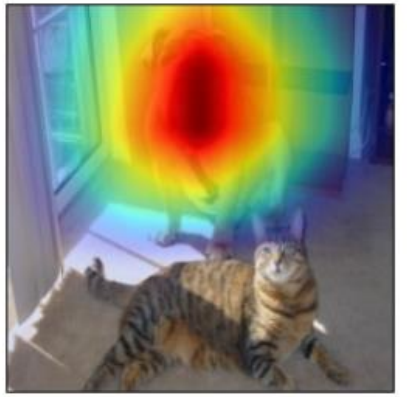
(h) Guided Backprop 'Dog'

(i) Grad-CAM 'Dog'

(j)Guided Grad-CAM 'Dog'



(k) Occlusion map 'Dog'

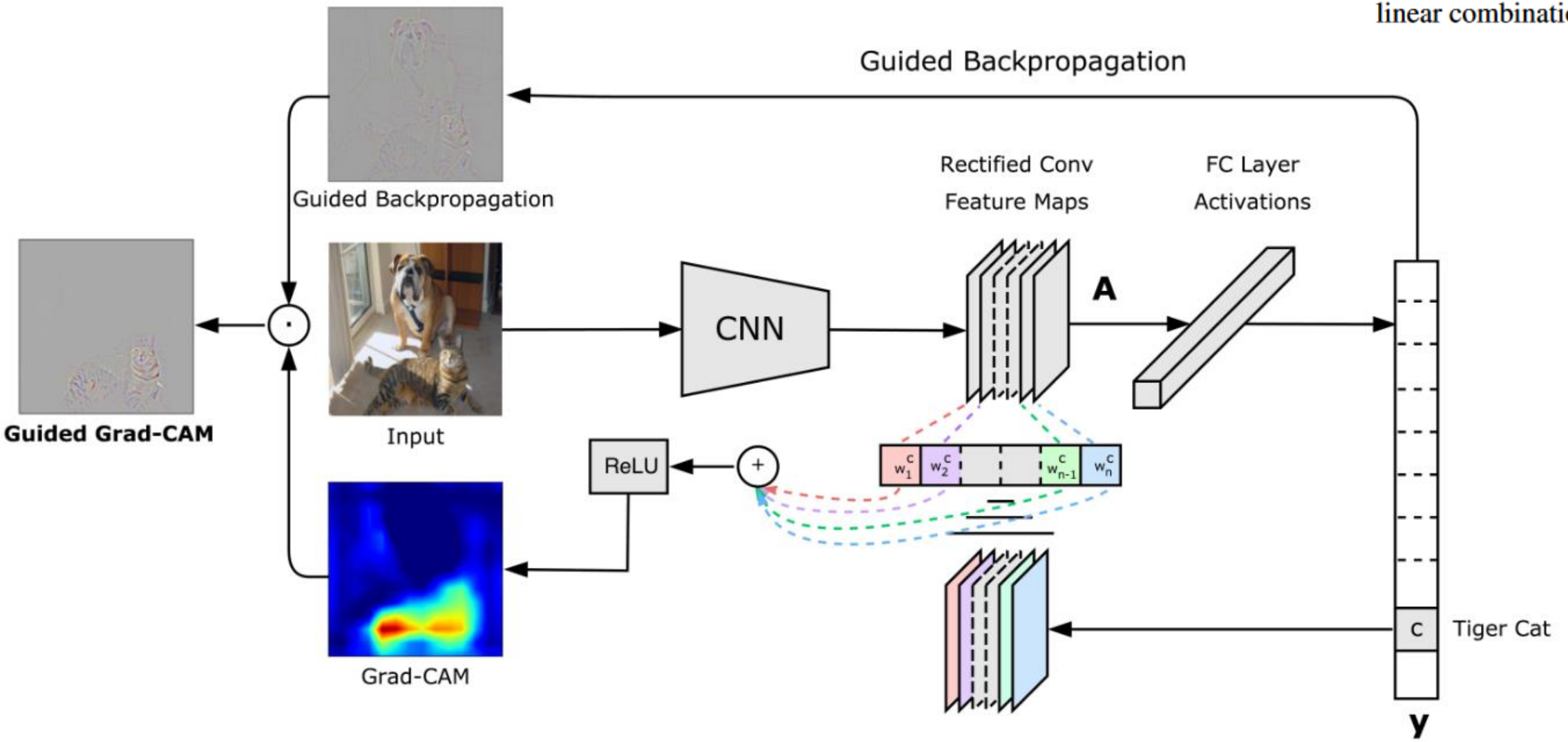


(l)ResNet Grad-CAM 'Dog'

Grad-CAM

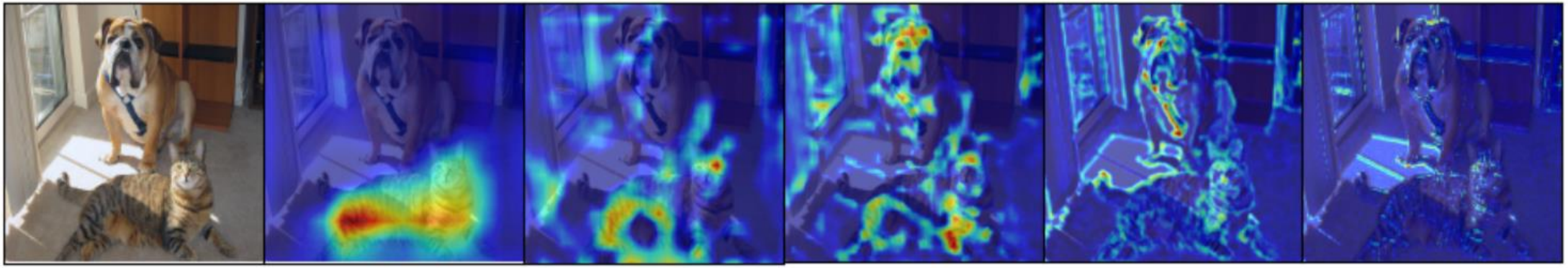
$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

$$L^c_{\text{Grad-CAM}} = ReLU\underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}}$$



Grad-CAM

Алгоритм лучше всего применять к самым глубоким (последним) слоям. С более близкими к началу он работает хуже.



Выводы

- LIME может быть полезен для локального приближения сложной модели, но он очень нестабилен.
- Значения вектора Шепли – математически доказанные решения нашей задачи, но считаются за экспоненциальное время.
- SHAP объединяет LIME и значения Шепли и имеет улучшенную реализацию для случая, в котором исходная модель – дерево.

- Grad-CAM – работает только для объяснения классификации картинок с помощью CNN, зато работает со всеми архитектурами.
- CAM – его частный случай.

Источники

Книга по интерпретируемости: https://christophm.github.io/interpretable-ml-book/

Статьи:

- Про LIME: https://arxiv.org/pdf/1602.04938.pdf
- Про вектор Шепли для интерпретации: https://sci-hub.do/10.1007/s10115-013-0679-x
- Про SHAP: https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf
- Про CAM: https://arxiv.org/pdf/1512.04150.pdf
- Про Grad-CAM: https://arxiv.org/pdf/1610.02391.pdf

Остальные источники:

- https://www.cs.princeton.edu/courses/archive/spring20/cos598C/lectures/lec20-interpretability.pdf
- https://dou.ua/lenta/articles/interpreting-machine-learning-1/
- https://deepmind.com/blog/article/alphago-zero-starting-scratch
- https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%BA%D1%82%D0%BE%D1%80_%D0%A8%D0%B5%D0%BF%D0%BB%D0%B8
- https://www.kaggle.com/dansbecker/shap-values
- https://github.com/slundberg/shap