Ход мамонтом: что мы узнали о Yandex Scale 2021

paveltro 30 октября 2018 в 14:02

Как интерпретировать предсказания моделей в SHAP

Big Data *, Машинное обучение *

Tutorial

Одной из важнейших задач в сфере data science является не только построение модели, способной делать качественные предсказания, но и умение интерпретировать такие предсказания.

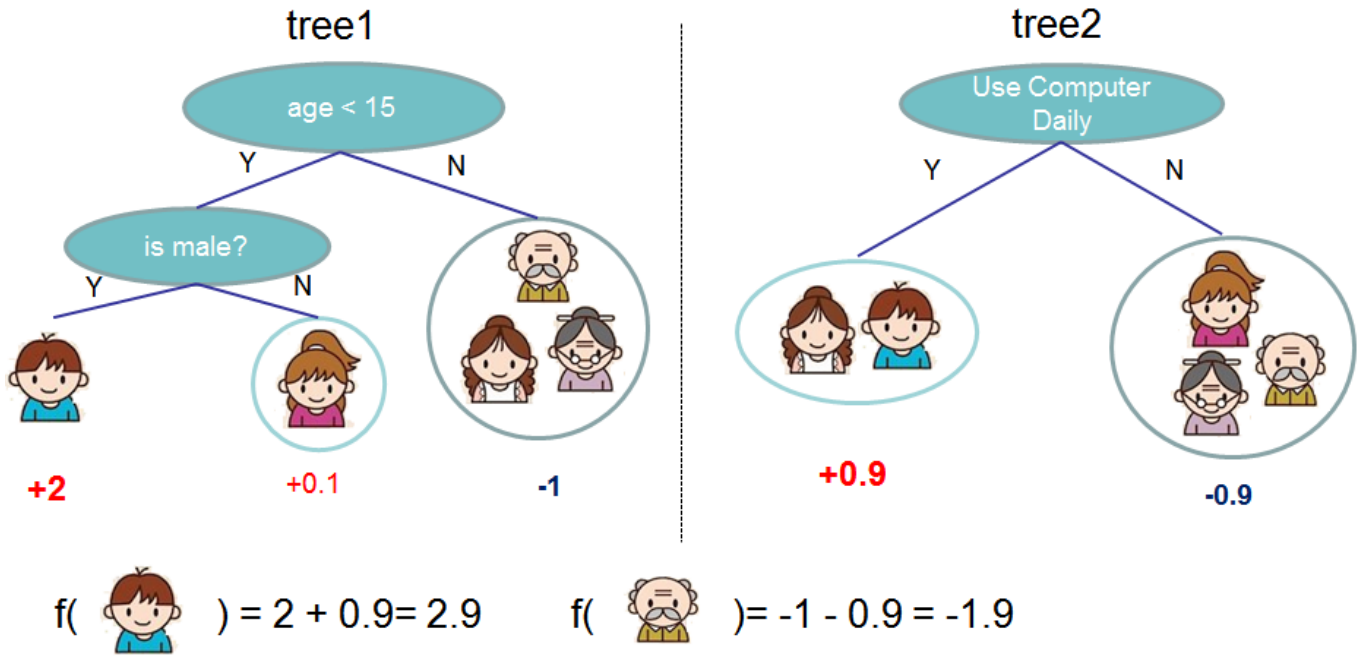
Если мы не просто знаем, что клиент склонен купить товар, но так же понимаем, что влияет на его покупку, мы сможем в будущем выстраивать стратегию компанию, направленную на повышение эффективности продаж.

Или модель предсказала, что пациент скоро заболеет. Точность таких предсказаний не бывает очень высокой, т.к. много скрытых от модели факторов, но объяснение причин, почему модель сделала такое предсказание, может помочь доктору обратить внимание на новые симптомы. Таким образом, можно расширить границы применения модели, если её точность сама по себе не слишком высока.

В этом посте я хочу рассказать о технике SHAP, которая позволяет заглянуть под капот самых разных моделей.

Если с линейными моделями всё более менее понятно, чем больше абсолютное значение коэффициента при предикторе, тем данный предиктор важнее, то объяснить важность фичей того же градиентного бустинга заметно сложнее.

Почему возникла необходимости в такой библиотеке



В стеке sklearn, в пакетах xgboost, lightGBM были встроенные методы оценки важности фичей (feature importance) для «деревянных моделей»:

1. **Gain**
   Эта мера показывает относительный вклад каждой фичи в модель. для расчета мы идем по каждому дереву, смотрим в каждом узле дерева какая фича приводит к разбиению узла и насколько снижается неопределенность модели согласно метрике (Gini impurity, information gain).
   Для каждой фичи суммируется её вклад по всем деревьям.
2. **Cover**
   Показывает количество наблюдений для каждой фичи. Например, у вас 4 фичи, 3 дерева. Предположим, фича 1 в узлах дерева содержит 10, 5 и 2 наблюдения в деревьях 1, 2 и 3 соответственно Тогда для данной фичи важность будет равна 17 (10 + 5 + 2).
3. **Frequency**
   Показывает, как часто данная фича встречается в узлах дерева, то есть считается суммарное количество разбиений дерева на узлы для каждой фичи в каждом дереве.

Основная проблема во всех этих подходах, что непонятно, как именно данная фича влияет на предсказание модели. Например, мы узнали, что уровень дохода важен для оценки платежеспособности клиента банка для выплаты кредита. Но как именно? Насколько сильно более высокий доход смещает предсказания модели?

Мы, конечно, можем сделать несколько предсказаний, меняя уровень дохода. Но что делать с другими фичами? Ведь мы попадаем в ситуацию, что надо получить понимание влияние дохода **независимо** от других фичей, при их некотором среднем значении.

Есть этакий среднестатистический клиент банка «в вакууме». Как будут меняться предсказания модели в зависимости от изменения дохода?

Тут-то на помощь и приходит библиотека **SHAP**.

Рассчитываем важность фичей с помощью SHAP

В библиотеке **SHAP** для оценки важности фичей рассчитываются значения Шэпли (по имени американского математика и названа библиотека).

Для оценки важности фичи происходит оценка предсказаний модели **с** и **без** данной фичи.

Немного предистории

Значения Шэпли идут из теории игр.

Рассмотрим сценарий: группа людей играет в карты. Как распределить призовой фонд между ними в соответствие с их вкладом?

- Сумма вознаграждения каждого игрока равна общей сумме призового фонда
- Если два игрока сделали равный вклад в игру, они получают равную награду
- Если игрок не внес никакого вклада, он не получает вознаграждения

- Если игрок не внес никакого вклада, он не получает вознаграждения

- Если игрок провел две игры, то его суммарное вознаграждение состоит из сумма вознаграждений за каждую из игр

Мы представляем фичи модели в качестве игроков, а призовой фонд — как итоговое предсказание модели.

Рассмотрим пример

Формула для расчета значения Шэпли для i-той фичи:

$$\phi_i(p) = \sum_{S \subseteq N/\{i\}} \frac{|S|!(n - |S| - 1)!}{n!}(p(S \cup \{i\}) - p(S))$$

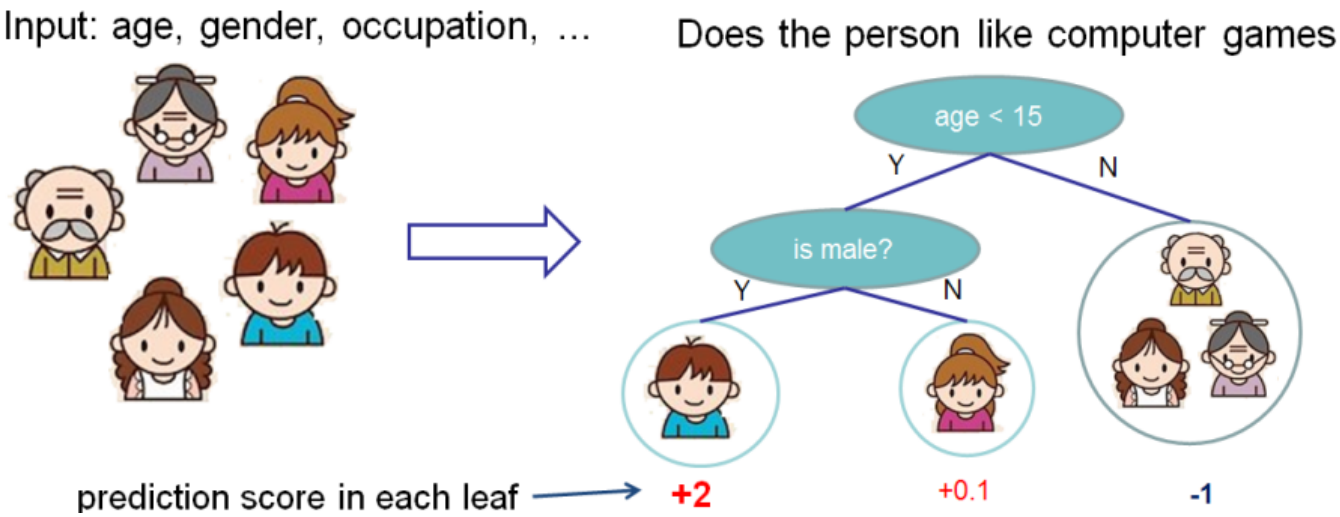
Здесь:

$p(S \cup \{i\})$ — это предсказание модели с i-той фичей,
$p(S)$ — это предсказание модели без i-той фичи,
n — количество фичей,
S — произвольный набор фичей без i-той фичи

Значение Шэпли для i-той фичи рассчитывается для каждого сэмпла данных (например, для каждого клиента в выборке) на всех возможных комбинациях фичей (включая отсутствие всех фичей), затем полученные значения суммируются по модулю и получается итоговая важность i-той фичи.

Данные вычисления чрезвычайно затратны, поэтому под капотом используются различные алгоритмы оптимизации вычислений, подробнее можно посмотреть по ссылке выше на гитхабе.

Возьмём ванильный пример из документации xgboost.



Мы хотим оценить важность фичей для предсказания, нравятся ли человеку компьютерные игры.

В этом примере для простоты у нас есть две фичи: age (возраст) и gender (пол). Gender (пол) принимает значения 0 и 1.

Возьмём Bobby (маленький мальчик в самом левом узле дерева) и посчитаем значение Шэпли для фичи age (возраст).

У нас есть два набора фичей S:

$\{\}$ — нет фичей,
$\{gender\}$ — есть только фича пол.

Ситуация, когда нет значений фичей

Разные модели по-разному работают с ситуациями, когда для сэмпла данных нет фичей, то есть для всех фичей значения равны NULL.

Будет считать в данному случае, что модель усредняет предсказания по веткам дерева, то есть предсказание без фичей будет $[(2+0.1)/2+(-1)]/2=0.025$.

Если же мы добавим знание возраста, то предсказание модели будет $(2+0.1)/2=1.05$.

В итоге значение Шэпли для случая отсутствия фичей:

$$\frac{|S|!(n-|S|-1)!}{n!}(p(S\cup\{i\})-p(S))=\frac{1(2-0-1)!}{2!}(1.025)=0.5125$$

Ситуация, когда знаем пол

Ситуация, когда знаем пол

Для Bobby для $gender$ предсказание без фичи возраст, только с фичей пол, равно $[(2+0.1)/2+(-1)]/2=0.025$. Если же мы знаем возраст, то предсказание — это самое левое дерево, то есть 2.

В итоге значение Шэпли для этого случая:

$$\frac{|S|!(n-|S|-1)!}{n!}(p(S\cup\{i\})-p(S))=\frac{1(2-1-1)!}{2!}(1.975)=0.9875$$

Суммируем

Итогое значение Шэпли для фичи age (возраст):

$$\phi_{AgeBobby}=0.9875+0.5125=1.5$$

Реальный пример из бизнеса

Библиотека SHAP обладает богатым функционалом визуализации, который помогает легко и просто объяснить модель как для бизнеса, так и для самого аналитика, чтобы оценить адекватность модели.

На одном из проектов я анализировал отток сотрудников из компании. В качестве модели использовался xgboost.

Код в python:

```
import shap

shap_test = shap.TreeExplainer(best_model).shap_values(df)
shap.summary_plot(shap_test, df,
                      max_display=25, auto_size_plot=True)
```

Получившийся график важности фичей:

Как его читать:

- значения слева от центральной вертикальной линии — это negative класс (0), справа — positive (1)
- чем толще линия на графике, тем больше таких точек наблюдения
- чем краснее точки на графике, тем выше значения фичи в ней

Из графика можно сделать интересные выводы и проверить их адекватность:

- чем меньше сотруднику повышают зарплату, тем выше вероятность его ухода
- есть регионы офисов, где отток выше
- чем моложе сотрудник, тем выше вероятность его ухода
- ...

Можно сразу сформировать портрет уходящего сотрудника: ему не повышали зарплату, он достаточно молод, холост, долгое время на одной позиции, не было повышений грейда, не было высоких годовых оценок, он стал мало общаться с коллегами.

Просто и удобно!

Можно объяснить предсказание для конкретного сотрудника:

Или посмотреть зависимость предсказаний от конкретной фичи в виде 2D графика:

Можно визуализировать даже предсказания нейронных сетей на картинках:

Заключение

Я сам узнал о SHAP значениях около полугода назад и это полностью заменило другие методы оценки важности фичей.

Главные преимущества:

- удобные визуализация и интерпретация

- честный расчет важности фичей
- возможность оценить фичи для конкретной подвыборки данных (например, чем отличаются наши покупатели от других клиентов в выборке), делается простым фильтром датасета в pandas и его анализом в shap, буквально пара строчек кода

Теги: data science, machine learning, feature importance, shap

Хабы: Big Data, Машинное обучение

8 Карма 0 Рейтинг

Павел Трошенков @paveltro

Пользователь

Facebook

Комментировать

ПОХОЖИЕ ПУБЛИКАЦИИ

28 ноября 2019 в 14:00

Как я решал соревнование по машинному обучению data-like

+26 16K 108 5 +5

4 марта 2019 в 12:58

Интуитивный RL (Reinforcement Learning): введение в Advantage-Actor-Critic (A2C)

+11 8K 67 0

22 сентября 2015 в 13:17

Big Data и Machine Learning? Вам на HighLoad++

+13 15K 77 9 +9

ЛУЧШИЕ ПУБЛИКАЦИИ ЗА СУТКИ

вчера в 17:31

И продолжается «вечеринка со свинцом (Pb)»...

+68 13K 33 53 +53

вчера в 18:21

Введение в программирование: заготовка игры-платформера на SDL в 300 строк C++

+37 6.7K 82 39 +39

вчера в 17:46

Больше механических клавиатур хороших и разных: новые модели, на которые стоит обратить внимание

+31 14K 29 58 +58

вчера в 20:18

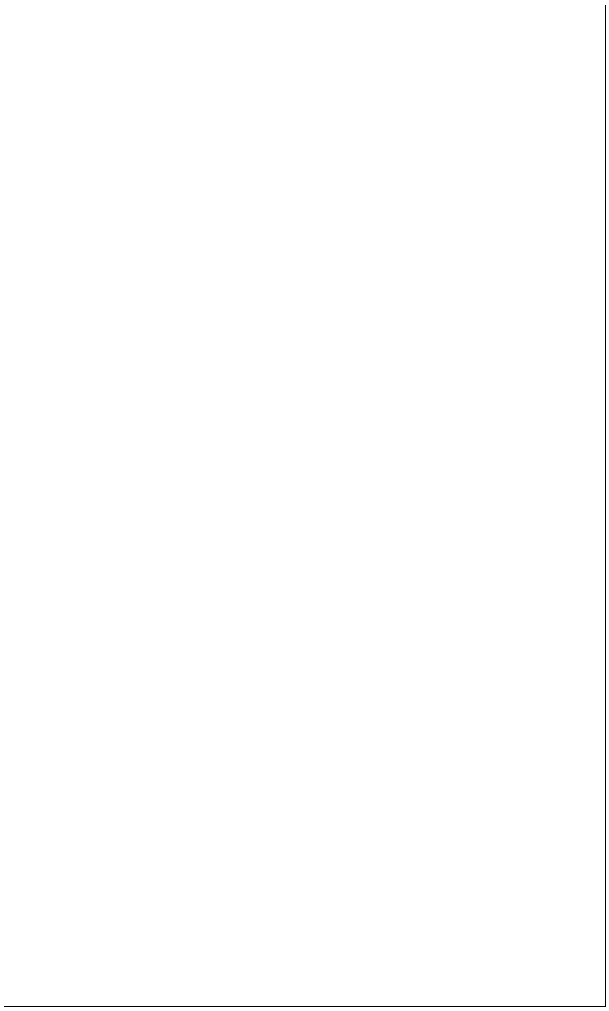
Закон о связи – он «все лучше и лучше»

+29 15K 17 24 +24

сегодня в 03:30

Барахолка в Испании: визит после двухмесячного перерыва

+27 4.4K 4 5 +5



О сайте

Техническая поддержка

Вернуться на старую версию