

Введение в анализ данных

Вопросы к контрольной работе 21.04.2021

vk: vk.com/uselessofflane, tg: [@fmakhnach](https://t.me/fmakhnach)

1. Что такое объекты и признаки в машинном обучении? Для чего нужен функционал ошибки? Что такое алгоритм (модель)?

Объект (sample) – то, для чего делается прогноз, "входные данные".

Признак – [числовая] характеристика объекта, которая используется в моделях при обучении и предсказании.

Функционал ошибки – функция, отражающая меру ошибочности ответов алгоритма. Примеры: MSE (Mean Squared Error), MAE (Mean Absolute Error).

Алгоритм, модель – функция, которая на вход принимает объект (в виде множества его признаков), а на выходе возвращает предсказание значения целевой переменной для этого объекта.

2. Чем задача классификации отличается от задачи регрессии? Приведите примеры задач классификации и регрессии.

Ответы на задачу регрессии вещественные: $\mathbb{Y} \subseteq \mathbb{R}$.

Ответы на задачу классификации целочисленные, число различных возможных ответов конечное. Зачастую классификация решает задачу отнесения объекта к какой-либо группе.

Пример задачи классификации:

Определение котиков на изображении (бинарная классификация: «котик»-«не котик»).

Пример задачи регрессии:

Определение ожидаемого дохода предприятия в следующий месяц.

3. Что такое вещественные (числовые), бинарные, категориальные признаки? Приведите примеры.

Вещественные (числовые) признаки – признаки, представляющие из себя вещественные числа, над которыми можно выполнять математические операции. Примеры: средние расходы в месяц, число проданных товаров, средняя температура.

Бинарные признаки – признаки, принимающие всего два значения (зачастую – «истина» или «ложь»). Например: болен ли пациент раком, шел ли в этот день дождь, открыт ли магазин.

Категориальные признаки – признаки, значения которых ограничены конечным множеством. Например, город проживания, марка автомобиля, исполнитель песни.

4. В чём заключается обобщающая способность алгоритма машинного обучения? К чему приводит её отсутствие? Что такое переобучение?

Говорят, что алгоритм машинного обучения **обладает обобщающей способностью**, если ошибка этого алгоритма на тестовой выборке невелика и сопоставима с ошибкой на обучающей выборке.

Отсутствие обобщающей способности приводит к **переобучению** алгоритма, т.е. ошибка на обучающей выборке оказывается малой, однако ошибка алгоритма на новых данных становится существенно выше. Т.е. алгоритм «подстраивается» под обучающую выборку.

5. Что такое гиперпараметр? Чем гиперпараметры отличаются от обычных параметров алгоритмов? Приведите примеры **ред** параметров и гиперпараметров в линейных моделях.

Гиперпараметр – параметр модели, выбор которого лишь на основании обучающей выборке приведёт к потере качества. Для подбора таких параметров необходимо использовать дополнительные данные. В методе kNN количество соседей k является гиперпараметром.

Параметры нужны, чтобы подогнать модель под данные.

Гиперпараметры, чтобы контролировать сложность модели. Позволяют бороться с переобучением, подбираются кросс-валидацией.

6. Что такое отложенная выборка? Что такое кросс-валидация (скользящий контроль)? Как ими пользоваться для выбора гиперпараметров?

Отложенная выборка – метод проверки переобученности модели путём разбиения обучающей выборки на 2 части. Первая часть используется для обучения, вторая – для проверки качества обученной модели. Таким образом проверяется как ведёт себя модель на новых данных.

Кросс-валидация – метод проверки переобученности модели. Обучающая выборка разбивается на n частей, после чего модель обучают n раз, используя одну из n частей в качестве тестовой выборки, а остальные объекты – в качестве обучающей. Затем для оценки всей модели используют оценки качества модели в каждом эксперименте (можно взять среднее / наибольшее / наименьшее / что-то другое).

Эти методы можно использовать для подбора гиперпараметров: рассчитывать качество модели при различных значениях гиперпараметра и выбирать то значение, которое приводит к наилучшей метрике качества.

7. Как метод k ближайших соседей определяет класс для нового объекта?

Модель хранит все объекты, ответы на которые уже известны. При расчёте предсказания на новый объект рассчитывается «расстояние» (метрика) от этого объекта до каждого объекта из обучающей выборки, затем эти объекты упорядочиваются по возрастанию расстояния,

после чего выбирается k «ближайших» объектов. Наиболее часто встречающийся класс среди выбранных объектов и выбирается в качестве предсказания.

8. Опишите метод k ближайших соседей с парзеновским окном. Какие в нём есть параметры?

kNN с парзеновским окном – дополнение стандартного kNN , учитывающее расстояние до объектов (чем ближе объект, тем больший вклад он вносит). Предсказание вычисляется по следующей формуле:

$$a(x) = \arg \max_{y \in \mathbb{Y}} \sum_{i=1}^k w_i [y_{(i)} = y], \quad w_i = K \left(\frac{\rho(x, x_{(i)})}{h} \right)$$

где K – ядро (некоторая функция, напр $\frac{1}{x+1}$), h – «ширина окна», регулирует то, насколько нам важны дальние объекты.

9. Запишите формулу метода kNN для регрессии.

kNN для регрессии выдаёт среднее значение целевой переменной у k ближайших объектов:

$$a(x) = \frac{1}{k} \sum_{i=1}^k y_{(i)}$$

10. Что такое градиент? Какое его свойство используется в машинном обучении?

Градиент – функция, сопоставляющая каждой точке (x_1, \dots, x_n) функции $f(x_1, \dots, x_n)$ вектор, направление которого указывает в наибольшее **возрастание** функции в точке. Этот вектор, компоненты которого равны частным производным функции $f(x_1, \dots, x_n)$ по всем её аргументам.

Антиградиент – вектор-градиент, направленный противоположную сторону (градиент со знаком «минус»). Указывает в сторону наибольшего **убывания** функции в точке.

Градиентный спуск – метод нахождения минимума функции в случае, когда аналитическое решение невозможно / слишком затратное. Использует **антиградиент** для нахождения ближайшей точки *локального* минимума. Градиентный спуск используется для минимизации функции потерь при поиске коэффициентов в линейной регрессии.

11. Опишите алгоритм градиентного спуска.

Некоторым образом (например, случайно) выбирается начальный набор коэффициентов (начальное приближение) линейной регрессии. Затем выполняются некоторое количество итераций, на каждой из которых

1. Рассчитывается градиент функции потерь от текущих коэффициентов. Если он достаточно близок к нулю (достигли минимума), вычисление завершается. Альтернативно, можно остановить вычисление, когда разница между значениями, полученными на этом и предыдущем шагах мала;

2. К текущим коэффициентам прибавляется антиградиент, умноженный на заданный параметр λ (длина шага) – делаем «шаг вниз» по функции.

$$w^t = w^{t-1} - \eta \nabla Q(w^{t-1})$$

Полученные на последней итерации коэффициенты применяем для предсказаний модели.

12. Что такое стохастический градиентный спуск? В чём его отличия от обычного градиентного спуска? Какие у него плюсы и минусы?

Стохастический градиентный спуск – оптимизация градиентного спуска, заключающаяся в том, что вместо использования всех объектов обучающей выборки для вычисления градиента на каждом шаге используется лишь часть объектов.

Например, если имеется обучающая выборка на 100000 объектов, для вычисления точного значения градиента потребуется вычислить градиент функционала ошибки для каждого объекта. Чтобы этого избежать, на каждом шаге выбирается случайная подвыборка размера (например) 10, по которой и строится градиент. Это позволяет сократить затраты ресурсов для вычисления шага, однако шаг становится менее точным. Однако достаточное число таких не совсем точных шагов всё равно, как правило, приводят к достаточно точному результату.

Стохастический = случайный.

13. Как обучается линейная регрессия?

Линейная регрессия обучается путём минимизации функционала ошибки. То есть стараемся «подобрать» такие коэффициенты, чтобы функционал ошибки был минимальным. Этого можно достигнуть, например, с помощью аналитической формулы (сложно, не всегда применимо) или с помощью градиентного спуска (менее затратно, но результат может быть хуже).

P.S. Если между признаками существует линейная зависимость, то у задачи нахождения минимума будет несколько решений. В этом случае аналитическое решение не работает, а градиентный спуск даст какой-то результат (один из минимумов).

14. Что такое регуляризация? Как она помогает бороться с переобучением?

Существует эмпирическое наблюдение – большие коэффициенты линейной модели свидетельствуют о **переобучении**. Вывод: большие коэффициенты – плохо. Следующий вывод: можем препятствовать возникновению больших весов и таким образом улучшать качество модели.

Будем штрафовать за большие веса с помощью **регуляризатора** – функции, значение которой добавляется к значению функционала ошибки, например

$$\frac{1}{k} \sum_{i=1}^k (\langle w, x_i \rangle - y_i)^2 + \lambda \|w\|^2 \rightarrow \min$$

λ – коэффициент регуляризации (чем больше, тем важнее регуляризация).

P.S. w_0 не входит в регуляризацию.

15. Чем L1-регуляризация отличается от L2-регуляризации?

L₁-норма: $\|z\|_1 = \sum_{j=1}^d |z_j|$

L₂-норма: $\|z\|_2 = \sqrt{\sum_{j=1}^d z_j^2}$

L₁ регуляризация способствует появлению нулевых весов, L₂ регуляризация препятствует появлению больших по модулю весов. Помимо этого из-за модуля возникают проблемы с дифференцированием.

16. Что такое масштабирование признаков? Как его проводить? Зачем это нужно?

Масштабирование признаков – процедура приведения признаков к одному масштабу.

Пример: StandardScaler

$$x_i^j := \frac{x_i^j - \mu_j}{\sigma_j}$$
$$\mu_j = \frac{1}{k} \sum_{i=1}^k x_i^j, \quad \sigma_j = \sqrt{\frac{1}{k} \sum_{i=1}^k (x_i^j - \mu_j)^2}$$

После проведения масштабирования признаков можно делать выводы о важности признаков по весам.

P.S. Последнее возможно только в случае непереобученности модели, а значит следует также применить регуляризацию.

17. Чем функционал MSE отличается от MAE? Что такое функция потерь Хубера и для чего она нужна?

MSE – Mean Squared Error – среднеквадратичная ошибка, вычисляется по формуле

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

MAE – Mean Absolute Error – средняя абсолютная ошибка, вычисляется по формуле

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

MSE довольно сильно меняется из-за выбросов, в отличие от MAE. В свою очередь MAE не дифференцируема.

Функция потерь Хубера – функционал ошибки, являющийся комбинацией MSE и MAE. На достаточно близких к нулю значениях она ведёт себя как MSE, а на остальных – как MAE. Это позволяет избавиться проблемы с выбросами у MSE и проблемой с недифференцируемостью MAE. Формула:

$$L(y, a) = \begin{cases} \frac{1}{2}(y - a)^2, & |y - a| < \delta, \\ \delta (|y - a| - \frac{1}{2}\delta), & |y - a| \geq \delta \end{cases}$$

18. Как выглядит модель линейной классификации в случае двух классов?

Прогноз принадлежности объекта x некоторому классу строится следующим образом:

$$a(x) = \text{sign} \left(w_0 + \sum_{j=1}^d w_j x_j \right) = \text{sign} \langle w, x \rangle$$

Здесь значения 1 и -1 будут как-то соответствовать двум классам.

Вообще, $\langle w, x \rangle = 0$ задаёт гиперплоскость, а значит можно утверждать, что модель при обучении строит гиперплоскость, разделяющую пространство объектов на 2 части (по принадлежности классу).

19. Что такое отступ? Для чего он нужен?

Отступ (margin) – величина модели, характеризующая для каждого объекта

1. права ли модель в своём предсказании на данном объекте;
2. насколько она «уверена» в своём ответе.

Вычисляется по формуле

$$M_i = y_i \langle w, x_i \rangle$$

$M_i > 0 \Rightarrow$ классификатор даёт верный ответ,

$M_i < 0 \Rightarrow$ классификатор ошибается,

$|M_i|$ характеризует степень уверенности модели (чем больше величина, тем более уверена).

С помощью отступа можно определять выбросы – искать объекты, у которых отступ отрицательный и большой по модулю.