Кантонистова Елена Олеговна
Машинное обучение

Содержание

Часть I
Модуль 1

1. Лекция 1. Линейный методы регрессии

Базовые вещи знаем из курса математики - как обучать с помощью градиентного спуска и как решать аналитически.

1.1. Напоминание о том, что такое линейная регрессия

Предположим, что мы хотим предсказать стоимость одома y по его площади (x_1) и количеству комнат (x_2).

Линейная модель дял предсказания стоимости: $\alpha(x) = w_0 + w_1x_1 + w_2x_2$,

где w_0, w_1, w_2 - параметры модели (веса)

Линейная регрессия означает то, что все веса линейны от признаков (сами признаки могут быть любыми).

1.2. Общий вид линейной регрессии

$$\alpha(x) = w_0 + w_1x_1 + \cdots + w_nx_n$$

Сокращённая запись:

$$\alpha(x) = w_0 + \sum_{i=1}^{n} w_ix_i$$

Запись через скалярное произведение (с добавлением признака $x_0 = 1$):

$$\alpha(x) = (w, x)$$

1.3. Как мы обучаем модель

Мы просто пытаемся минимизировать ошибку предсказаний:

$$Q(\alpha, X) = \frac{1}{l}\sum_{i=1}^{l}(\alpha(x_i) - y_i)^2 = \frac{1}{l}\sum_{i=1}^{l}((w, x_i) - y_i)^2 \to \min_w$$

где l - кол-во объектов

1.4. Проблемы возникающие при обучении на данных

Допустим к предсказанию стоимость квартиры мы захотели добавить два признака: x_3 - район, в котором находится квартира и x_4 - удалённость от МКАД.

- Что такое район? Это категориальный признак и непонятно как на нём обучать.
- Также, когда мы добавляем признак, мы предполагаем, что он как-то монотонно влияет на ответ. А вот с удалённостью от МКАД так нельзя сказать. Есть районы, которые лежат вне МКАД, но при этом там квартиры дороже, чем внутри.

1.5. One-hot encoding - решение проблемы категориальных признаков

One-hot encoding (OHE) - мы создаём новые числовые столбцы, каждых из которых является индикатором одного из районов.

1.5.1. Проблемы OHE при обучении

Но при OHE мы можем столкнуться с проблемой, что мы один из столбцов OHE можем выразить через остальные, как 1 минус сумма остальных. Столбцы линейно зависимы. Это плохо для линейных моделей:

- При аналитическом решении у нас сломается формула

- При градиентном спуске, если $x_1, \ldots, x_l$ линейно зависимы, то $\exists v : (v, x) = 0$, а это означает, что мы можем добавить сколько угодно добавить $(w + \alpha v, x)$ и предсказание не поменяется, а вектор весов получается неоднозначный и получается много решений у задачи. При минимизации функции потерь могут получиться большие веса, а большие веса - переобучение.

В качестве решения мы можем выкинуть одну из колонок OHE (в sklearn за это отвечает параметр $\mathrm{drop}_f irst).,,-,.$

1.6. Решение проблемы немонотонных признаков

Мы можем разбить признак удалённости от МКАД на отрезки и сделать что-то типо OHE: сделать признаки, что квартира находится в $[0; 10)$ км от МКАД, в $[10; 30)$ км от МКАД. Получаются бинарные признаки

1.6.1. Как разбивать эти переменные?

Можно разбивать по квантилям - универсальное решение, чтобы не думать.

1.7. Метрики

Функцию потерь мы минимизируем. Метрика - другая функция, которую мы считаем, чтобы понять - насколько модель хорошая.

1.7.1. Среднеквадратичное отклонение (MSE)

Mean Squared Error - MSE

$$MSE(\alpha, X) = \frac{1}{l} \sum_{i=1}^{l} (\alpha(x_i) - y_i)^2$$

где l - количество объектов. Почему чаще всего используют MSE для обучения линейной регрессии? Минимизация этой ошибки - это максимизация правдоподобия, в случае если мы предполагаем, что наши данные имеют нормальное распределение, а это часто так.

Какие плюсы:

- Позволяет сравнивать модели между собой
- Подходит для контроля качества во время обучения

С какими проблемами мы столкнёмся?

- Выбросы
- Плохая интерпретируемость
- Неочевидно, хорошая ошибка или нет, нужно, например, сравнивать со средним значением целевой переменной, чтобы понять, хорошо мы предсказываем или нет. Неограничена сверху.

1.7.2. RMSE

Root Mean Squared Error - RMSE

$$RMSE(\alpha, X) = \sqrt{\frac{1}{l} \sum_{i=1}^{l} (\alpha(x_i) - y_i)^2}$$

Плюсы:

- Все плюсы MSE
- Сохраняет единицы измерения (в отличие от MSE)

Минусы:

- Тяжело понять, насколько хорошо данная модель решает задачу, так как тоже не ограничена сверху, как и MSE

1.7.3. $\mathbf{R}^2$

Коэффициент детерминации - R^2

$$R^2(\alpha, X) = 1 - \frac{\sum\limits_{i=1}^{l}(\alpha(x_i) - y_i)^2}{\sum\limits_{i=1}^{l}(y_i - \overline{y})^2}$$

где $\overline{y} = \frac{1}{l}\sum\limits_{i=1}^{l} y_i$

Коэффициент детерминации - это доля дисперсии целевой переменной, объясняемая моделью.

- Чем ближе R^2 к 1, тем лучше модель объясняет данные
- Чем ближе R^2 к 0, тем ближе модель к константному предсказанию
- Отрицательный R^2 говорит о том, что модель плохо решает задачу, и даже хуже, чем константное предсказание.

1.7.4. MAE

Mean Absolute Error - MAE

$$MAE(\alpha, X) = \frac{1}{l}\sum_{i=1}^{l}|\alpha(x_i) - y_i|$$

1.7.5. MAPE

Mean Absolute Percentage Error - MAPE

$$MAPE(\alpha, X) = \frac{1}{l}\sum_{i=1}^{l}\frac{|y_i - \alpha(x_i)|}{|y_i|}$$

MAPE измеряет относительную ошибку

Плюсы:

- Ограничена: $0 \leq MAPE \leq 1$
- ХОрошо интерпретируема: например, MAPE = 0.16 значает, что ошибка модели в среднем осставляет 16% от фактических значений

Минусы:

- По разному относится к недо- и перепрогнозу. Например, если правильный ответ $y = 10$, а прогноз $\alpha(x) = 20$, то ошибка $\frac{|10-20|}{10} = 1$, а если ответ $y = 30$, то ошибка $\frac{|30-20|}{30} = \frac{1}{3} = 0.33$

1.7.6. SMAPE

Symmetric Mean Absolute Percentage Error - SMAPE. Симметричный вариант MAPE:

$$SMAPE(\alpha, X) = \frac{1}{l}\sum_{i=1}^{l}\frac{|y_i - \alpha(x_i)|}{(|y_i| + |\alpha(x_i)|)/2}$$

SMAPE - попытка сделать симметричным прогноз - то есть дать одинаковую ошибку для недо- и перепрогноза

Проверим: пусть правильный ответ $y = 10$, а прогноз $\alpha(x) = 20$, то ошибка $= \frac{|10-20|}{|10+20|/2} = \frac{2}{3} = 0.67$, а если ответ $y = 30$, то ошибка $\frac{|30-20|}{|30+20|/2} = \frac{2}{5} = 0.4$

Сейчас уже в среде прогнозистов сложилось более-менее устойчивое понимание, что SMAPE не является хорошей ошибкой. Тут дело не только в завышении прогнозов, но ещё и в том, что наличие прогноза в знаменателе позволяет манипулировать результатами оценки.

1.7.7. MSLE

Mean Squared Logarithmic Error - MSLE. Среднеквадратическая логарфмическая ошибка

$$MSLE(\alpha, X) = \frac{1}{l}\sum_{i=1}^{l}(\log(\alpha(x_i)+1) - \log(y+1))^2$$

Особенности:

- Подходит для задач с неотрицательной целевой переменной ($y \geq 0$)
- Штрафует за отклонения в порядке величин
- Штрафует заниженные прогнозы сильнее, чем завышенные

1.7.8. Квантильная регрессия

Квантильная функция потерь:

$$Q(\alpha, X^l) = \sum_{i=1}^{l}\rho_\tau(y_i - \alpha(x_i))$$

где $\rho_\tau(z) = (\tau-1)[z<0]z + \tau[z\geq 0]z = (\tau - \frac{1}{2}) + \frac{1}{2}|z|$

Параметр $\tau \in [0;1]$. Чем больше τ, тем больше штрафуем за занижение прогноза.

Теорема: Пусть в каждой точке $x \in X$ (пространство объектов) задано распределение $p(y|x)$ на ответах для данного объекта. Тогда оптимизация функции потерь $\rho_\tau(z)$ даёт алгоритм $\alpha(x)$, приближающий τ-квантиль распределения ответов в каждой точке $x \in X$

Иными словами: допустим мы предсказываем стоимость квартиры. Если мы используем MSE, то мы получим $\overline{y}$, если по признакам все параметры совпадают у разных объектов, но целевая переменная одинаковая. К примеру $\{10, 20, 90\}$ выдаст $\{40\}$. Мы можем попросить завысить прогноз, поставить квантиль.

- Если мы хотим завысить прогноз, то берём τ ближе к единице.
- Если занизить - берём τ ближе к нулю.

1.8. Небольшое дополнение по линейной регрессии в skelarn

В sklearn класс LinearRegression всегда использует MSE. В то время, как в SGDRegressor, эта та же самая линейная регрессия, но мы можем подставить нужную функцию потерь.