Машинное обучение, ФКН ВШЭ
Семинар №11

1 Когда нужна интерпретация?

При решении любой ML-задачи есть выбор — использовать интерпретируемую по определению модель или нет. В первом случае методы интерпретации могут оказаться избыточны, во втором случае методы интерпретации необходимы. С их помощью мы анализируем, как признаки влияют на прогноз в целом и точечно.

Существует множество способов разделить методы интерпретации между собой. Основные способы разделения, это:

Ante-hoc и Post-hoc

- **Ante-hoc** (или **интерпретируемые по определению**) — это модели, в которых интерпретация заложена изначально в структуру. Их преимущество в том, что объяснение получается в комплекте с моделью.
- **Post-hoc** — это методы, которые применяются уже после обучения модели, когда сама структура модели сложна (такие модели называют моделями "черного ящика"(например, нейросети или иногда бустинги)).

Локальная и глобальная интерпретация

- **Локальная интерпретация** — объяснение конкретного предсказания. Отвечает на вопрос: «Почему модель приняла такое решение для данного объекта?»
- **Глобальная интерпретация** — анализ модели в целом: какие признаки наиболее важны, как они взаимодействуют между собой, как их изменение влияет на результат в среднем.

Model specific и Model agnostic

- **Model specific** — методы, предназначенные для конкретных классов моделей.
- **Model agnostic** — универсальные методы, которые можно применять к любой модели («чёрному ящику»).

Приложение полученных выводов может быть разным, в зависимости от предметной области.

Например, мы знаем:

- какие показатели здоровья для конкретного пациента привели к тому, что он отнесён в группу риска. Эта информация может быть использована для верификации и интерпретации врачом;
- какие показатели привели к отказу от кредита клиенту. Тогда, поскольку по закону каждый человек имеет *право на объяснение*, если решения по отношению к нему принимаются ИИ, интерпретация позволяет обеспечить это право;
- какие показатели привели к повышению вероятности найма для сотрудника, автоматически отобранного по резюме. Тогда возможно *верифицировать показатели на предвзятость*;
- какие показатели привели к тому, что оборудование было помечено как ломающееся (и впоследствии вышло из строя). Зная, что повлияло на поломку, сотрудник может быстрее локализовать и исправить проблему.

В общем же, в литературе выделяют четыре основных направления приложений XAI:

1. повышение доверия к ИИ;
2. обеспечение согласованности алгоритмов с человеческими ожиданиями и законодательством;
3. извлечение новых знаний из предметной области или генерация новых гипотез для бизнеса;
4. дебаггинг модели.

2 Интерпретация по определению

К интерпретируемым по определению (моделям, для которых интерпретация ante-hoc) относят модели, чья структура и параметры имеют математический смысл. К таким моделям относятся:

Линейная регрессия и обобщённые линейные модели

Классическая модель имеет вид:

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

где каждый коэффициент β_j показывает изменение отклика y при увеличении признака x_j на единицу при прочих равных. Благодаря этому коэффициенты имеют ясную количественную интерпретацию.

Логистическая регрессия

Модель описывает логарифм отношения шансов:

$$\log \frac{P(y=1)}{P(y=0)} = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p,$$

и коэффициенты β_j интерпретируются как вклад признака x_j в изменение шансов. Это делает модель прозрачной даже для задач бинарной классификации.

Наивный Байесовский классификатор

Модель основана на формуле Байеса:

$$P(y \mid x_1, \ldots, x_p) = P(y) \prod_{j=1}^{p} P(x_j \mid y).$$

Интерпретация строится на том, что каждое предсказание определяется априорной вероятностью класса $P(y)$ и вкладом каждого признака x_j через условные вероятности $P(x_j \mid y)$. «Наивное» допущение независимости признаков позволяет прозрачно показать, какой именно фактор усиливает или ослабляет вероятность класса.

Решающие деревья

С одной стороны, дерево решений представляет собой иерархию правил вида «если–то», где на каждом шаге осуществляется пороговое разбиение признака. Интерпретация здесь заключается в том, что каждое предсказание можно объяснить конкретным маршрутом по дереву и набором условий.

С другой стороны, каждый узел в дереве строится как решение задачи уменьшения меры загрязнённости. Важность признака в узле N_k вычисляется как:

$$I_k = \frac{w_k \cdot impurity_k \; - \; w_{left(k)} \cdot impurity_{left} \; - \; w_{right(k)} \cdot impurity_{right}}{100},$$

где:

- w_k — доля наблюдений в узле от исходных данных,
- $impurity_k$ — мера загрязнённости (например, индекс Джини или энтропия),
- $w_{left(k)}, w_{right(k)}$ и $impurity_{left}, impurity_{right}$ — аналогичные величины для дочерних узлов.

Общая важность признака вычисляется как сумма по всем узлам, где он использовался, нормированная на общую важность всех признаков:

$$FeatureImportance(f_k) = \frac{\sum_{N_k : f = f_k} I_k}{\sum_j \sum_{N_k} I_k}.$$

Таким образом, интерпретация дерева по определению возможна как на:

- **локальном уровне** — через правило для конкретного объекта,
- **глобальном уровне** — через сравнение важностей признаков.

Существуют также сопособы дизайна интерпретируемых DNN. Но это отдельная область исследований.

3 Графические (plot-based) методы

Название *plot-based* для класса методов, с которыми мы ознакомимся ниже, не является строгим или формальным. Оно скорее отражает смысл: наиболее информативная часть этих методов сосредоточена в их графическом представлении. Эти методы были одними из первых в силу развития области — все графические методы удобны на табличных данных,

которые были наиболее распространены в началах дисциплины машинного обучения. Разберем те, которые применяются по сегодняшний день — ICE, PDP и ALE.

3.1 ICE: Individual Conditional Expectation

Individual Conditional Expectation (ICE) — графики индивидуального условного ожидания. Задача метода: оценить, как меняется прогноз модели с изменением конкретного признака для каждого объекта в некотором наборе X_{test}.

Построение:

1. Зафиксируем множество X_{test} и некоторый признак j. Пусть j имеет m уникальных значений $[j_1, j_2, \ldots, j_m]$.
2. Исходный датасет X_{test} дублируем m раз:

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

   так, что для датасета X'_i значение признака j фиксируется равным j_i.
3. На каждом X'_i рассчитываем прогноз модели $f(X'_i)$.
4. На графике строим линии для каждого объекта, показывающие, как прогноз (ось y) меняется при изменении признака (ось x).

ICE реализован в `scikit-learn`. Метод даёт **локальное объяснение** — для отдельных объектов.

3.2 PDP: Partial Dependence Plot

Следующий тип графической оценки важностей — график частичной зависимости (Partial Dependence Plot, PDP). Он является логическим продолжением ICE: фактически, это усреднённый ICE-график. Смысл усреднения показан на рисунке 1. Усредняя ICE, PDP представляет собой **глобальную интерпретацию**.

Формализация. Пусть:

- X — набор данных, состоящий из n наблюдений, описанных m признаками,
- x_s и x_c — два множества признаков, где x_s содержит признаки, важность которых мы исследуем, а x_c — остальные признаки. Тогда $x_s \cup x_c = x$.

Функция частичной зависимости имеет вид:

$$PD(x_s) = \int \hat{f}(x_s, x_c)\, dP(x_c),$$

где $\hat{f}()$ — обученная модель, а $P(x_c)$ — распределение остальных признаков.

При $|x_s| = 1$ эта формула сводится к усреднению ICE по всем объектам. PDP также реализован в `scikit-learn`.

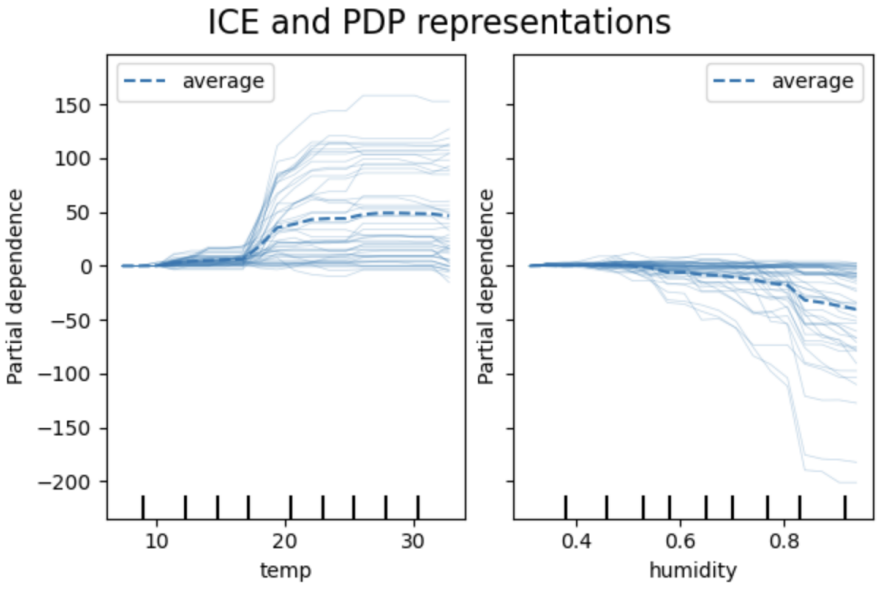

Рис. 1: ICE (голубые линии) и PDP (пунктир)

3.3 ALE: Accumulated Local Effects

Заключительный из основных графических методов — **Accumulated Local Effects (ALE)** (рис. 2. Метод был предложен в 2019 году как альтернатива PDP для случаев мультиколлинеарных признаков.

Формализация. Пусть:

- $f(x)$ — предсказания модели,
- F — интересующий нас признак,
- S_j — все остальные признаки, кроме F,
- $[x_F^{(k)}, x_F^{(k+1)}]$ — интервалы значений признака F, где $k = 1, 2, \ldots, K-1$,
- $N_j(k)$ — число наблюдений, попавших в интервал k.

Тогда эффект ALE вычисляется как:

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

Чтобы центрировать эффект относительно среднего по всем наблюдениям:

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

Алгоритм построения.

1. Выбрать интересующий признак F.
2. Разделить диапазон его значений на интервалы.
3. Для каждого интервала:
   а) подставить нижнюю границу $x_F^{(k)}$ и рассчитать прогноз $f(x_F^{(k)}, S_j)$;
   б) подставить верхнюю границу $x_F^{(k+1)}$ и рассчитать прогноз $f(x_F^{(k+1)}, S_j)$;
   в) усреднить разницу прогнозов по наблюдениям интервала.
4. Суммировать эффекты по интервалам и центрировать.

ALE также является **глобальным методом**, менее чувствительным к мультиколлинеарности, чем PDP.

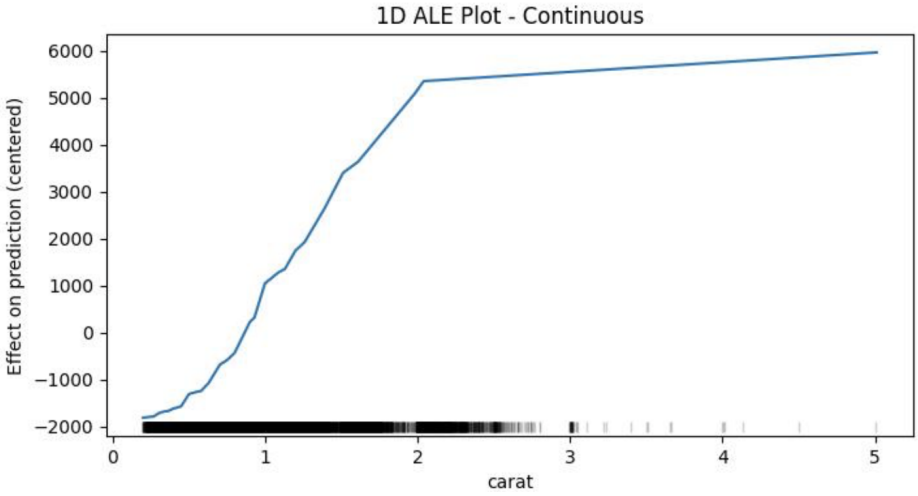

Рис. 2: AlE plot, pyALE (https://github.com/DanaJomar/PyALE)

4 Permutation Importances

Следующий класс методов дает для оценки важности коэффициенты. Это позволяет сравнивать важность признаков между собой в числовом формате, что удобнее и нагляднее, чем визуально. Разберем 3 метода — универсальных и применяемых в настощее время, в том числе для DNN.

Первым разберем Permutation Importances. Идея метода такова — если признак значим, то случайные перестановки признака сильно ухудшат производительсть модели.

Формально, пусть $a(x)$ — модель, обученная на множестве признаков $F = \{f_1, f_2, \ldots, f_n\}$, а e_{orig} — ошибка модели на тестовом наборе данных. Зафиксируем признак f_i. Для него выполняются следующие шаги:

1. Случайным образом переставим его значения в тестовом наборе данных.
2. Вычислим прогноз модели на тестовых данных с перестановкой.
3. Оценим ошибку модели на наборе данных с перетасовкой e_{perm}.
4. Оценим важность признака как разность $e_{orig} - e_{perm}$ или отношение $\frac{e_{perm}}{e_{orig}}$.

Алгоритм повторяется для всех признаков, после чего они сортируются по убыванию важности.

Общая важность признака f_i определяется как:

$$Importance(f_i) = e_{orig} - \frac{1}{K}\sum_{k=1}^{K} e_{perm,k},$$

где K — число перестановок.

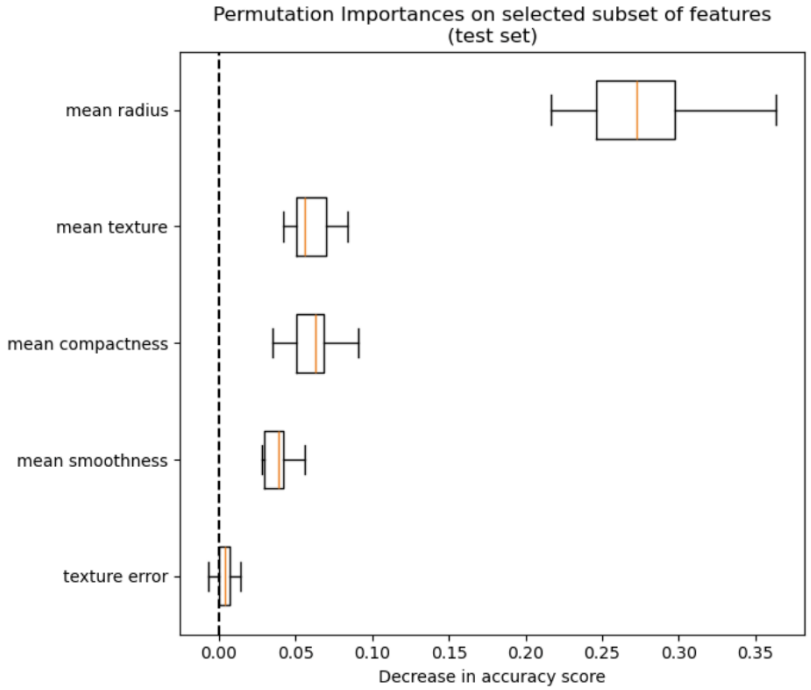

Рис. 3: Пример использования оценок, полученных из перестановочной важности — построение изменений метрики и анализ разброса, позволяющий оценить силу и устойчивость влияния перестановки на признак.

5 SHAP

Метод SHAP (SHapley Additive exPlanations) — один из самых популярных методов. Популярность обусловлена универсальностью — используя

его основу можно интерпретировать любые модели — от моделей машинного до моделей глубинного обучения, обученных при том на разных модальностях данных (текст, картинки, таблицы и даже геномные данные).

Сами значения Шепли были предложены ещё в 1951г. и изначально они не связаны с машинным обучением. Они пришли в область из задач теории кооперативных игр. Поэтому, чтобы ввести метод, нам понадобится ряд вспомгательных определений.

Пусть $N = \{1, \ldots, n\}$ — конечное множество игроков.
Любое подмножество $S \subset N$ называется **коалицией**, а само N — **грандкоалицией**.
Пару (S, v) будем называть **кооперативной игрой**, где $v : 2^N \to R$ — характеристическая функция игры.

В контексте интерпретации признаков:

- **Игроки** — это признаки.
- **Коалиции** — это подмножества признаков, используемых для прогноза.
- **Характеристическая функция** $v(S)$ — прогноз модели на подмножестве S признаков.

Вычисление значения Шепли.
Чтобы оценить вклад признака i:

1. Рассматриваем все подмножества $S \subseteq N \setminus \{i\}$ без игрока i и считаем прогноз $v(S)$.
2. Добавляем игрока i и считаем прогноз $v(S \cup \{i\})$.
3. Берём разность:

$$\Delta(i, S) = v(S \cup \{i\}) - v(S).$$

Значение Шепли для игрока i определяется как:

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

Здесь:

- $\Delta(i, S) = v(S \cup \{i\}) - v(S)$ — прирост от добавления игрока i в коалицию S,
- $\frac{|S|!(n-|S|-1)!}{n!}$ — нормирующий множитель (учитывает количество перестановок игроков).

Практические особенности Основная слабая черта метода — вычислительная сложность. Для n признаков требуется перебор всех 2^n подмножеств, что делает точный расчёт невозможным при большом n. На практике используются приближённые алгоритмы (например, KernelSHAP, TreeSHAP). Однако, вопреки недостаткам — метод используется чаще всего в силу утойчивости и теоретического фундмаента. Кроме того, он дает возможность строить как локальные так и глобальные объяснения.

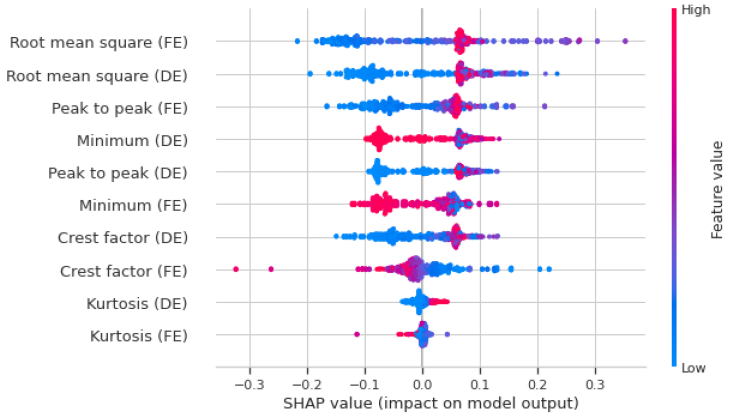

Рис. 4: Пример глобальной интерпретации с SHAP

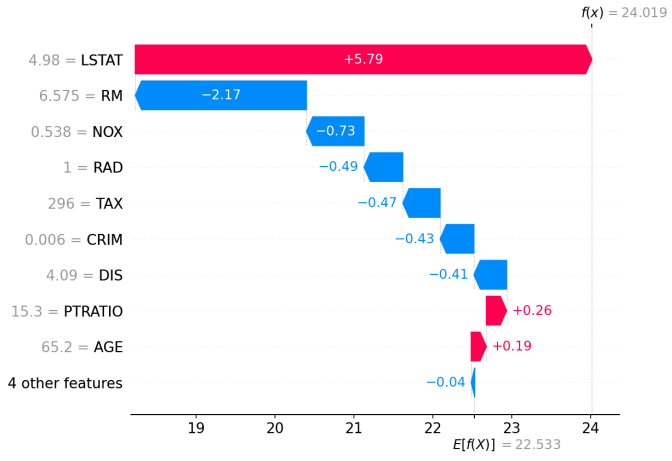

Рис. 5: Пример локальной интерпретации с SHAP

6 LIME

LIME (Local Interpretable Model-agnostic Explanations) — метод пост-хок интерпретации, основанный на аппроксимации сложной модели $f(x)$ простой и интерпретируемой моделью $g(z)$ в локальной окрестности точки x.

Основная идея Вместо того чтобы интерпретировать модель f глобально, мы обучаем *суррогатную модель* $g \in G$ (обычно линейную или деревообразную), которая аппроксимирует поведение f в малой окрестности точки x.

Формально решается задача оптимизации:

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

где:

- $L(f, g, \pi_x)$ — функция потерь, измеряющая, насколько хорошо g приближает f в окрестности x;
- π_x — весовая функция, задающая близость объектов к точке x;
- $\Omega(g)$ — штраф за сложность суррогатной модели, обеспечивающий её интерпретируемость.

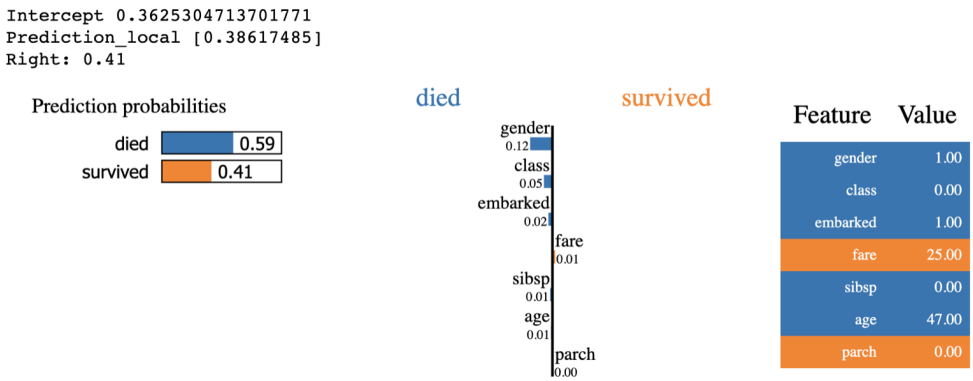

Рис. 6: LIME

Функция потерь Функция потерь имеет вид:

$$L(f, g, \pi_x) = \pi_x\big(f(z) - g(x')\big)^2,$$

где:

- $f(z)$ — предсказание исходной модели для объекта z,
- $g(x')$ — предсказание суррогатной модели в пространстве интерпретируемых признаков,
- π_x — коэффициент, зависящий от расстояния между z и точкой x.

Таким образом, объекты, близкие к x, получают больший вес при обучении g, а удалённые — меньший и в общем LIME позволяет понять, какие признаки повлияли на конкретное предсказание модели f и в какой степени. В отличие от глобальных методов, он даёт локальное объяснение, хотя существуют расширения для глобальной интерпретации.