ТЕХНОЛОГИЧЕСКАЯ КАРТА ЗАНЯТИЯ

Тема занятия: Метрика машинного обучения

Аннотация к занятию: обучающиеся познакомятся с метриками машинного обучения. Рассмотрят метрики для задач регрессии (MSE и MAE) и классификации (Precision, Recall, F1-мера, ROC-AUC и LogLoss). Обсудят различия между метрикой и функцией потерь.

Цель занятия: сформировать у обучающихся представление о метриках машинного обучения. Рассмотреть метрики для задач регрессии и классификации. Показать различия между метрикой и функцией потерь.

Задачи занятия:

- познакомить обучающихся с метриками машинного обучения;
- рассмотреть основные метрики регрессии: MSE и MAE;
- ввести коэффициенты детерминации;
- рассмотреть метрики классификации: Precision, Recall, F1-мера, ROC-AUC и LogLoss;
- показать различия между метрикой и функцией потерь.

Ход занятия

Этап занятия	Время	Деятельность педагога	Комментарии, рекомендации для педагогов
Организационный этап	5 мин.	Друзья, здравствуйте! Сегодня мы поговорим о метриках машинного обучения.	Приветствие. Создание в классе атмосферы психологического комфорта.
Постановка цели и задач занятия. Мотивация учебной деятельности обучающихся	7 мин.	Тема урока — «Метрика машинного обучения» Прочитав тему урока, как вы думаете, что мы будем изучать сегодня? **Возможные ответы школьников:** ● узнаем, что такое метрика, виды метрик; ● рассмотрим применение метрики для конкретных задач. Мы: ● узнаем, что такое метрика; ● рассмотрим метрики для задач регрессии и классификации; ● обсудим отличие метрики от функции потерь.	Способствовать обсуждению мотивационных вопросов.

Изучение нового материала	50 мин.	Ранее мы встречали несколько метрик. Например, в задаче классификации датасета «Игрушка дьявола» мы измеряли долю правильно предсказанных ответов алгоритма к общему количеству объектов в выборке. Эту величину называют accuracy и считают одной из метрик в задаче классификации. Точного перевода на русский язык у термина нет. При оптимизации в линейной регрессии мы измеряли сумму квадратов отклонений ответа алгоритма от правильного ответа. Эту величину называют MSE — mean squared error или средняя квадратичная ошибка — и относят к метрикам регрессии. Давайте введём общее определение. Метрика — это численный показатель качества работы алгоритма для данной задачи. Метрика должна принимать на вход две вещи: ответы алгоритма и правильные ответы на объектах. Выдаёт метрика, как правило, одно число. Существует конвенция: чем больше значение метрики, тем более точно алгоритм предсказывает правильные ответы. Для примера посмотрим на метрику accuracy — долю правильных ответов. Она принимает значения от 0 до 1, где 1 соответствует верной классификации. А вот для MSE, среднеквадратичной ошибки, верно обратное: чем больше значение метрики, тем хуже. Поэтому иногда рассматривают метрику negative mean squared error, которая равна той же среднеквадратичной ошибке, только со знаком минус. Тогда чем больше эта величина, тем ближе она к нулю и тем лучше сработал алгоритм. Мы не будем придерживаться этой конвенции: у нас будут разные метрики.	Для справки: https://habr.com/ru/company/ods/blog/328372/ Перед уроком рекомендуется ознакомиться с материалами, представленными на сайте.

		Важно: то, какую метрику мы используем, зависит от конкретной задачи и формата данных. Помимо стандартных, вам ничего не мешает использовать метрики, созданные вручную. Такие случаи бывают в задачах компьютерного зрения или обработки естественного языка. Даже в привычных задачах — например, регрессии — часто возникает необходимость в более сложных метриках. Представьте, что вы создаёте модель предсказания спроса на определённый товар в магазине. Скажем, на ноутбуки. Вы построили модель, которая предсказала спрос на 20 ноутбуков, и в магазин привезли 20 штук. В реальности купили только 15. Что ж, придётся платить за хранение оставшейся техники: ничего хорошего, но не критично. А теперь представьте, что ваша модель предсказала покупку 10 ноутбуков — столько и привезли в магазин. А спрос показал 15. Тогда вы и денег не заработаете, и клиентов потеряете. Метрика качества алгоритма в данном случае может звучать так: альфа умножить на модуль разности между предсказанным значением и реальным значением, если разность больше нуля, и бета умножить на этот же модуль, если разность меньше нуля. Альфа и бета — это различные константы. На экране видно график этой метрики: она будет сильнее штрафовать алгоритм за непривезённые ноутбуки, чем за непроданные. Теперь давайте поговорим о метриках регрессии. Пусть у нас есть некоторая выборка из ell элементов.	

		Обозначим через y_true с верхним индексом i правильный ответ на i-ом объекте, а через y_pred с верхним индексом i — предсказанное значение целевой переменной. Вспомним две основные метрики, о которых вы уже знаете. Первая метрика — MSE, среднеквадратичная ошибка, то есть среднее значение квадрата отклонения от правильного ответа. Вторая метрика — MAE, средняя абсолютная ошибка, то есть среднее значение модуля отклонения от правильного ответа. Что можно сказать про эти метрики? Первая очень удобна для теоретических построений, например, для оптимизации в задаче линейной регрессии. Кроме того, MSE больше штрафует алгоритм за слишком большие отклонения, поэтому MSE в большей степени говорит о стабильности алгоритма. Главное преимущество MAE — в её понятной интерпретируемости: это просто среднее значение отклонения от правильного ответа. Есть две метрики, похожие на MSE и MAE. Первая — это RMSE. RMSE расшифровывается как root mean squared error и равняется корню из значения MSE. Её часто используют вместо MSE, потому что она находится в том же масштабе чисел, что и само значение целевой переменной. Например, если все отклонения алгоритма одинаковы и равны x, то значение MSE будет равно x^2 умножить на количество объектов и делить на количество объектов, то есть просто x^2. А вот значение RMSE будет равно просто x, что разумно.	

		Следующая метрика называется MAPE — mean absolute percentage error. Это среднее значение относительного отклонения ответа алгоритма от верного значения. Метрика хороша тем, что измеряет среднее значение абсолютной ошибки в процентах от реального значения целевой переменной. При этом, если вы внимательно посмотрите на неё, то заметите, что у некоторых моделей она может быть больше 1. Это значит, что алгоритм не очень хороший. Последняя метрика, о которой мы поговорим, называется R^2 или коэффициент детерминации. На экране вы видите формулу для её вычисления. Давайте вместе посмотрим на неё и поймём, что здесь написано. В числителе дроби написано среднее квадратичное отклонение, умноженное на l. А в знаменателе — сумма квадратов отклонения истинного значения целевой переменной Y от среднего значения целевой переменной. Да, здесь под y_true с чертой понимается именно среднее значение y_true по всем объектам выборки. То есть это выборочная дисперсия выборки игреков, умноженная на ell. Напомню: выборочная дисперсия характеризует разброс значения переменной вокруг своего среднего. Заметьте, что знаменатель вообще не зависит от работы алгоритма, то есть это просто постоянный нормировочный коэффициент. Таким образом, коэффициент детерминации это всего лишь преобразование, применённое к MSE. Почему же мы	

		применили именно такое преобразование: разделили на странную константу и вычли из единицы? Дело в интересных свойствах именно такой формулы. Первое свойство. Заметим, что если положить y_pred тождественно равным среднему значению целевой переменной y_true, то коэффициент детерминации R^2 становится равным 1−1 = 0. Из этого следует, что константная модель машинного обучения, которая предсказывает среднее значение целевой переменной независимо от объектов, имеет R^2 = 0. При этом R^2 может быть сколь угодно малым отрицательным числом. Такие модели неадекватны, поскольку результат их работы хуже первой попавшейся константной модели, которая всегда предсказывает среднее значение по выборке. Кроме того, очевидно, R^2 не превосходит 1 просто потому, что дробь не может быть отрицательной. Таким образом, значения метрики для «адекватных» моделей находятся в промежутке [0, 1]. Для приемлемых моделей предполагается, что коэффициент детерминации должен быть не менее 0,5. Модели с коэффициентом детерминации выше 0,8 считаются весьма хорошими. Значение коэффициента детерминации 1 может быть только в том случае, если мы в точности угадали все ответы. Наконец, поговорим об отличии понятий метрики и функции потерь. Многие величины являются одновременно и метриками, и функциями потерь. В чём же отличие одного от другого? Дело в том, что метрика	Для справки: Функция потерь — это мера количества ошибок, которые наша линейная регрессия делает на наборе данных.

		и функция потерь используются в различных целях. Метрика используется для измерения качества работы алгоритма. Поэтому для метрики важнее всего две вещи: адекватность для конкретной задачи машинного обучения, а также интерпретируемость, то есть возможность, посмотрев на значение метрики, сразу понять, хорошо решена задача или плохо. Интерпретируемость, кстати, особенно важна для заказчика. Функция потерь используется не как показатель качества работы алгоритма, а для организации процесса обучения модели. Например, при обучении линейной регрессии используется функция потерь MSE, потому что именно её мы минимизируем. А для регрессии с регуляризацией к MSE прибавляется ещё одно регуляризационное слагаемое — сумма квадратов весов с некоторым коэффициентом. Для функции потерь самое главное — то, насколько хорошим получится после обучения алгоритм. А то, насколько хорошим он получится, уже можно измерять с помощью метрики. Метрики в задачах классификации разнообразнее, чем в задачах регрессии. Объясню, почему. Рассмотрим задачу поиска мошеннических операций в банковской системе. Мы можем разделить операции на два класса: мошенническая или нет.	

		Изобразим все операции в строке, где 1 — это мошенническая операция, а 0 — нет. Предположим, что 10% всех операций в банке мошеннические. Тогда я могу предложить отличный алгоритм, который решает мою задачу с accuracy 90%. Как вы уже догадались, это «беспечный» алгоритм: любую операцию он классифицирует как честную и таким образом добивается accuracy в целых 90%. При этом он абсолютно бесполезен. Алгоритм, который отбраковывает 30% всех операций, в том числе и все мошеннические, естественно, больше отвечает нашим целям, но при этом его accuracy равна 80%. Хорошо, мы поняли, что одной accuracy не обойтись. В чём же проблема этой метрики? Она не чувствительна к разному балансу классов в задаче. Чтобы придумать что-то получше, посмотрим на классы по отдельности. Давайте нарисуем картинку. Вся совокупность объектов разделена на две части: ● слева объекты класса 1, мошеннические транзакции, ● справа — класса 0, прочие транзакции. Представим, что мы построили некоторый классификатор. Результат его работы визуализируем круговой областью: внутрь области попали объекты, которые наш классификатор отнёс к первому классу, вовне — объекты, предположительно, нулевого класса.	

		У идеального классификатора эта область должна совпадать с левой частью изначального разбиения. Мы видим, что наша плоскость разбилась двумя областями на четыре части. Левая половина круга — это объекты, которые мы будем называть True Positive. True — поскольку алгоритм на них не ошибается, и Positive, поскольку ответ алгоритма положительный. Правая половина круга — это False Positive. False — поскольку алгоритм ошибся. Тогда, по аналогии, левая половина вне круга — это False Negative, правая половина вне круга — True Negative. Первая важная метрика, которую мы изучим, называется precision — точность. Она равна проценту объектов первого класса среди тех, которые алгоритм отнёс к первому классу. Переводя на язык нашей задачи: сколько процентов из тех операций, которые мы нашли, действительно мошеннические. Конечно, чем больше эта метрика, тем меньше честных операций мы заблокируем, что для нас хорошо. В наших новых терминах precision вычисляется как отношение True Positive к True Positive + False Positive. Вторая важная метрика называется recall — полнота. Она равна проценту найденных объектов первого класса среди всех объектов первого класса. Иными словами, сколько процентов мошеннических операций мы нашли. Полнота — это эффективная метрика именно для этой задачи, ведь чем она больше, тем меньше мошеннических операций мы пропустим. В наших	

		новых терминах recall вычисляется как отношение True Positive к True Positive + False Negative. Итак, мы нашли две важнейшие метрики для бинарной классификации, особенно при несбалансированном распределении классов. Часто при поиске оптимального алгоритма делают так: фиксируют порог по одной из двух метрик, ниже которого мы не хотим опускаться, и оптимизируют вторую метрику. Иногда используют комбинацию этих двух метрик, которую называют F1-мерой. Она вычисляется по формуле 2 * precision * recall / (precision + recall) и возрастает с ростом каждой из метрик. Поговорим ещё об одной метрике в задаче классификации. В названии метрики аббревиатура ROC-AUC. Разберёмся, почему именно так. В отличие от предыдущих метрик, ROC-AUC работает не с предсказанными лейблами классов, а с предсказанными вероятностями классов. Вернёмся к задаче классификации банковских операций. Рассмотрим несколько объектов. Пусть наш алгоритм — например, логистическая регрессия — для каждого объекта предсказал вероятность того, что объект относится к классу 1. Давайте упорядочим объекты по убыванию этой вероятности. А теперь посмотрим, какие из объектов на самом деле относятся к первому классу. Видно, что алгоритм хорошо справился с задачей, но не идеально.	

		Построим на координатной плоскости некоторую кривую, которая будет описывать качество работы нашего алгоритма. Кривая строится следующим образом: мы стартуем из точки (0,0) и хотим прийти в точку (1,1). Всего у нас есть четыре объекта класса 1, поэтому поделим отрезок [0,1] на оси ординат на 4 части. У нас шесть объектов класса 0, поэтому делим отрезок на оси абсцисс на 6 частей. Теперь давайте идти по нашей выборке сверху вниз. Встречая объект класса 1, мы будем подниматься на одно деление вверх по кривой. Встречая объект класса 0, будем идти на одно деление вправо. У нас получилась кривая, проходящая из точки (0, 0) в (1, 1). Каждая точка на этой кривой соответствует некоторому порогу вероятности отсечения объектов. Например, выделенная точка соответствует вероятности 0,6 и имеет координаты (1/6, 3/4). Это значит, что при отнесении всех объектов с вероятностью выше этого порога к классу 1, а всех ниже этого порога — к классу 0, доля True Positive к общему количеству объектов первого класса будет равна ¾, а доля False Positive к общему количеству объектов нулевого класса будет равна ⅙. Эти величины принято называть соответственно True Positive Rate и False Positive Rate. Кривая, которую мы построили, называется ROC-кривой, от английского receiver operator characteristic. Теперь мы готовы определить метрику ROC-AUC. Она равна площади фигуры под ROC-кривой. Например, для	

		данной кривой она равна 0,875. Ясно, что в случае идеального упорядочивания объектов, когда все объекты первого класса имеют большую вероятность, а все объекты нулевого класса — маленькую, метрика будет равна 1. Перед вами три ROC-кривые, построенные для трёх разных моделей решения одной и той же задачи классификации. Первая кривая соответствует хорошему решению: ROC-AUC близка к единице, а сама кривая напоминает стороны прямоугольника. Вторая кривая соответствует решению похуже: ROC-AUC порядка 0,7, но кривая всё ещё выпукла вверх. Третья кривая идёт близко к диагонали квадрата. Это ROC-кривая случайного алгоритма, который вообще не смотрит на признаки и перемешивает объекты между собой. Такая модель не нашла в данных никаких закономерностей. В этом случае ROC-AUC равна 0,5. Соответственно, если ROC-AUC оказалась меньше одной второй, то алгоритм очень плохой: можно перевернуть его предсказания наоборот — будет лучше. Метрика ROC-AUC хорошая, поскольку на её значение не влияет дисбаланс классов, в отличие от той же accuracy. Это делает ROC-AUC одной из самых популярных метрик на практике.	

		Наконец, последняя метрика, о которой мы поговорим, стоит особняком от всех остальных. Мы упоминали её, когда изучали логистическую регрессию. Это метрика LogLoss, которая также известна как логистическая функция потерь или категориальная кросс-энтропия. Названий очень много. Мы опишем её ещё раз и обобщим несколько классов. Для примера решим задачу классификации рукописных цифр. У нас будет три вида цифр: 0, 1 и 2. Как и в прошлом примере, предположим, что наш алгоритм предсказывает вероятности принадлежности к классам — именно для такой постановки задачи применяется LogLoss. Перед нами таблица, которая фактически показывает результаты работы алгоритма. Всего у нас пять объектов, соответственно, пять строк. В первом столбце находится Y — верные ответы на объектах, принимающие значения 0, 1 или 2. Поскольку в задаче необходимо предсказать вероятности принадлежности к трём классам, у нас будет три столбца. В каждом из них — вероятность принадлежности объекта тому или иному классу. Напомню, в метрике LogLoss мы максимизируем вероятность правильной классификации. То есть для первого объекта, скажем, мы максимизируем вероятность p_1, потому что объект принадлежит классу 1. То же самое для всех i. Максимизировать вероятность — всё равно что максимизировать логарифм вероятности, потому что	

		логарифм — функция монотонная. Логарифм вероятности будет отрицательным, ведь вероятность меньше единицы. Поэтому естественно вместо этого минимизировать минус логарифм p_i. В итоге мы должны минимизировать минус логарифмы p_i, то есть логарифмы всех чисел, которые выделены на картинке. Теперь мы записываем среднее значение этих минус логарифмов правильных вероятностей. Эту величину мы и будем называть LogLoss. Пример вычисления LogLoss есть на слайде: здесь получился довольно высокий показатель, по крайней мере для трёх классов. LogLoss часто используют в качестве функции потерь. Например, для обучения нейронных сетей. Дело в том, что эта метрика вычисляется проще, чем, например, ROC-AUC или F1-мера, поэтому её гораздо легче оптимизировать. Можно использовать LogLoss и в качестве метрики: она подходит для сбалансированных классов. Впрочем, обучать на LogLoss можно и нейронную сеть.	
Закрепление изученного материала	15 мин.	**Вопросы для обсуждения** ● Что такое метрика? ● Какие бывают метрики для задач регрессии? ● Какие бывают метрики для задач классификации? ● Каковы области применения метрик?	Педагог организует беседу по вопросам.

Этап подведения итогов занятия (рефлексия)	8 мин.	**Вопросы для обсуждения** ● Чему я научился? ● С какими трудностями я столкнулся? ● Каких знаний мне не хватает для более глубокого понимания изученного материала? ● Достиг ли я поставленных целей и задач?	Педагог способствует размышлению обучающихся над вопросами.
Информация о домашнем задании, инструктаж по его применению	5 мин.	Дома повторите основные определения.	

Рекомендуемые ресурсы для дополнительного изучения:

1. Метрики в задачах машинного обучения. [Электронный ресурс] – Режим доступа: https://habr.com/ru/company/ods/blog/328372/.
2. Как считаются метрики регрессии. [Электронный ресурс] – Режим доступа https://nplus1.ru/material/2020/03/27/course-data-science-chapter-6.
3. Основные метрики задач классификации. [Электронный ресурс] – Режим доступа: https://webiomed.ru/blog/osnovnye-metriki-zadach-klassifikatsii-v-mashinnom-obuchenii/.
4. Оценка качества моделей. [Электронный ресурс] – Режим доступа: https://ml-handbook.ru/chapters/model_evaluation/intro.
5. Функции потерь регрессии. [Электронный ресурс] – Режим доступа: https://habr.com/ru/company/ods/blog/327250/.