KARPOV.COURSES >>>
КОНСПЕКТ

Конспект > 11 урок > Матрица ошибок и основные метрики классификации

>Оглавление

>Матрица ошибок

На прошлом урок мы научились строить модель логистической регрессии, поговорили про ряд геометрических интерпретаций, а также теоретических предпосылок. Качество алгоритма мы замеряли с помощью метрики Accuracy - доли правильно размеченных объектов, но у данной метрики есть свои минусы:

Чем плох Accuracy?

- Плохо работает при несбалансированных классах в выборке
- Например, есть 50 000 объектов: 47 000 - положительного и 3000 отрицательного классов. Пусть есть модель a(x) = +1. В таком случае мы получим Accuracy = 94%. Хотя в таком случае ни одного отрицательного класса не было угадано.

Давайте алгоритмы классификации научимся оценивать не только с помощью метрики Accuracy, а изобретем еще несколько новых методов.

Например, начнем с построения так называемой Матрицы Ошибок, в колонках которой будут расположены истинные классы наших объектов, а в строках - те классы, которые предсказывает модель.

	y = +1	y = -1
a(x) = +1	True Positive	False Positive
a(x) = -1	False Negative	True Negative

Если модель предсказала, что класс положительный и он в действительности оказался положительным: такую ситуацию называют True Positive.

Если модель предсказала класс как положительный, а на самом деле он отрицательный: такая ситуация - False Positive.

Алгоритм предсказывает отрицательный класс, а на самом деле он положительный: False Negative. И наоборот - True Negative.

Соответственно, для каждого объекта можно сказать, в какую ячейку из таблицы его можно отнести. Тогда каждую из этих ячеек можно заполнить количеством объектов, которые относятся к данным типам и посчитать, например, Accuracy.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

В терминах матрицы ошибок: Accuracy - доля верно размеченных объектов, т.е отношение суммы True Positive и True Negative к сумме всех типов.

Пример заполнения таблицы:

Пусть в выборке было 50 000 объектов: 47 000 положительного и 3000 отрицательного классов. Пусть модель не ошиблась на 40 000 объектах положительного и 1000 объектах отрицательного классов соответственно. В таком случае матрица ошибок будет выглядеть следующим образом:

	y = +1	y = -1
a(x) = +1	True Positive (40 000)	False Positive (2000)
a(x) = -1	False Negative (7000)	True Negative (1000)

>Precision и Recall

Давайте теперь перейдем непосредственно к метрикам, которые позволяют нам более справедливо оценивать качество работы наших моделей по сравнению с Accuracy.

Две самые популярные метрики - это так называемые Precision и Recall.

$$Precision = \frac{TP}{TP+FP}$$

Precision позволяет понять, какая доля объектов среди тех, которые мы назвали положительным классом, действительно к нему относится.

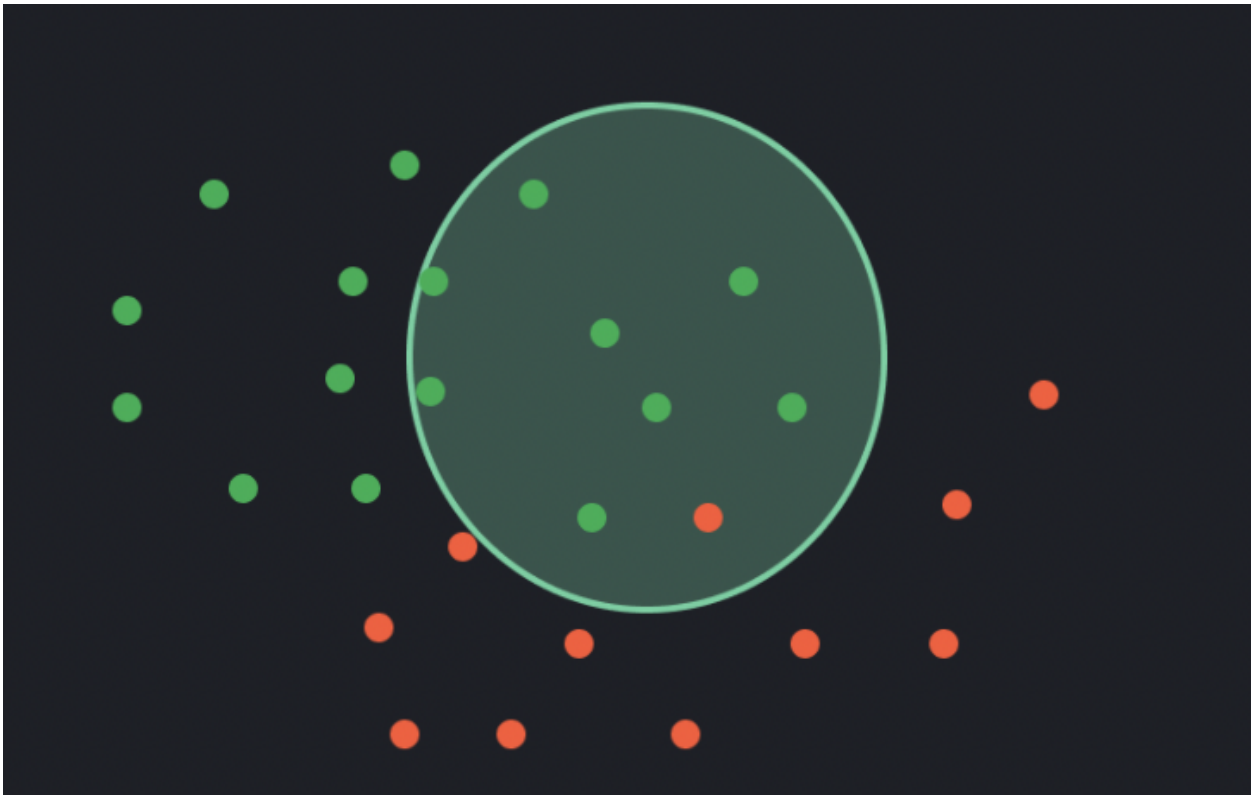
$$Recall = \frac{TP}{TP+FN}$$

Recall же показывает, сколько объектов среди тех, которые в общем были положительные, наша модель смогла выявить.

Но модель достаточно редко обладает как хорошим значением Precision, так и хорошим значением Recall одновременно. И, как правило, приходится балансировать между двумя этими метриками.

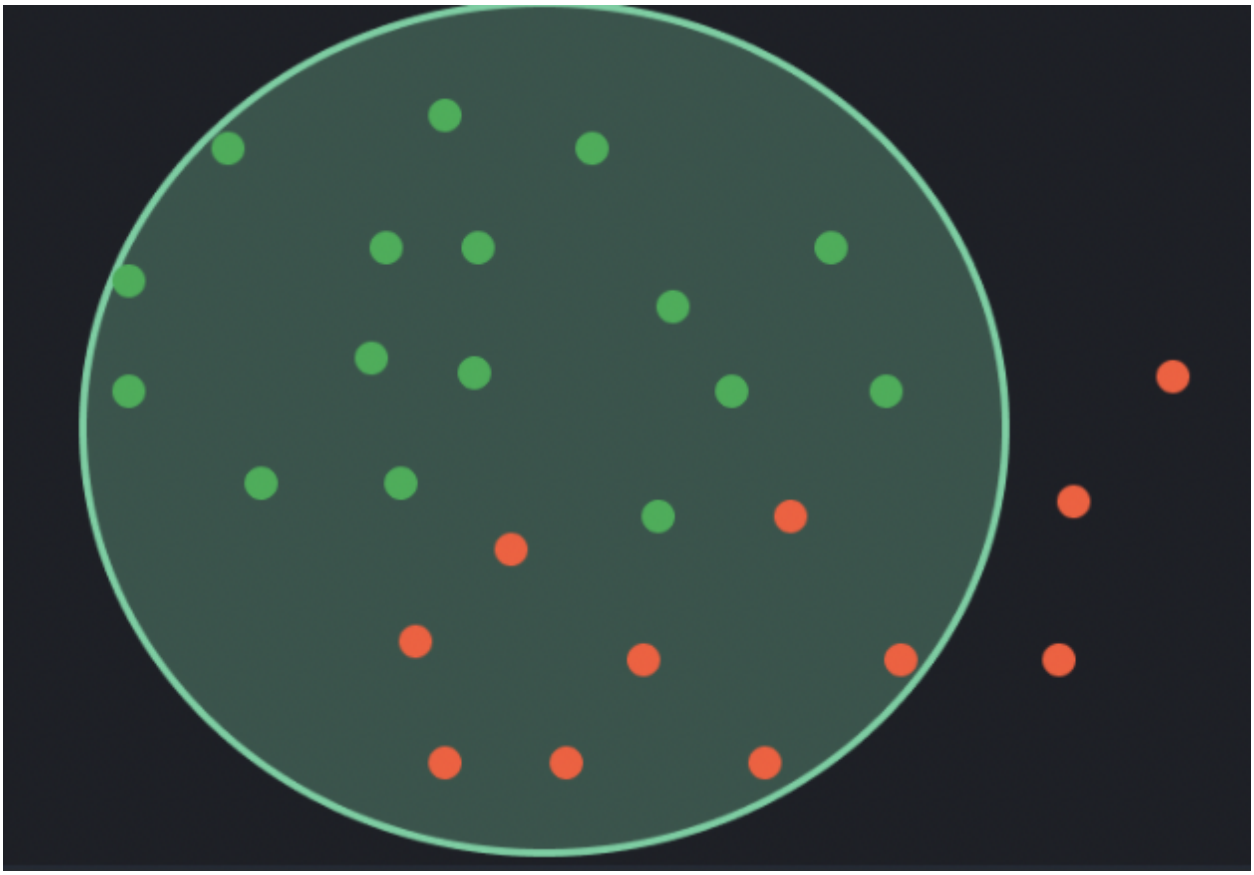
Давайте посмотрим, как будут выглядеть модели при сильном Precision и при сильном Recall.

Сильный Precision



Precision будет высоким, если количество неправильно предсказанных объектов будет минимальным, но при этом будет высокая доля верно размеченных

Сильный Recall



Recall будет высоким, в случае, когда модель покрывает как можно большее количество положительных классов, которые изначально были в выборке.

>Объединение Precision и Recall

На прошлом шаге мы обсудили, что, как правило, приходится балансировать между двумя этими метриками. Давайте рассмотрим варианты, как их можно было бы объединить.

Есть несколько вариантов:

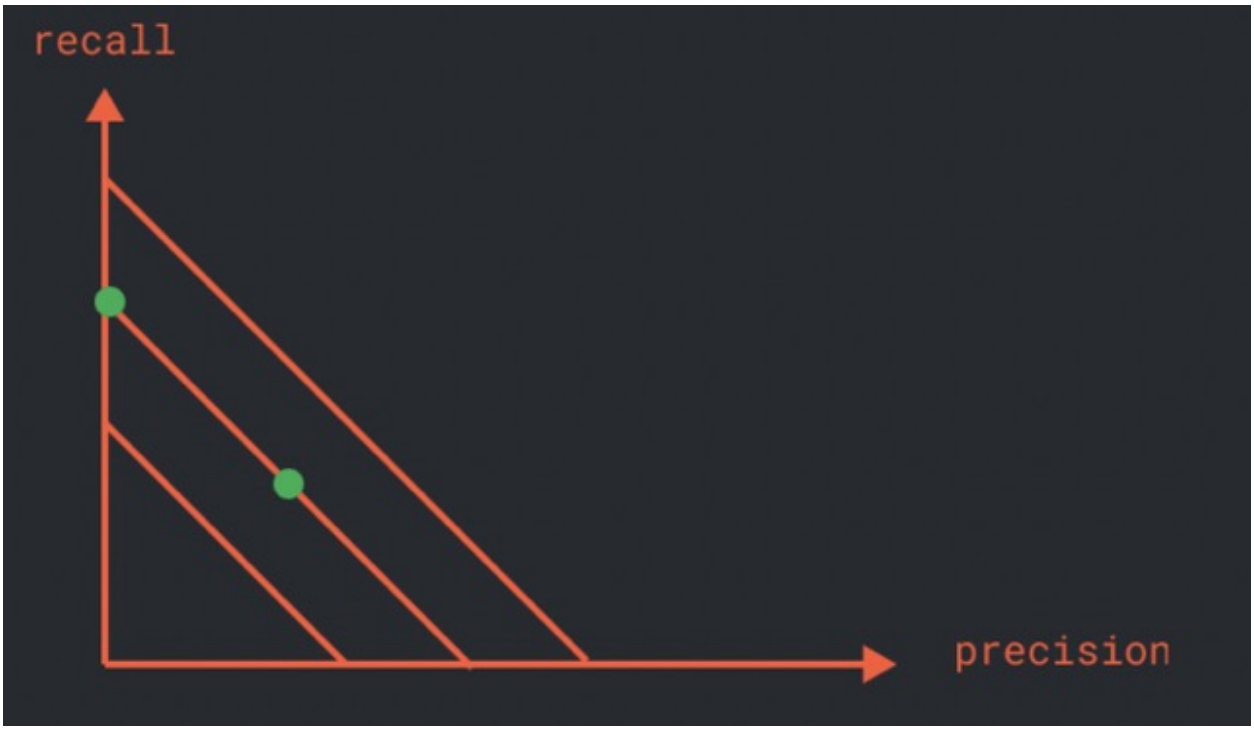
1. Посчитать арифметическое среднее: $Average = \frac{1}{2} \cdot (recall + precision)$

В чем минус такого подхода:

Пусть есть одна модель, у которой recall = 0.6, precision = 0.5, average = 0.55. В среднем неплохое значение, но и не очень хорошее. А также есть вторая модель, у которой recall = 0.8, precision = 0.3, average = 0.55. Такая модель уже будет менее

желательной. Т.е можно сделать вывод, что среднее арифметическое не чувствует разницу между умеренными и крайними случаями.

Если взять арифметическое среднее как функцию от precision и recall, нарисуем ее линии уровня в осях recall-precision, то увидим, что линии уровня будут линейны:



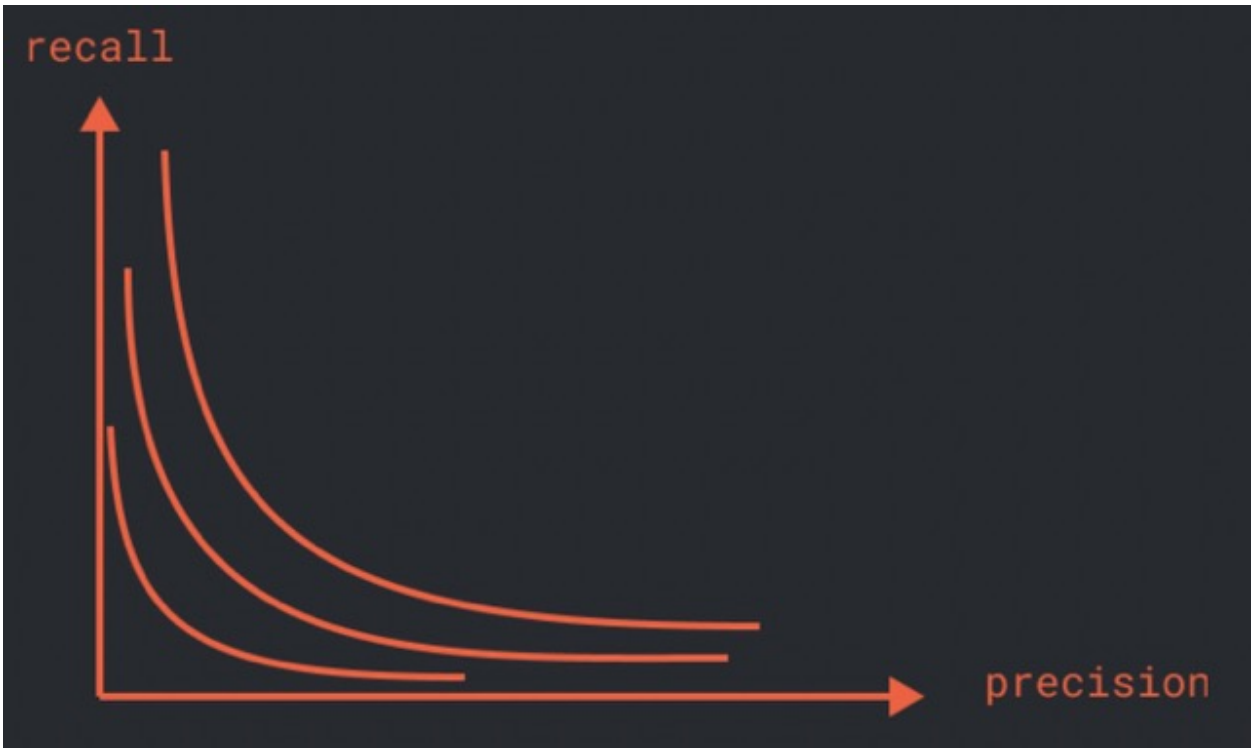
Поэтому, когда мы стоим на одной линии уровня, нам неважно: находиться в точке в центре или ближе к какой-то из осей.

2. Посчитать геометрическое среднее: $Average_g = \sqrt{recall \cdot precision}$

В чем недостаток такого подхода:

Пусть есть одна модель: recall = 0.6, precision = 0.5, Average = 0.54. А у второй модели: recall = 0.8, precision=0.3, Average = 0.49. В таком случае мы действительно получим некоторую разницу в значении среднего.

Как будут выглядеть линии уровня у геометрического среднего:



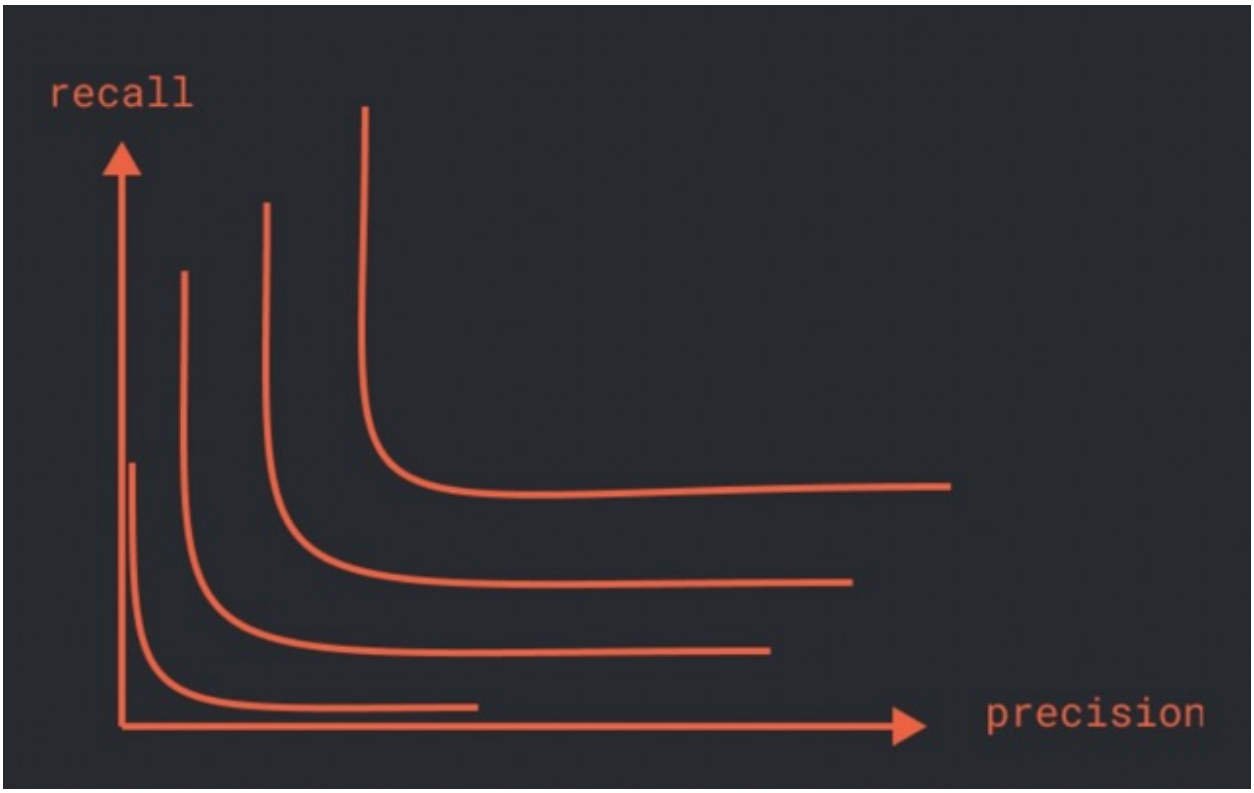
В таком случае уже нет линейных участков: модель будет выбирать precision и recall одинаково хорошие.

3. Посчитать F-меру

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

По-другому это значение еще называют средним гармоническим. F-мера достаточно близка к функции минимума: к той функции, у которой и precision, и recall должны быть достаточно большими, ведь она возвращает всего лишь минимальное значение среди двух параметров.

Давайте посмотрим на линии уровня F-меры:



Линии уровня оказались еще более пологими в зависимости от того, к какой из осей мы подходим. Это говорит о том, что F-мера еще более склонна к тому, чтобы метрики оказывались где-то посередине.

Преимущество F-меры еще в том, что можно немного изменить формулу и добавить учет предпочтения между precision и recall.

Модифицированная формула выглядит следующим образом:

$$F = \frac{(1+\beta^2)\cdot precision \cdot recall}{(\beta^2 \cdot precision)+recall}$$

Чем больше β^2, тем больше уклон в recall. И наоборот: при $0 < \beta^2 < 1$ уклон будет больше в precision.

>Как влиять на Precision и Recall

Остался вопрос: как влиять на эти метрики. Например, если мы получили модель с низким precision, но высоким recall - как изменить эту ситуацию?

Есть несколько вариантов:

1. Построить различные модели

Например, можно увеличить набор признаков. Но возможность добавить хороших фичей есть не всегда.

2. У модели линейной классификации поменять функцию активации

Напомним, что функция активации - это та функция, с помощью который мы сглаживали индикатор от отступа.

3. Менять threshold

Для начала вспомним, как устроена модель линейной классификации:

$$a(x) = sgn(\langle \beta, x \rangle) \rightarrow \{-1; +1\}$$

Чтобы понять, к какому классу будет принадлежать объект, берем скалярное произведение между признаками объекта и коэффициентами и смотрим на знак результата.

Также мы научились из выражения скалярного произведения для каждого объекта получать вероятность принадлежности положительному классу.

$$P(y_i = +1|x_i) = \frac{1}{1+e^{-(\beta, x_i)}} \rightarrow [0; 1]$$

Оказывается, что оба варианта связаны следующей формулой:

$$\text{Если} P(y_i = +1|x_i) > 0.5, \text{то } a(x_i) = +1$$

Т.е. когда вероятность больше 0.5, то модель возвращает положительный результат.

Идея: манипулировать уверенностью алгоритма, т.е. значением вероятности, переступая порог которого объект относится к положительному классу.