ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

Национальный исследовательский университет «Высшая школа экономики»

Факультет экономических наук
Образовательная программа «Экономика»

КУРСОВАЯ РАБОТА

«Методы интерпретации моделей машинного обучения»

Студентка группы БЭК171
Махнева Елизавета Александровна

Научный руководитель:
Соколов Евгений Андреевич

Москва 2020

Содержание

1 Введение

2 Интерпретация - подумать над названием

2.1 Зачем она нужна

2.2 Что можно интерпретировать

2.3 Методы

3 Методы и их принципы

3.1 PDP

PDP (Partial Dependence Plot, график частичной зависимости) – график, который показывает зависимость прогноза модели от значения отдельного признака. С его помощью мы можем понять, как некоторый признак влияет на предсказание. Данный график можно изобразить для двух либо трех признаков из имеющихся.

Идея: визуализация – это отличный способ интерпретации. Если мы хотим понять, как признаки влияют на результат, можно посмотреть, как меняется прогноз от изменения одного признака при прочих равных. В идеальной ситуации мы бы построили график зависимости результата от всех признаков и меняли бы только один признак. Однако мы сталкиваемся с проблемой: если признаков больше двух, построить график не получится. Поэтому чтобы сохранить возможность визуализации, можно анализировать зависимость результата от одного признака без учета влияния остальных, построив график зависимости от одного признака. Аналогично можно изучать влияние одновременно двух признаков, построив трехмерный график.

<Пример 2мерного и 3мерного графиков>

3.1.1 Принцип работы

Обозначения:
$X = (x_1, \dots, x_d)$ – матрица признаков
x_1, x_2 – векторы исследуемых признаков
$X_b = (x_3, \dots, x_d)$ – векторы остальных признаков
$a(x_1, \dots, x_d)$ – предсказания модели как функция от признаков

Нам нужно получить функцию зависимости предсказания от одного-двух признаков при зафиксированных остальных: $g(x_1, x_2) = a(x_1, x_2 | \, x_3, \dots, x_d)$. Но если x_1 и/или x_2 зависимы с признаками из X_b, то возникает проблема. При изменении анализируемого признака меняется и зависимый с ним, который мы не рассматриваем – мы не сможем рассмотреть чистый предельный эффект одного признака, на него всегда будет наложен эффект другого предиктора. Поэтому одной из предпосылок метода является независимость исследуемых признаков от остальных.

Но даже с предпосылкой о независимости признаков функция $g(x_1, x_2)$ не будет показывать точный результат, так как предельные эффекты предикторов разные для разных объектов выборки. Поскольку нашей задачей является посмотреть влияние выбранных признаков в целом, мы рассмотрим, как влияют анализируемые признаки на среднее предсказание. То есть найдем матожидание предсказания модели при фиксированных исследуемых признаках (как констант с точки зрения матожидания):

$$\bar{g}(x_1, x_2) = \mathbb{E}(a(x_1, x_2, X_b) \, | X_b)$$

Таким образом, мы получим функцию, которая показывает предельные эффекты признаков для среднего предсказания. Но чтобы найти матожидание, мы должны знать истинные распределения признаков. Поскольку нам недоступна данная информация, можно воспользоваться методом Монте-Карло, чтобы примерно оценить искомую функцию:

$$\hat{g}(x_1, x_2) = \frac{1}{n} \sum_{i}^{n} a(x_1, x_2, X_b^{(i)}),$$

где $X_b^{(i)}$ – i строка матрицы X_b

Результат: функция показывает, как исследуемые признаки в среднем влияют на результат работы модели. Мы можем построить ее график, чтобы более наглядно посмотреть на влияние предикторов на предсказание.

3.1.2 Реализация

3.2 LIME

LIME (Local Interpretable Model-Agnostic Explanations, локально интерпретируемые не зависящие от модели объяснения) – метод, показывающий важность значений признаков для отдельного предсказания

3.2.1 Идея

Результаты некоторых моделей легко интерпретировать. Например, в линейной регрессии можно посмотреть на веса. Они показывают, насколько изменится предсказание при изменении признаков. Так для каждого конкретного предсказания можно понять, почему модель выдала именно такой результат – виден непосредственный вклад каждого признака.

Но не все модели легко интерпретировать. Например, некоторые архитектуры нейронных сетей. Они зачастую значительно превосходят линейные модели, но при этом сама структура модели представляет собой «черный ящик» – непонятно, как именно модель сформировала предсказание, какие признаки сильнее повлияли на решение нейронной сети.

Идея состоит в том, чтобы перенести свойство интерпретируемости простых моделей на более сложные. Мы можем обучить интерпретируемую модель по выборке, где ответами являются предсказания сложной модели. В процессе обучения модель анализирует зависимости непосредственно между признаками и предсказаниями сложной модели. Тогда мы сможем интерпретировать результаты простой модели, которые являются аппроксимацией предсказаний сложной модели.

Возникает проблема: сложная модель выявляет зависимости, которые, например, линейная модель может не уловить. Но мы можем воспользоваться свойством, что дифференцируемые функции можно линеаризовать в окрестности заданной точки. То есть, если мы будем рассматривать одно предсказание, то в его небольшой окрестности мы можем считать простую модель аппроксимацией более сложной.

<Свой пример для чиселок>

3.2.2 Принцип работы

У нас есть модель $f : \mathbb{R}^d \to \mathbb{R}$, предсказания которой мы хотим интерпретировать. Пусть $x \in \mathbb{R}^d$ – векторное представление предсказания, которое мы хотим интерпретировать, $x' \in \{0,1\}^{d'}$ – интерпретация предсказания в виде бинарного вектора.

Мы хотим найти модель g из класса интерпретируемых моделей G. Область определения g: $\{0,1\}^{d'}$. Стоит отметить, что сложность моделей обычно обратно зависит от ее интерпретируемости. Например, линейную модель с 2-3 признаками гораздо проще интерпретировать, чем модель с 10 и более признаками. Поэтому чтобы не терять интерпретируемость модели при ее приближении к более сложной, нужно ввести меру сложности $\Omega(g)$ как регуляризацию в нашей задаче.

Введем меру близости $\pi_x(z)$ между объектами x и z, чтобы определить окрестность рядом с x, внутри которой мы можем использовать простую модель. И наконец определим нашу функцию потерь, которую мы будем оптимизировать: $L(f, g, \pi_x)$ – разница между моделями f и g в окрестности, заданной π_x. Тогда в целом задача алгоритма выглядит следующим образом:

$$explanation(x) = \xi(x) = \underset{g \in G}{\operatorname{argmin}}(L(f, g, \pi_x) + \Omega(g))$$

Одной из особенностей алгоритма является его независимость от модели, которую необходимо интерпретировать. Поэтому мы не можем приписывать модели f никакие свойства. Вместо этого мы будем аппроксимировать ее, искусственно создавая объекты в окрестности x и получая для них предсказания из f.

3.2.3 Реализация

3.3 SHAP + реализация

4 Примеры

4.1 Первый прикольный пример

4.2 Второй прикольный пример

5 Данные и модели

5.1 Данные

5.2 Модели

5.3 Попытка интерпретации

6 Анализ результатов

7 Заключение

Список литературы

[1] Interpretable Machine Learning | Christoph Molnar | Christoph Molnar | 2020 | all pages

[2] “Why Should I Trust You?” Explaining the Predictions of Any Classifier | ...