DLS

Базовые методы оценки важности признаков в ML

Садиех Сабрина, HSE, XAI researcher

План занятия

1. Зачем нужно применять методы для интерпретации;
2. Графические методы представления влияния признака и их количественные характеристики;
3. Метод SHAP;
4. Метод LIME;
5. Анализ взаимодействия признаков.

Что такое Explainable AI (aka XAI)?

Объяснимый искусственный интеллект (XAI) — область, охватывающая фреймворки и методы, обеспечивающие возможность объяснять прогнозы, сделанные DNN или ML моделью.

Что такое Explainable AI (aka XAI)?

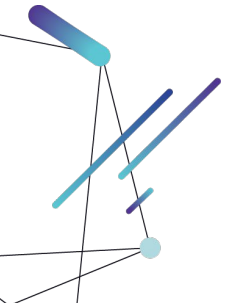
Объяснимый искусственный интеллект (XAI) — область, охватывающая фреймворки и методы, обеспечивающие возможность объяснять прогнозы, сделанные DNN или ML моделью.

- **Хотим понять:**

  1. Как модель делает конкретный прогноз (какие компоненты модели отвечают за этот прогноз)
  2. Почему модель делает именно этот прогноз (какие признаки в input или train data за это отвечают)

- **Можем использовать:**

  1. В деббагинге модели;
  2. Для повышения доверия (бизнеса/пользователей);
  3. В науке;



Историческая справка

1. 1970-1980-е — Expert Systems
- правила IF-THEN, прозрачная логика, легко объяснять

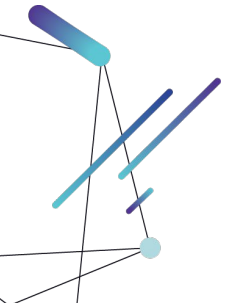
2. 1980-1990-е — Чёрные ящики (ML)
- нейронные сети и ансамбли, интерпретируемость теряется
- попытки делать модели с интерпретируемой структурой (GLM, GAM, MoE)

3. 1990-2000-е — Первая волна методов интерпретации
- графические методы важности признаков для ML моделей
- использование линейных/древесных моделей как компонент сетей (GAMI-NET, Neural Tree), статистики для анализа взаимодействий

Историческая справка

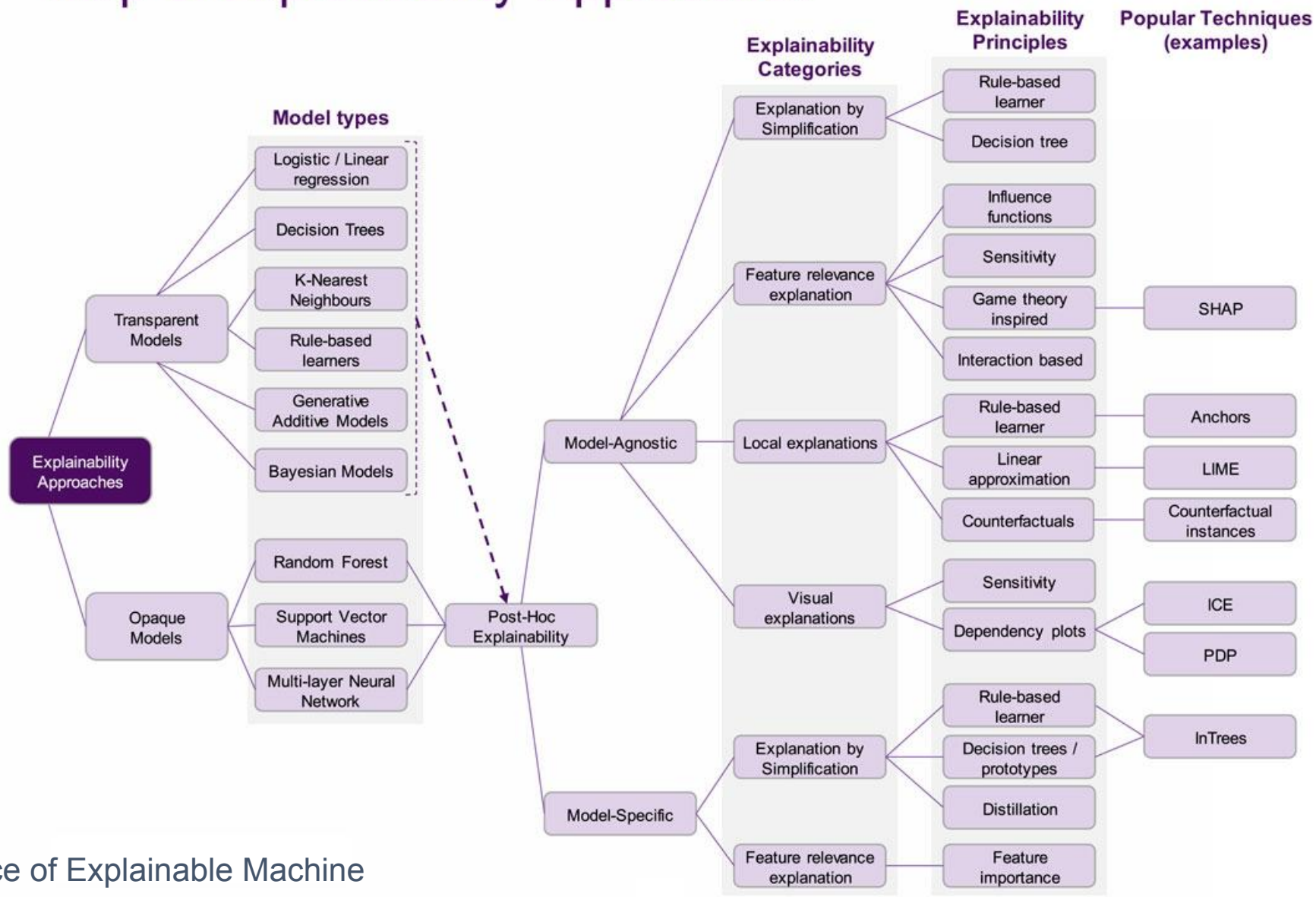
4. 2016–2018 — Бум XAI
- LIME, SHAP, Integrated Gradients
- рост интереса из-за «чёрных ящиков» deep learning

5. 2020-е — XAI as model biology и законодательство за XAI
- появление механистической интерпретируемости
- интеграция задачи объяснения в нормативные требования (GDPR, AI Act)



ТАКСОНОМИЯ

Map of Explainability Approaches



Principles and Practice of Explainable Machine Learning Vaishak Belle, Ioannis Papantonis

Почему появлялись и появляются новые методы

Рост сложности моделей
- от деревьев к глубоким нейронным сетям и LLM → старые методы перестают работать/быть устойчивыми

Проблема “кому объяснять”
- необходимость объяснять прогнозы в медицине, финансах и пр. — разный уровень пользователя — разные объяснения

Cost problem
- нужны дешевые и быстрые методы, особенно для больших моделей-

Интерес исследователей и адаптация к разным задачам
- понять внутренние представления модели (интерпретация как наука)
- мультимодальные модели, генеративные модели → нужны специальные подходы

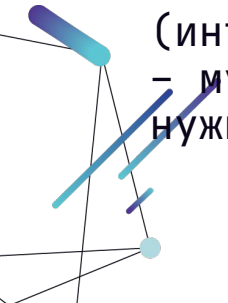
Почему появлялись и появляются новые методы

Рост сложности моделей
- от деревьев к глубоким нейронным сетям и LLM → старые методы перестают работать/быть устойчивыми

Проблема “кому объяснять”
- необходимость объяснять прогнозы в медицине, финансах, гос.секторе — разный уровень пользователя — разные объяснения

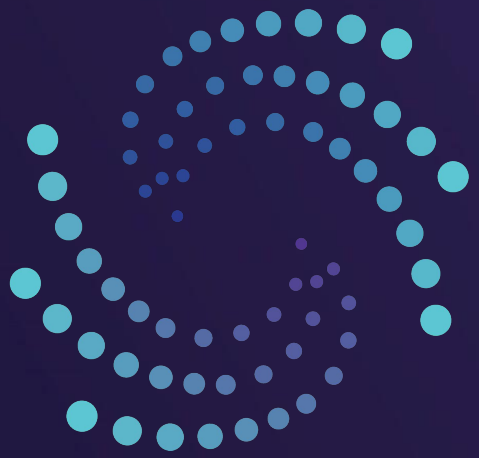
Cost problem
- нужны дешевые и быстрые методы, особенно для больших моделей-

Интерес исследователей и адаптация к разным задачам
- понять внутренние представления модели (интерпретация как наука)
- мультимодальные модели, генеративные модели → нужны специальные подходы



Можно ли выделить необходимый и достаточный pipeline для XAI?

Да, этим мы и займемся.



Deep
Learning
School

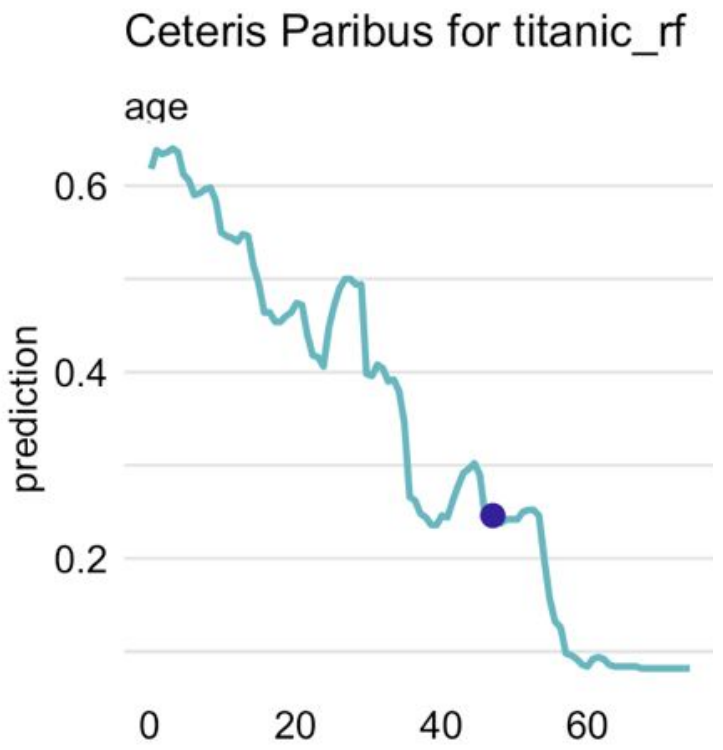
Базовые методы интерпретации


Deep
Learning
School
Графический
анализ
изменений в
прогнозе

CPO: Ceteris Paribus Oscillations

Ceteris Paribus Oscillations — стабильность модели через “Ceteris Paribus” профили — от лат. при прочих неизменных.

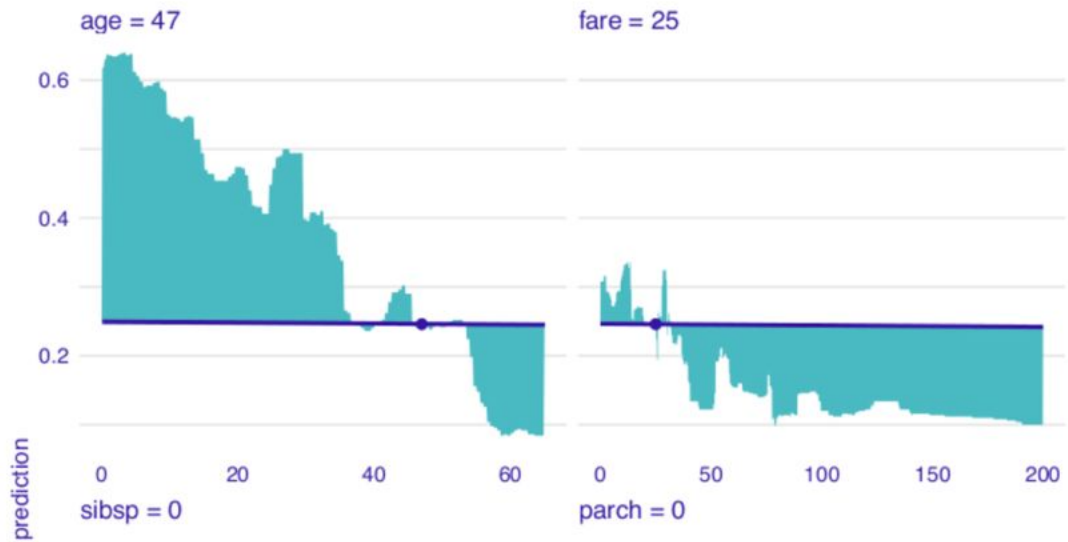
Вопрос: насколько предсказания модели меняются при изменении отдельного признака j, фиксируя остальные (отсюда и название).



Реализован в: [pyCeterisParibus](), [dalex]()

CPO. Ceteris Paribus Oscillations: построение

Рассмотрим объект x_i и его признак j. Зафиксируем прогноз модели $f_0(x_i)$ — прогноз при данном значении признака j. Как будет меняться прогноз модели, если мы будем "расшатывать" признак j?



CPO. Ceteris Paribus Oscillations: построение

Далее:

1. Зафиксируем все признаки объекта x_i, кроме j;
2. Определим упорядоченный диапазон конкретных значений признака $j_i, j_2, \ldots, j_m$;
3. Начнем изменять значения j в диапазоне $\{x_j^1, x_j^2, \ldots, x_j^m\}$, делаем прогноз при каждом значении признака j $f(x_i^{j_k})$;
4. Строим график зависимости:

$$CPO(x_i, j) : j \mapsto f(x_i^{j_k})$$

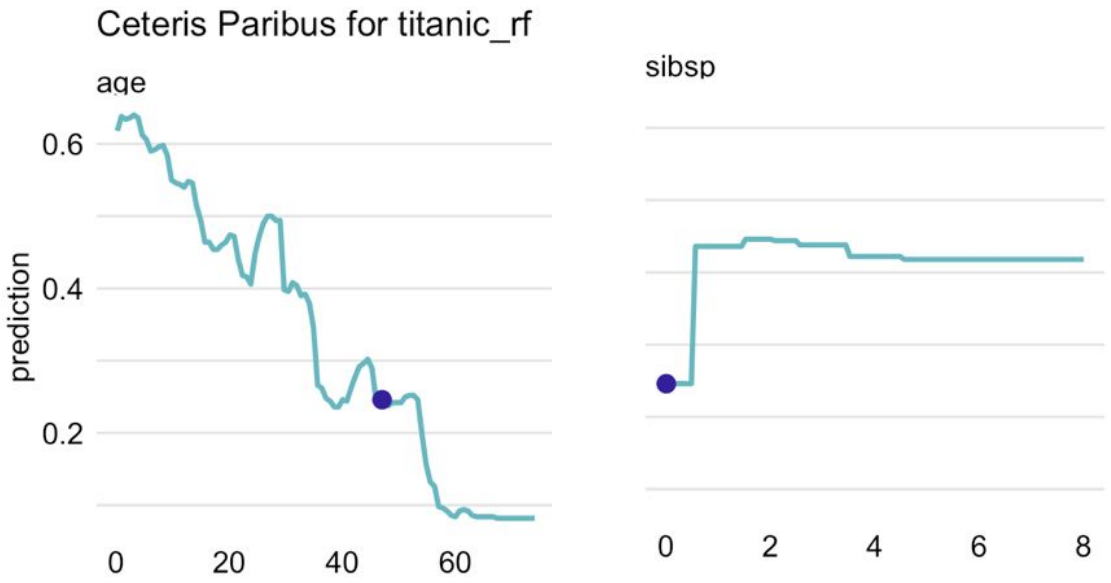
CPO. Ceteris Paribus Oscillations: интерпретация

1. Высокий CPO:

- Признак j сильно влияет на предсказания модели для данного объекта.
- Возможна нестабильность модели в этом диапазоне.

2. Низкий CPO:

- Модель практически не реагирует на изменения j.
- Признак может быть малозначимым для прогноза на этом объекте.



Ceteris-paribus профили для пассажира Генри и модели случайного леса titanic_rf. [dalex]

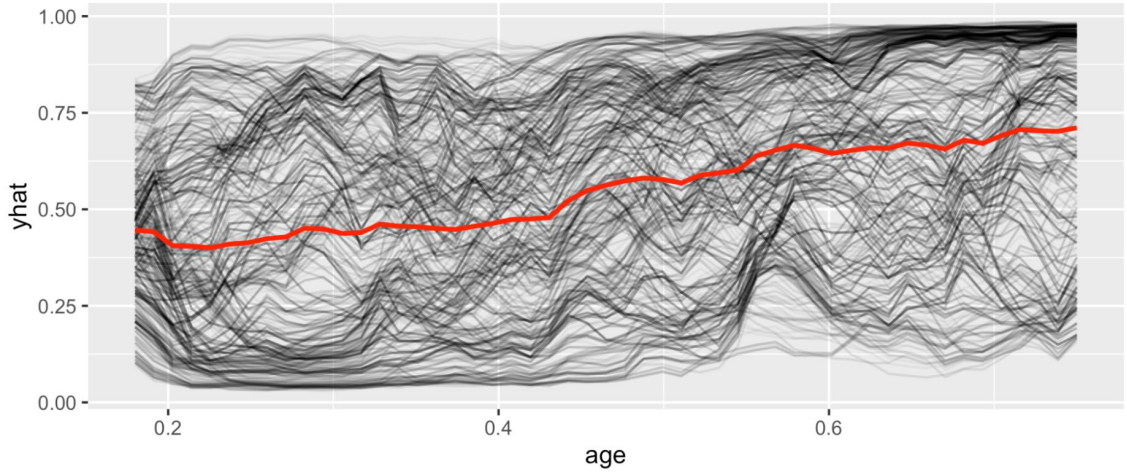
CPO. Ceteris Paribus Oscillations: практические нюансы

- Требует вычисления профилей для каждого объекта
- Отсутствует обобщение
- Не эффективен для интерпретации при нелинейных взаимодействиях признаков

Individual Conditional Expectation (ICE)

ICE — визуализируют поведение модели для *нескольких* объектов, показывая, как предсказания изменяются при варьировании одного признака, когда остальные зафиксированы.

Вопрос: насколько предсказания модели меняются при варьировании отдельного признака j на **группе** объектов, фиксируя остальные (отсюда и названия).



Individual Conditional Expectation (ICE): Построение

Пусть признак j имеет m уникальных значений.

1. Рассматриваем исходный датасет X и дублируем его m раз:

   $$X'_1, X'_2, \ldots, X'_m$$

   (как бы фиксируем все признаки, кроме одного)
2. Рассчитываем прогноз модели $f(X_i)$
3. На одном графике отображаются линии для всех объектов, демонстрируя их индивидуальные профили.

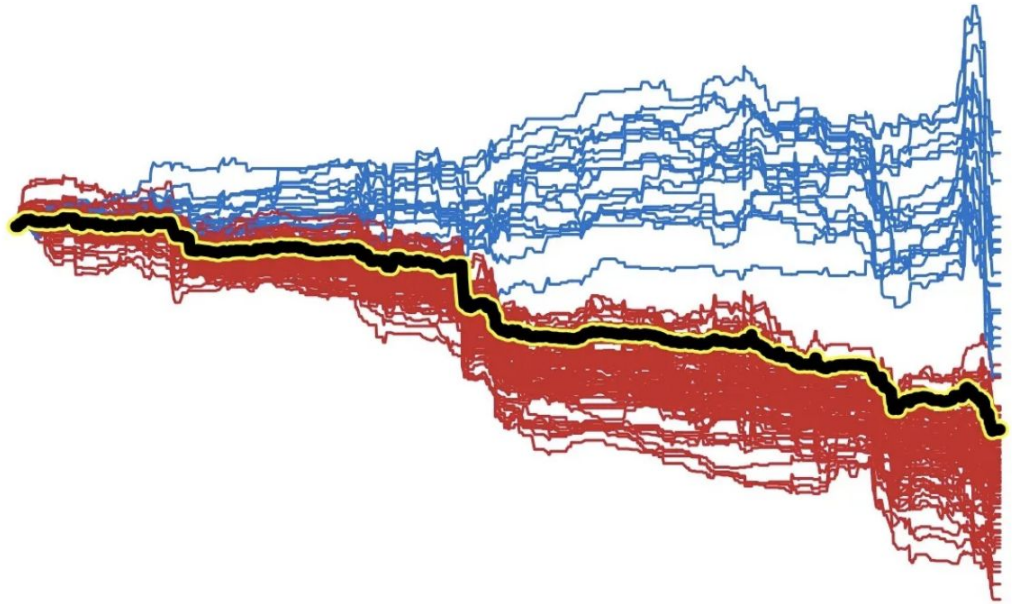
ICE. Individual Conditional Expectation: интерпретация

Полезно красить линии под группы данных.

Разные наклоны линий: Взаимодействие модели с объектами или значимая нелинейность.

Сходящиеся линии: Признак действует одинаково для всех объектов.

Шум: Разнообразные профили могут указывать на переобучение или недообучение модели.



car_age and car_type

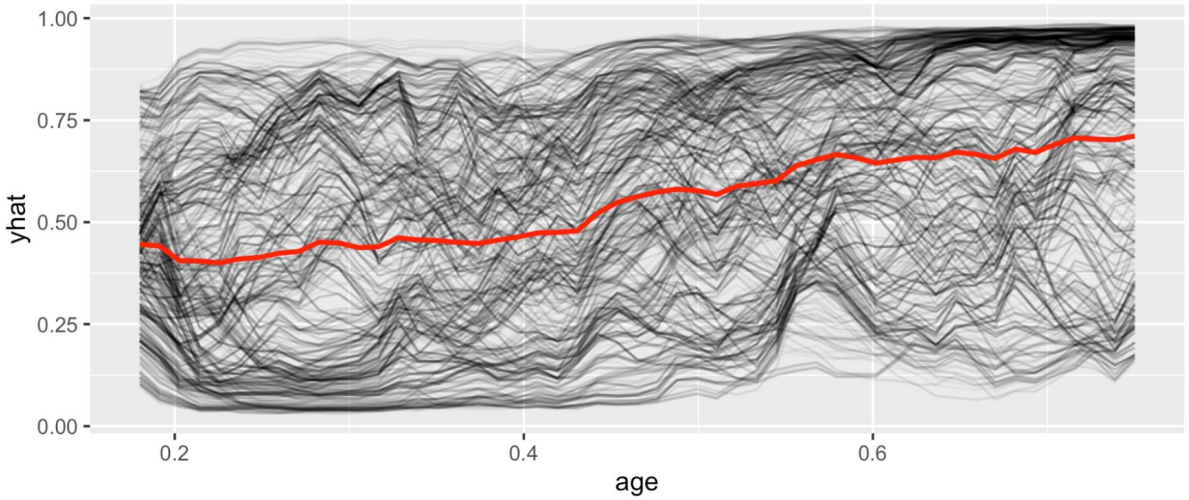
ICE. Individual Conditional Expectation: практические нюансы

- Чуть более оптимальная версия CPO;
- Встроен в sklearn (и легко реализовать руками).
- Отсутствует обобщение — анализ каждого признака отдельно.
- Неприятен для восприятия и донесения информации при нелинейных взаимодействиях и на слишком большом количестве объектов.

Partial Dependence Plot (PDP)

PDP — показывает, как среднее предсказание модели меняется в зависимости от значений одного или нескольких признаков.

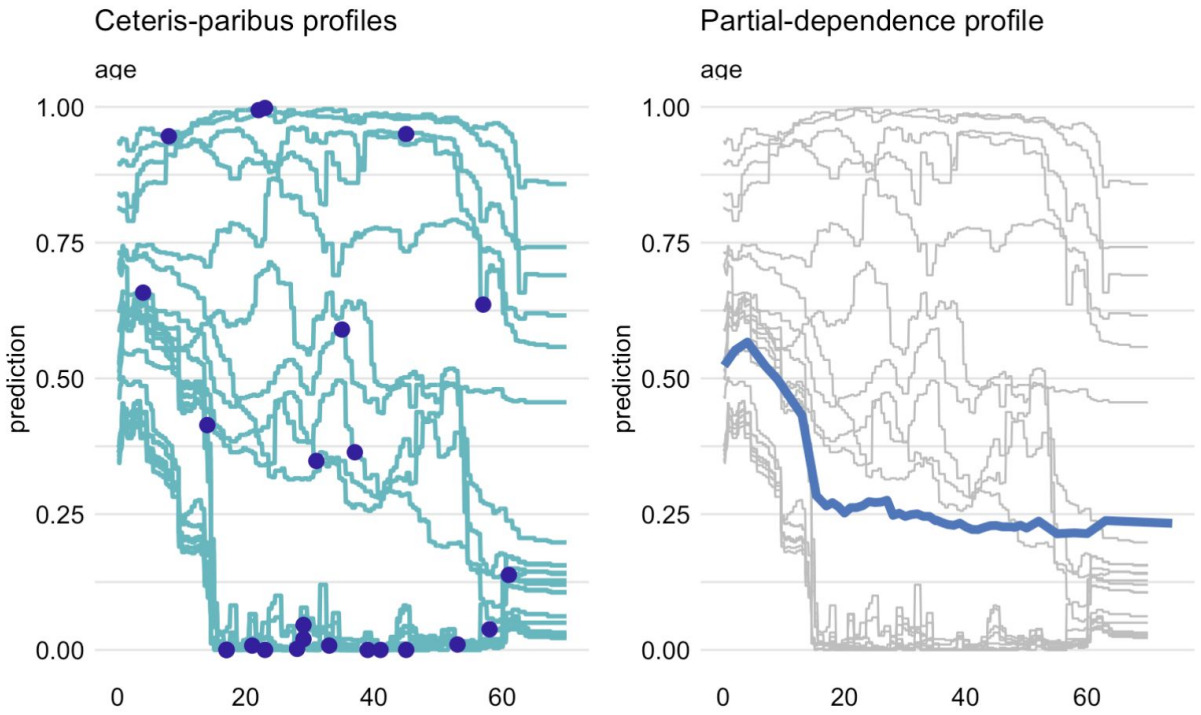
Глобальная интерпретация, усредняющая поведение модели по всему датасету.



Матожидание по ICE!

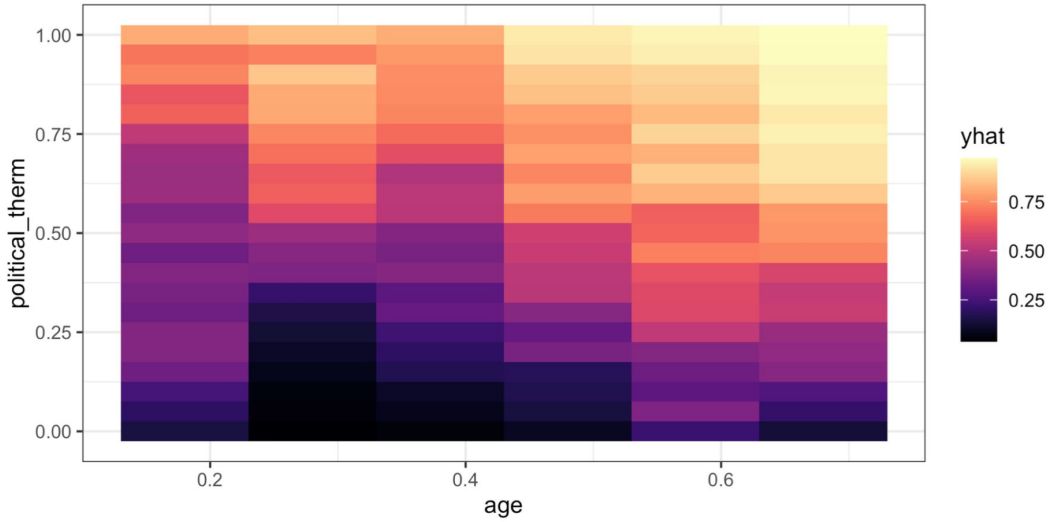
Partial Dependence Plot (PDP): интерпретация

- **Плоская линия:** Признак не влияет на предсказание.
- **Наклон и кривизна:** характер зависимости предсказаний от признака. Например, растущий PDP указывает на положительное влияние признака.



PDP. Partial Dependence Plot: практичесие нюансы

- Убирает из виду индивидуальное влияние ⟹ ICE, как пара
- Игнорирует взаимодействия признаков (в одномерном построении, можно использовать двумерное)
- Не валидный метод для линейных моделей — лучше использовать анализ весов.
- Встроен в sklearn и реализован в pdpbox



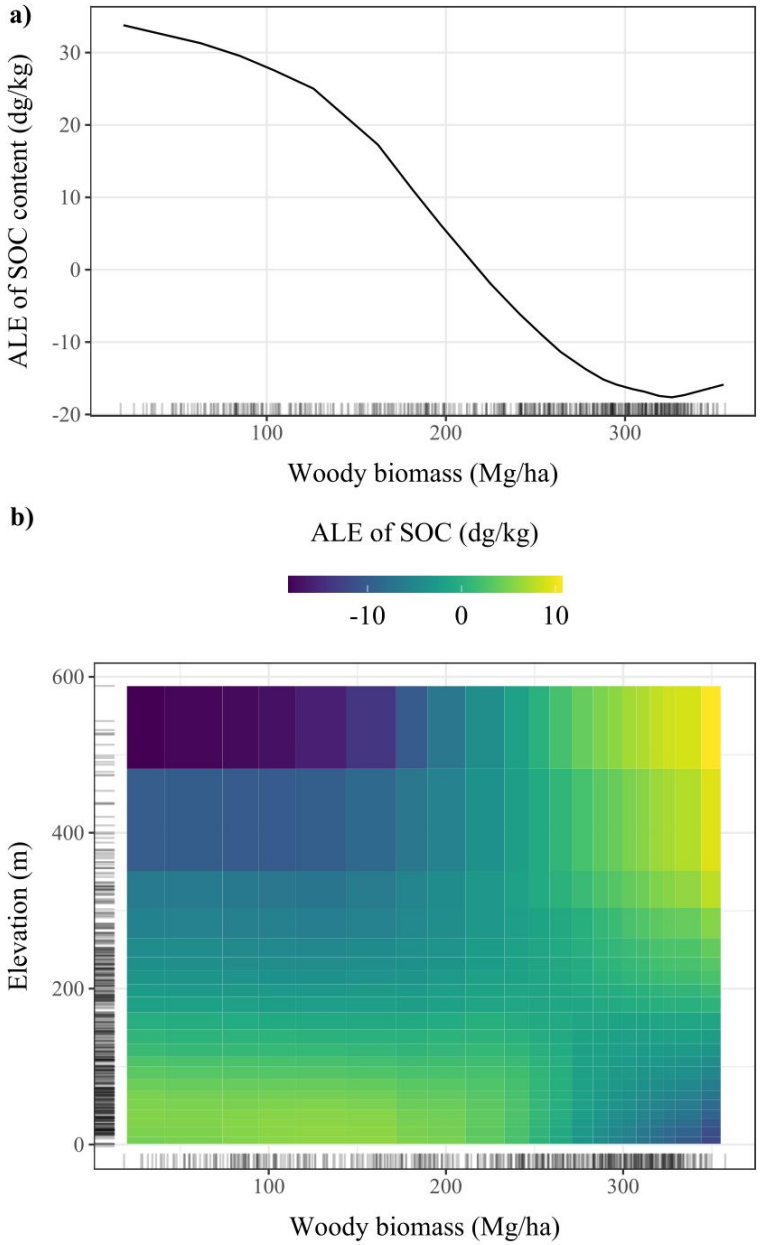
Источник

Accumulated Local Effects (ALE)

ALE — оценивает влияние признаков на предсказания модели, но устраняет влияние коррелированных признаков и взаимодействий.

Это достигается путем **локального анализа** изменений предсказания.

Источник



ALE: построение

пусть

- $f(x)$: предсказания модели.
- F: интересующие нас признаки
- S_j: все остальные признаки, кроме F.
- Разделение значений F на интервалы: $[x_F^{(k)}, x_F^{(k+1)}]$, где $k = 1, 2, \ldots, K-1$.
- K — общее число интервалов
- n число наблюдений на наборе, для которого строим объяснение

ALE: построение

Тогда ALE:

$$f_{F,ALE} = \sum_{k=1}^{K} \frac{1}{|x \in N_j(k)|} \sum_{|x \in N_j(k)|} [f(x_j^{(k+1)}, S_j) - f(x_j^{(k)}, S_j)]$$

где F интересующий нас признак, $N_j(k)$ количество наблюдений, попавших в интервал с номером k, а значения в скобках — прогнозы модели при замене признака на нижнюю и верхнюю границы интервала соответственно.

ALE: построение

1. Выбрать интересующий признак F
2. Разделить диапазон изменения признака на интервалы $[x_F^{(k)}, x_F^{(k+1)}]$, где $k = 1, 2, \ldots, K-1$.
3. Для каждого интервала:

   а) Заменить признак на значение нижней границы интервала и рассчитать прогнозы модели $f(x_F^{(k)}, S_j)$

   б) Заменить признак на значение верхней границы интервала и рассчитать прогнозы модели $f(x_F^{(k+1)}, S_j)$

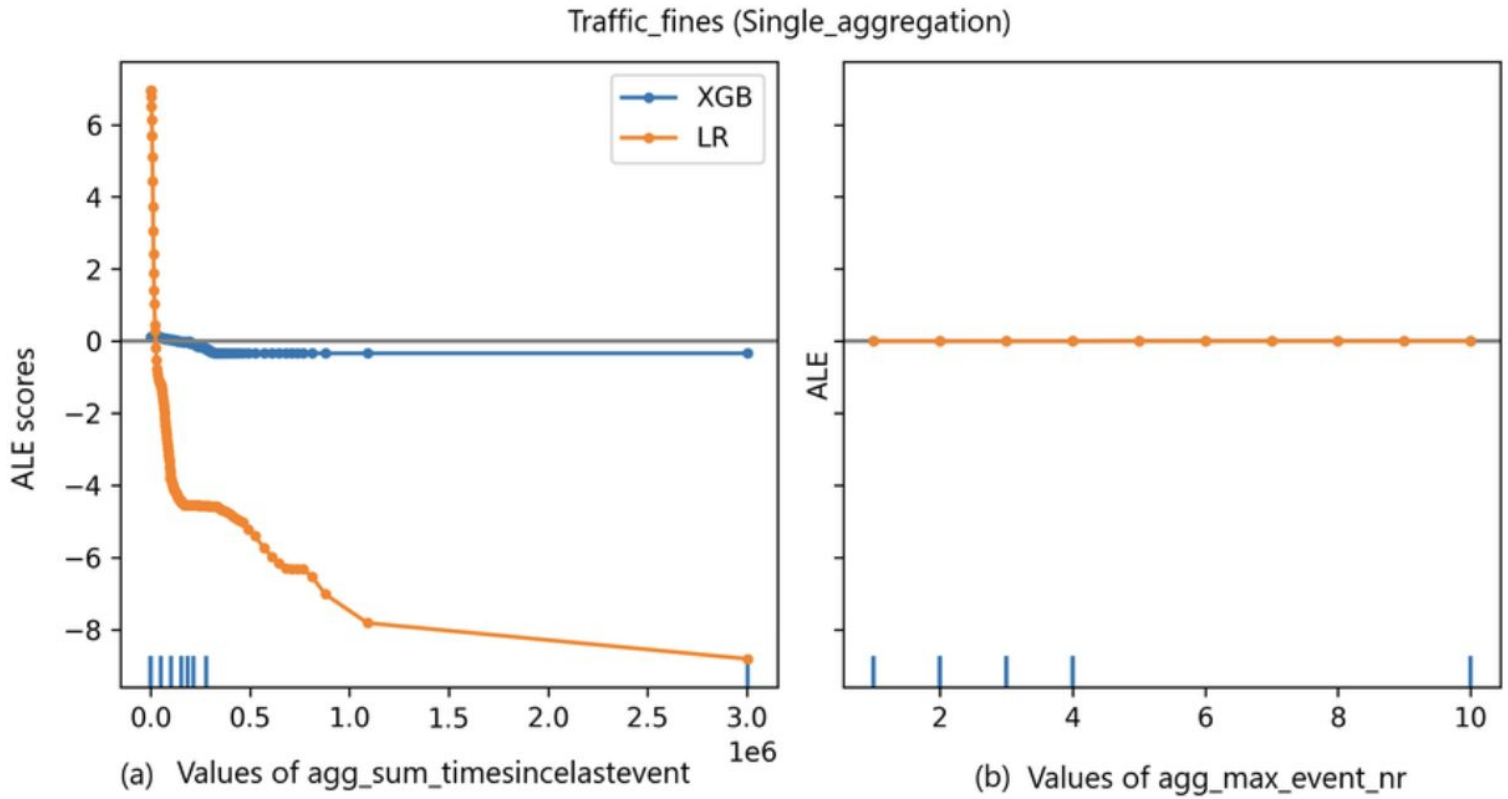
   в) Усреднить разницу в прогнозах между (а) и (б) согласно количеству объектов в интервале.
4. Суммировать эффекты по всем интервалам и далее усреднить полученный эффект по наблюдениям:

$$f_{F,centeredALE} = f_{F,ALE} - \frac{1}{n}\sum_{k=1}^{K} N_j(k) f_{F,ALE}$$

Шаги "а", "б" — являются компонентами **эффекта,** а шаг "в" — **локальности.**

Accumulated Local Effects (ALE): интерпретация

- Показывает среднюю динамику изменения предсказания модели.
- В отличие от PDP, не искажает зависимости, вызванные корреляцией признаков.

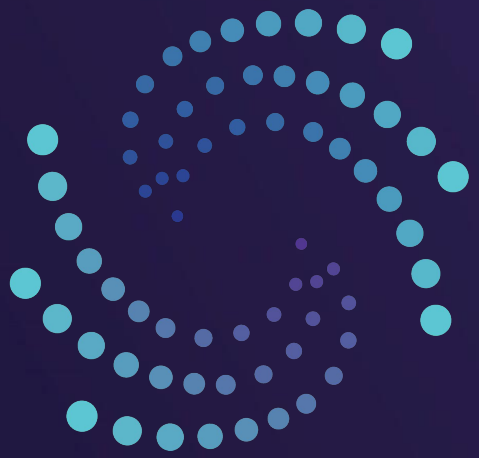


Accumulated Local Effects (ALE): практические нюансы

- важно аккуратно выбрать число интервалов (bins)и анализировать их вместе с доверительными интервалами

  слишком мало → сглаживание, потеря деталей
  слишком много → шум

- масштаб признака влияет на график → стоит нормировать
- для категориальных признаков не валиден
- из-за линейности внутри интервалов может искажать нелинейности



Deep
Learning
School

Анализ важности признаков при помощи статистик

PDP-based feature importance

Наивный подход получения важности признака — оценка дисперсии прогноза при изменении признака

Чувствительна к диапазону значений (необходимо масштабирование)

Для категориальных признаков нормировка — эвристика

Не учитывает взаимодействия при оценке

$$\sqrt{\left(\frac{1}{k-1}\sum_{i=1}^{k}\left(f(x_{1i}-\frac{1}{k}\sum_{i=1}^{k}f(x_{1i})\right)^2\right)}$$

$$\frac{max_i(f(x_1 i)) - min_i(f(x_1 i))}{4}$$

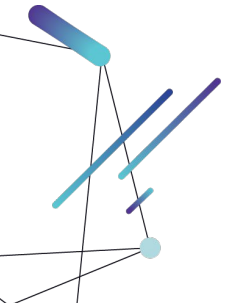
categorical and continuous pdp-based ft importance
1i — уникальные наблюдения, содержащие признак i

Permutation importances

Permutation importances — оценка потери качества модели при случайной перестановке значений признака.

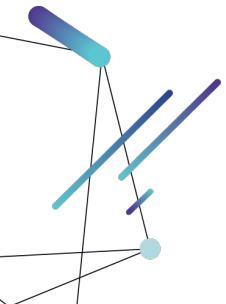
Чем сильнее амплитуда изменений метрики качества или прогноза — тем важнее признак

Height at age 20 (cm)	Height at age 10 (cm)
182	155
175	147
...	...
156	142
153	130



Permutation importances

Пусть $a(x)$ модель, обученная на множестве признаков $F = f_1, f_2, ..., f_n$, e_{orig} – ошибка модели на тестовом набор
данных. Зафиксируем признак f_i.



Permutation importances

Для этого признака:

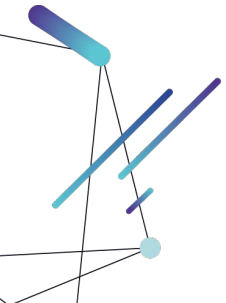
- случайным образом переставим его значения в тестовом наборе данных;
- вычислим прогноз модели на тестовых данных с перестановкой;
- оценим ошибку модели на наборе данных с перетасовкой e_{perm}
- оценим важность признака как разность $e_{orig} - e_{perm}$ или частное $\frac{e_{perm}}{e_{orig}}$

Данный алгоритм повторяется со всеми признаками и далее они сортируются по убыванию важности.

Таким образом важность признака $f_i = e_{orig} - \frac{1}{K}\sum_{k=1}^{K} e_{perm,k}$, где K – число перестановок.

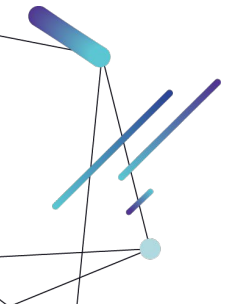
Permutation importances. Практические особенности

- метод, применимый к любым моделям
- читывает нелинейности и взаимодействия признаков
- даёт глобальную важность — для всех примеров из датасета сразу
- не справляется с корреляцией признаков важность распределяется между ними
- вычислительно дорог — сложность $O(p*n)$, где p — число признаков, n — число перестановок
- чувствителен к случайности — важно использовать несколько повторов и доверительные интервалы
- использовать на валидационной/тестовой выборке, а не на train



Permutation importances. Практические особенности

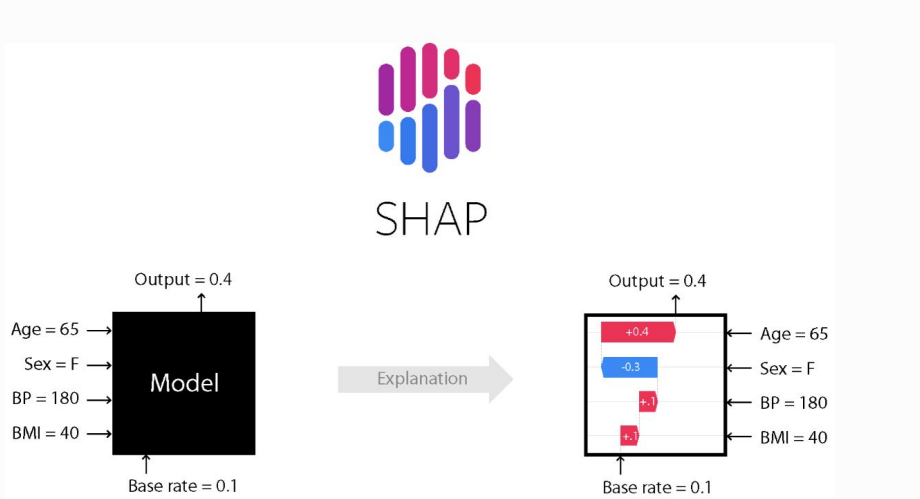
- вычислительно дорог — сложность $O(p*n)$, где p — число признаков, n — число перестановок
- чувствителен к случайности — важно использовать несколько повторов и доверительные интервалы
- использовать на валидационной/тестовой выборке, а не на train



SHAP

Семейство методов, основанных на Shapley Values (значения Шепли) — концепции из теории игр, предложенная Ллойдом Шепли в 1953г.. Изначально были предложены в задаче распределения вознаграждений среди участников игры.

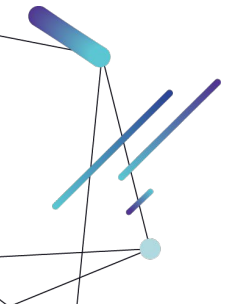
Как метод оценки важности признаков предложены в 2017 году в статье “A Unified Approach to Interpreting Model Predictions”. В настоящее время — один из наиболее частоиспользуемых методов.



SHAP: построение

Пусть определены:

- N — множество игроков, играющих в игру, |N| = n
- S — произвольное подмножество N (коалиция)
- v(S) — функция выигрыша, характеризующая успешность игры с участием игроков из S

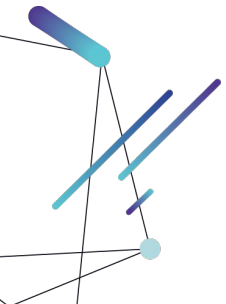


SHAP: построение

Алгоритм:

Чтобы учесть вклад конкретного игрока $i \in N$ в коалицию S нужно:

1. рассмотреть все возможные подмножества без игрока i и посчитать прогноз без них
2. рассмотреть все возможные подмножества с игроком и посчитать прогноз после добавления игрока i
3. получить разность вычисленных значений



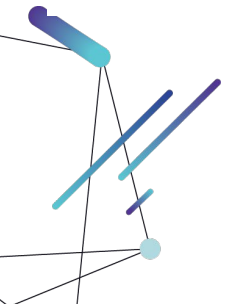
SHAP: построение

Согласно требуемым действиям, значение Шепли для игрока i можно определить как:

$$Sh(v)_i = \sum_{S \subseteq N \setminus i} \frac{|S|!(n - |S| - 1)!}{n!} \Delta(i, S)$$

где:

- $\Delta(i, S) = v(S \cup i) - v(S)$ прирост от добавления игрока i в коалицию S;
- S произвольная коалиция из множества $N \setminus i$
- $v(S)$ – характеристическая функция игры, с участием коалиции S
- $\frac{|S|!(n-|S|-1)!}{n!}$ нормирующий множитель для каждого слагаемого



SHAP: аксиоматика

1. **Локальная точность** — прогноз модели может быть разложен на комбинацию значений Шепли и признаков

2. **Аксиома болвана** — если признак отсутствует, то его вклад всегда нулевой

3. **Согласованность** — если при при дообучении вклад признака в модель увеличился, а вклад остальных признаков не уменьшился, то значение Шепли тоже не уменьшается

$$f(x) = \phi_0 + \sum_{i=1}^{M} \phi_i x_i$$

$$\text{If } x_i \in \{0, \emptyset\} \text{ then } \phi_i = 0$$

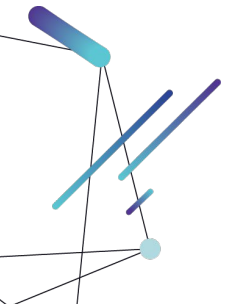
$$f'(S \cup \{i\}) - f'(S) \geq f(S \cup \{i\}) - f(S)$$
$$\forall S \subseteq F \setminus \{i\} \quad \Rightarrow \quad \phi_i(f') \geq \phi_i(f)$$

SHAP: практические особенности

- **Дорого вычислять:** 2^p число подмножеств — нужны аппроксимации
  — KernelSHAP (модель-агностичный)
  — TreeSHAP (для древесных моделей)

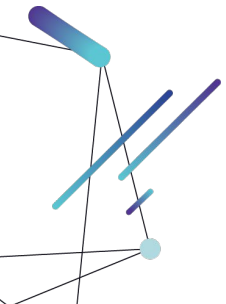
- **Устойчивее к коррелированным признакам,** чем Permutation Importance

- **При сильной мультиколлинеарности интерпретация может быть сложной** (SHAP делит вклад между признаками)



SHAP: практические особенности

- Тонкости с категориальными признаками
- **Можно получать как локальные (для одного предсказания), так и глобальные объяснения** (усреднённые SHAP)
- Для моделей с высокой размерностью признакового пространства вычислительно тяжёлый

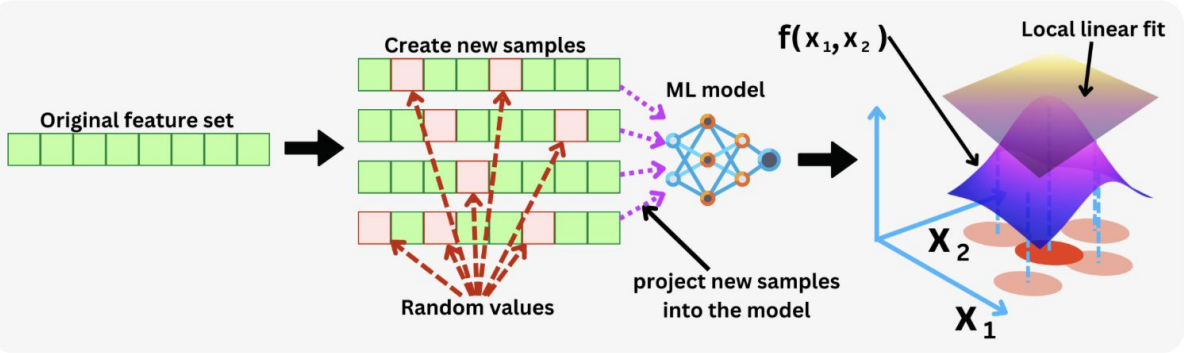


Explainable AI: SHAP

TheAiEdge.io

DLS

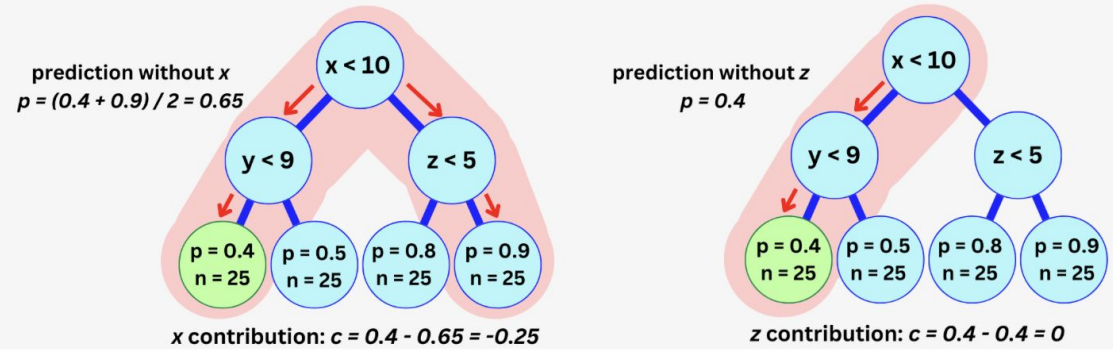
Kernel SHAP: LIME with Shapley Smoothing Kernel



Tree SHAP: Shapley estimates for Trees

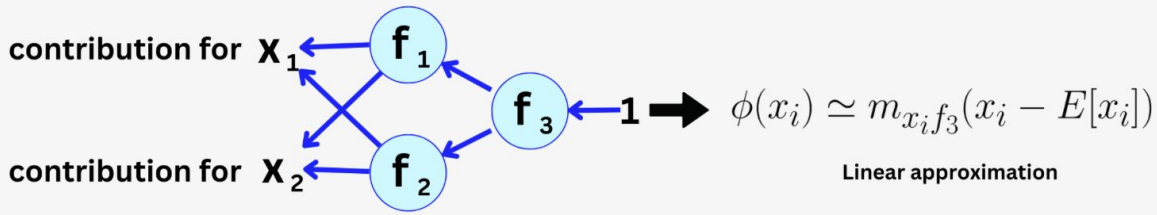
instance: *[x = 8, y = 5, z = 12]*

prediction with all the features: *p = 0.4*



Deep SHAP: Shapley estimates for Neural Networks

Backpropagate Shapley contributions



$$\phi(x_i) \simeq m_{x_i f_3}(x_i - E[x_i])$$

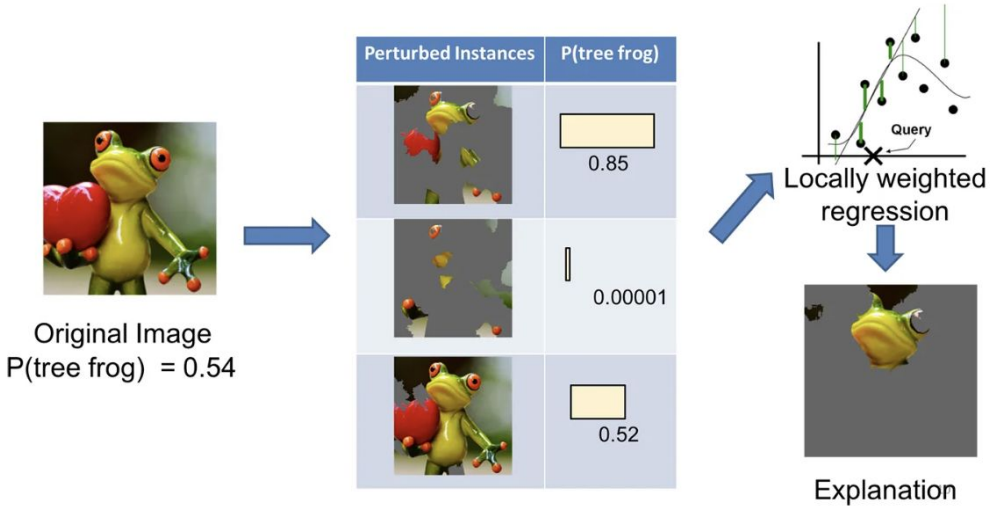
Linear approximation


Deep
Learning
School
Анализ
важности
признаков при
помощи
суррогатных
моделей

LIME

LIME — Local interpretable model-agnostic explanation — модель-независимый алгоритм объяснения модели, основанный на построен локальной интерпретируемой модели, котора аппроксимирует сложную модель вблизи конкретного прогноза.

Был предложен в 2016 году в статье ""Why Should I Trust You?"". Также один из наиболее используемых в настоящее время.



LIME: постановка

$f(x)$ — модель-черный ящик

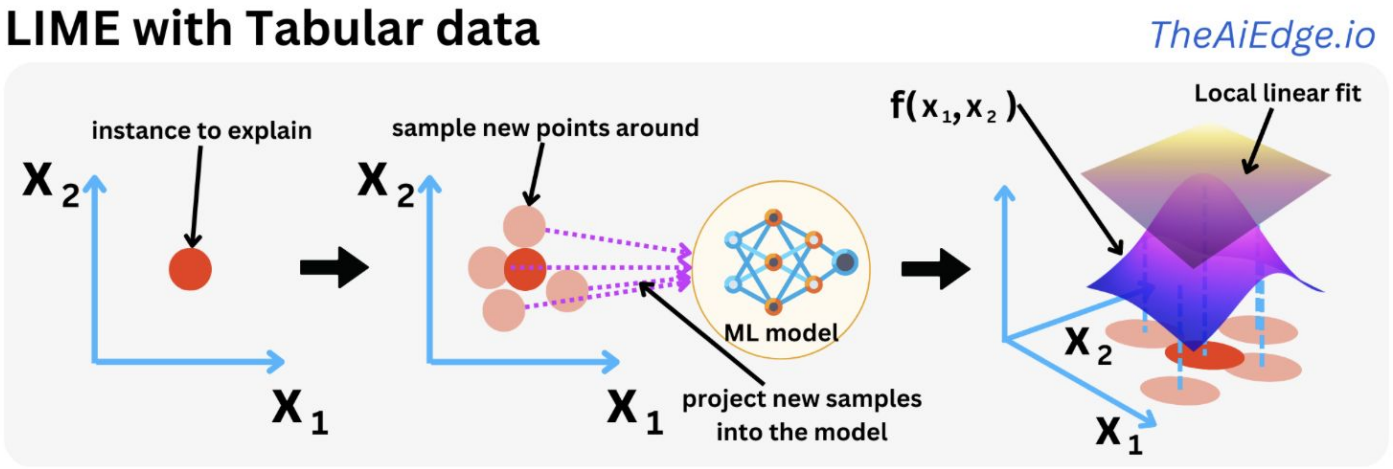
$g(z)$ — интерпретируемая модель

$$\xi(x) = argmin_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

L — расстояние между прогнозами моделей (MSE)

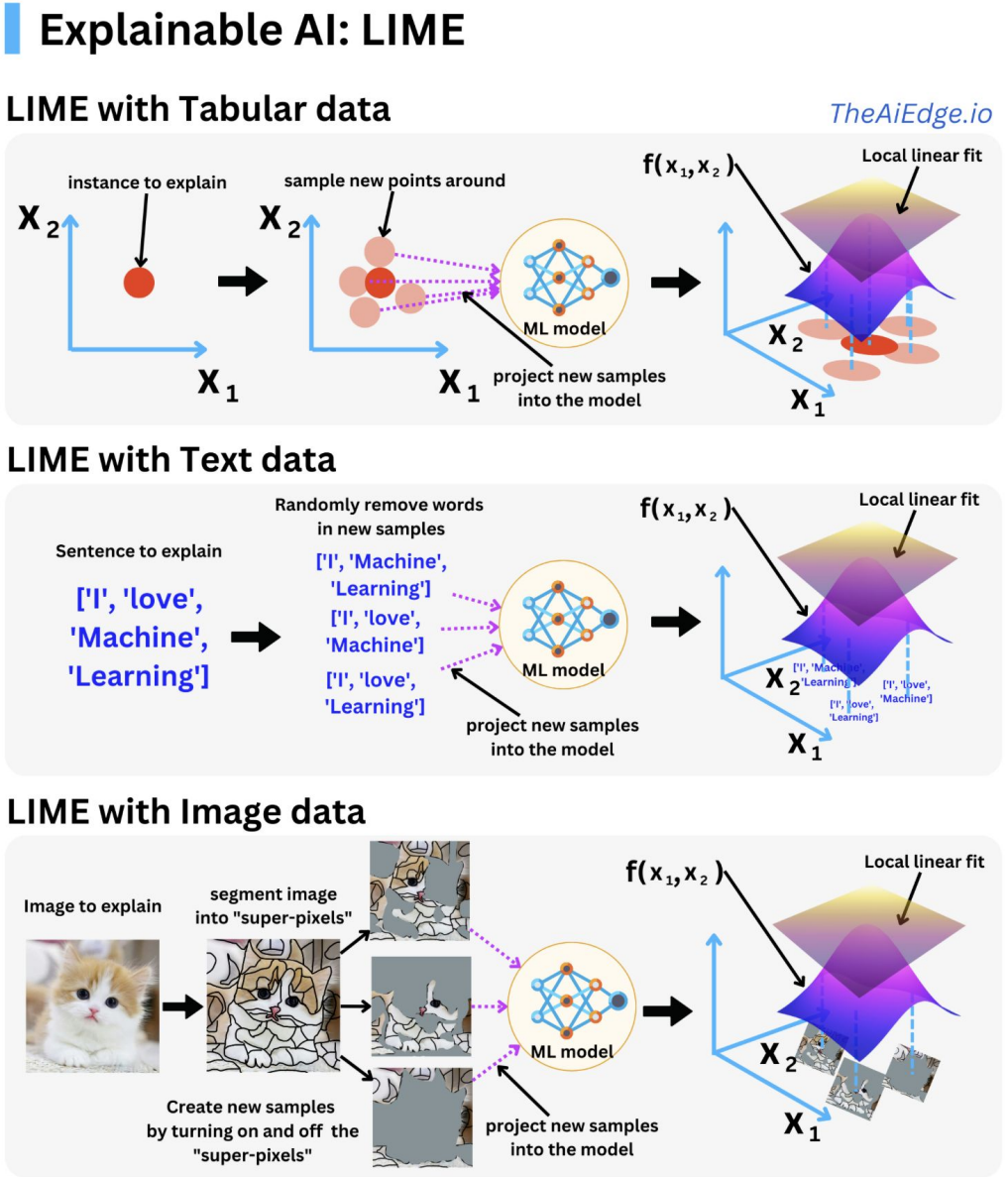
π_x — окресность наблюдения x

Ω — регуляризация



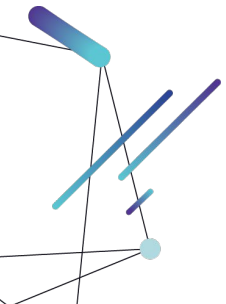
LIME: построение

1. Генерируем синтетические точки рядом с **x** (perturbations)
2. Считаем предсказания **f** для этих точек
3. Взвешиваем их по близости к **x**
4. Обучаем простую модель **g** (обычно линейную)
5. Коэффициенты **g** принимаем за локальные важности признаков



LIME: практические тонкости

- **Локальная** интерпретируемость (фокус на одном прогнозе)
- **Модель-независимый** метод
- **Важно следить за стабильностью** — выбор ширины окрестности влияет на результат
- **Не учитывает нелокальные взаимодействия** признаков
- Для дискретных и текстовых данных требуется аккуратная генерация perturbations



DLS

Спасибо за внимание!

tg: @sabrina_sadiekh,
jdata_blog

Садиех Сабрина

HSE, XAI researcher