Шпаргалка по метрикам и проблемам переобучения / Концепции Cheatsheet (XeLaTeX)

Краткий справочник

April 5, 2025

Contents

1 **Метрики Оценки: Регрессия** 1

2 **Метрики Оценки: Классификация** 1

3 **Статистическая Оценка Надежности** 2
- 3.1 Дисперсия (Variance) . . . 3
- 3.2 Доверительные Интервалы (Confidence Intervals, CI) . . . 3
- 3.3 Статистическая Значимость и p-value . . . 3
- 3.4 Статистические Тесты (z, t, Mann-Whitney) . . . 3

1 Метрики Оценки: Регрессия

Зачем нужны метрики?

Метрики — это численные показатели, позволяющие **объективно оценить качество** работы модели машинного обучения. Для задач регрессии (предсказание непрерывного значения, например, цены дома или температуры) используются свои метрики.

Основные метрики регрессии

Пусть y_i — истинное значение, а $\hat{y}_i$ — предсказанное моделью значение для i-го объекта, n — количество объектов, $\bar{y}$ - среднее истинных значений.

- **MAE (Mean Absolute Error) / Средняя Абсолютная Ошибка**: Показывает среднее абсолютное отклонение предсказаний от факта. Легко интерпретируется в единицах целевой переменной. Менее чувствительна к выбросам, чем MSE.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- **MSE (Mean Squared Error) / Среднеквадратичная Ошибка**: Среднее квадратов отклонений. Сильнее штрафует за большие ошибки из-за возведения в квадрат. Используется в оптимизации многих моделей. Единицы измерения - квадрат исходных единиц.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- **RMSE (Root Mean Squared Error) / Корень из Среднеквадратичной Ошибки**: Корень из MSE. Возвращает метрику к исходным единицам измерения, что упрощает интерпретацию. Как и MSE, чувствительна к выбросам.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- **R^2 (Коэффициент Детерминации)**: Показывает, какую долю дисперсии зависимой переменной объясняет модель по сравнению с простой моделью, всегда предсказывающей среднее. Значения от $(-\infty)$ до 1. Ближе к 1 — лучше. 0 — модель работает как среднее. Отрицательные значения — модель хуже среднего.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Аналогия R^2: Представьте, что вы пытаетесь предсказать рост людей. Если вы всегда предсказываете средний рост (простая модель), R^2 будет 0. Если ваша модель идеально предсказывает рост каждого, R^2 = 1.

2 Метрики Оценки: Классификация

Матрица ошибок (Confusion Matrix)

Основа для большинства метрик бинарной классификации. Показывает, сколько объектов какого класса и как были классифицированы.

- **TP (True Positive)**: Истинно положительные. Класс 1, предсказан как 1. (Нашли больного)
- **TN (True Negative)**: Истинно отрицательные. Класс 0, предсказан как 0. (Нашли здорового)
- **FP (False Positive)**: Ложно положительные. **Ошибка I рода**. Класс 0, предсказан как 1. (Здоровый признан больным)
- **FN (False Negative)**: Ложно отрицательные. **Ошибка II рода**. Класс 1, предсказан как 0. (Больной признан здоровым)

Матрица Ошибок:

	Предсказание: 1	**Предсказание: 0**
Реальность: 1	TP	FN
Реальность: 0	FP	TN

Основные метрики классификации

- **Accuracy (Доля правильных ответов)**: Общая доля верных предсказаний. **Плохо работает при дисбалансе классов!**

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

Аналогия: Если 99% писем - не спам, модель, всегда говорящая "не спам", будет иметь Accuracy 99

- **Precision (Точность)**: Какая доля объектов, названных моделью классом 1, действительно являются классом 1? Важна, когда цена FP высока (напр., отправка здорового на дорогую операцию).

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

- **Recall (Полнота, Sensitivity, True Positive Rate - TPR)**: Какую долю объектов класса 1 модель смогла правильно найти? Важна, когда цена FN высока (напр., пропуск больного пациента или мошеннической транзакции).

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

- **F1-мера (F1-Score)**: Гармоническое среднее Precision и Recall. Полезна, когда важен баланс между точностью и полнотой. Стремится к нулю, если хотя бы одна из метрик (Precision или Recall) близка к нулю.

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}$$

Можно использовать F_β**-меру** для придания большего веса Precision ($\beta < 1$) или Recall ($\beta > 1$).

- **Specificity (Специфичность, True Negative Rate - TNR)**: Какую долю объектов класса 0 модель верно определила?

$$\text{Specificity} = \frac{TN}{TN + FP}$$

- **False Positive Rate (FPR)**: Какую долю объектов класса 0 модель неверно назвала классом 1? $FPR = 1 - \text{Specificity}$.

$$\text{FPR} = \frac{FP}{TN + FP}$$

Аналогия Precision/Recall (Спам-фильтр):

- **Precision**: Из всех писем, что попали в папку "Спам", какая доля реально спам? (Не хотим терять важные письма - высокий Precision).
- **Recall**: Из всех реально спамовых писем, какая доля попала в папку "Спам"? (Хотим отловить как можно больше спама - высокий Recall).

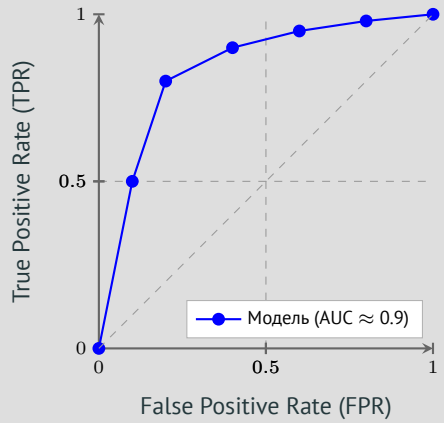
ROC AUC (Receiver Operating Characteristic Area Under Curve)

Показывает качество модели в задаче **ранжирования** классов, независимо от выбранного порога классификации.

- **ROC-кривая**: График зависимости **TPR (Recall)** от **FPR** при изменении порога классификации от 1 до 0.
- **AUC (Area Under Curve)**: Площадь под ROC-кривой. Варьируется от 0 до 1.
  - AUC = 1: Идеальный классификатор.
  - AUC = 0.5: Случайное угадывание (модель бесполезна, диагональная линия).
  - AUC < 0.5: Модель работает хуже случайной (возможно, перепутаны метки классов).
- **Интерпретация AUC**: Вероятность того, что случайно выбранный объект класса 1 получит от модели оценку выше (более высокую вероятность принадлежности к классу 1), чем случайно выбранный объект класса 0.
- **Преимущества**: Относительная устойчивость к дисбалансу классов (по сравнению с Accuracy). Позволяет сравнить модели в целом, без привязки к конкретному порогу.

Аналогия ROC AUC: Представьте соревнование: модели нужно выстроить всех людей в ряд так, чтобы все "больные" (класс 1) оказались правее всех "здоровых" (класс 0). AUC показывает, насколько хорошо модель справляется с этой задачей ранжирования.

Пример ROC-кривой



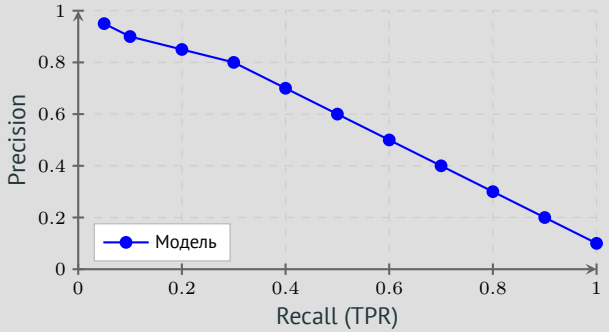
Precision-Recall AUC (PR AUC)

Альтернатива ROC AUC, особенно полезная при **сильном дисбалансе классов**, когда важнее всего найти объекты редкого положительного класса.

- **PR-кривая**: График зависимости **Precision** от **Recall (TPR)** при изменении порога классификации.
- **PR AUC**: Площадь под PR-кривой. Также от 0 до 1.
- **Почему при дисбалансе?**: ROC AUC может быть обманчиво высоким при дисбалансе, так как TN обычно много, и FPR остается низким. PR-кривая фокусируется на поиске редкого положительного класса (TP) и цене ошибок на нем (FP), что важнее при дисбалансе.
- **Baseline**: В отличие от ROC AUC (baseline 0.5), baseline для PR AUC зависит от доли положительного класса P в выборке: baseline $\approx P/(P+N)$. Для сильно несбалансированной выборки baseline PR AUC близок к 0.

Аналогия PR AUC: Представьте поиск иголок (класс 1) в стоге сена (все данные). PR-кривая показывает: при разной степени "старания" (меняем порог -> меняется Recall), насколько точны наши находки (Precision)? Насколько много мусора (FP) мы захватываем вместе с иголками?

Пример PR-кривой



Выбор метрики

Выбор метрики **критически зависит от бизнес-задачи**!

- **Медицинская диагностика (опасная болезнь):** Важнее найти всех больных (высокий **Recall**), даже если будут ложные срабатывания (низкий Precision). Цена FN (пропустить больного) очень высока. Используем Recall, F-меру с $\beta > 1$, PR AUC.
- **Спам-фильтр:** Важнее не отправлять нужные письма в спам (высокий **Precision**), даже если часть спама просочится (не идеальный Recall). Цена FP (потерять важное письмо) высока. Используем Precision, F-меру с $\beta < 1$.
- **Предсказание кликов (реклама):** Часто интересует общая точность предсказания вероятности клика, могут использовать **LogLoss** или **ROC AUC**.
- **Сильный дисбаланс классов (поиск мошенников):** Accuracy бесполезна. Смотреть на **F1-меру**, **PR AUC**, матрицу ошибок, Precision, Recall.

Всегда обсуждайте с заказчиком или продакт-менеджером, **какая ошибка для них страшнее** и как модель будет использоваться!

Кратко: Online vs Offline метрики

- **Offline метрики**: Рассчитываются на отложенной (исторической) выборке (например, на тестовом датасете). Это все метрики, рассмотренные выше (Accuracy, F1, AUC, MSE и т.д.). Позволяют оценить модель до выкатки в продакшен.
- **Online метрики**: Рассчитываются на реальных данных после внедрения модели в работающую систему. Это обычно **бизнес-метрики**: CTR (Click-Through Rate), конверсия в покупку, средний чек, время на сайте, отток клиентов и т.д. Оцениваются и сравниваются с помощью **A/B тестирования**.

3 Статистическая Оценка Надежности

Зачем нужна статистика в ML?

В ML мы почти всегда работаем с **ограниченными выборками** данных. Любая метрика (AUC, F1, MSE), посчитанная на такой выборке, является лишь **оценкой** истинного значения, которое мы получили бы на всех возможных данных (генеральной совокупности). Эта оценка подвержена **случайной изменчивости** (sampling variability). Статистические методы помогают:

- Оценить **разброс** и **неопределенность** наших данных и метрик.
- Понять, насколько **надежны** наши выводы (например, действительно ли модель Б лучше модели А, или разница случайна?).
- Принимать обоснованные решения на основе данных (например, при A/B тестировании).

3.1 Дисперсия (Variance)

Что такое Дисперсия?

Дисперсия — это мера того, насколько сильно значения в наборе данных **разбросаны** относительно их среднего значения ($\bar{x}$).

$$\text{Sample Variance } (s^2) \approx \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

(Используем $n - 1$ для несмещенной оценки дисперсии генеральной совокупности по выборке). **Стандартное отклонение (Standard Deviation, SD, s)** — это корень из дисперсии ($s = \sqrt{s^2}$). Оно измеряется в тех же единицах, что и исходные данные, и его легче интерпретировать.
Интуиция:

- **Низкая дисперсия/SD**: Данные сгруппированы близко к среднему.
- **Высокая дисперсия/SD**: Данные сильно разбросаны.

Зачем в ML?

- Понимание изменчивости признаков и целевой переменной.
- **Variance** в "Bias-Variance Tradeoff": относится к чувствительности *модели** к изменениям в обучающих данных (высокая дисперсия модели = переобучение).
- Компонент для расчета R^2.
- Оценка гомоскедастичности (постоянства дисперсии ошибок) в регрессии.
- Входной параметр для некоторых стат. тестов (t-test).

3.2 Доверительные Интервалы (Confidence Intervals, CI)

Что такое Доверительный Интервал?

Поскольку наша метрика (например, средний AUC по фолдам CV, средний чек в A/B тесте) посчитана по выборке, она не является точным истинным значением. **Доверительный интервал (CI)** — это диапазон значений, который с определенной долей уверенности (обычно 95%) **содержит истинное значение** параметра генеральной совокупности.

Формула (упрощенно для среднего):

$$CI = \text{Выборочная Оценка} \pm \text{Критическое Значение} \times \text{Стандартная Ошибка Оценки}$$

- **Выборочная Оценка**: Среднее значение метрики по выборке/фолдам.
- **Критическое Значение**: Зависит от уровня доверия (e.g., 1.96 для 95% CI при использовании z-статистики/нормального распределения, или значение из t-распределения для t-статистики).
- **Стандартная Ошибка (Standard Error, SE)**: Мера точности выборочной оценки (насколько сильно она может варьироваться от выборки к выборке). Обычно это $SE = \frac{SD}{\sqrt{n}}$, где SD - стандартное отклонение данных, n - размер выборки.

Интерпретация 95% CI [A, B]: "Мы на 95% уверены, что истинное значение параметра (например, средний AUC) находится между A и B". *Более строго:* Если бы мы многократно повторяли наше исследование (брали новые выборки того же размера), то 95% построенных таким образом доверительных интервалов содержали бы истинное значение параметра.
Зачем в ML?

- Понять **точность и надежность** оценки метрики (широкий CI = большая неопределенность).
- Сравнить модели или группы в A/B тесте: Если мы строим CI для **разницы** между метриками (например, $\text{AUC}_B - \text{AUC}_A$), и этот интервал **не включает ноль** (например, [0.01, 0.05]), это говорит о статистически значимом различии между моделями/группами на выбранном уровне доверия.

Аналогия CI: Вы ловите рыбу сетью (строите CI). Вы не знаете точно, где рыба (истинное значение). Но вы знаете, что ваша сеть (метод построения CI) достаточно хороша, чтобы в 95% случаев поймать рыбу, если вы будете забрасывать ее снова и снова.

3.3 Статистическая Значимость и p-value

Гипотезы и Значимость

Статистическая значимость помогает определить, является ли наблюдаемый эффект (например, разница в метриках между моделями А и Б, или разница в конверсии между вариантами А и В в тесте) **реальным** или он мог возникнуть **случайно** из-за вариативности выборки.
Процесс проверки гипотез:

1. **Формулируем Нулевую Гипотезу (H_0):** Гипотеза об *отсутствии эффекта* или разницы (e.g., H_0 : $\text{AUC}_A = \text{AUC}_B$, H_0 : $\text{Конверсия}_A = \text{Конверсия}_B$). Это статус-кво, который мы пытаемся опровергнуть.
2. **Формулируем Альтернативную Гипотезу (H_1 или H_a):** Гипотеза о *наличии эффекта* (e.g., H_1 : $\text{AUC}_A \neq \text{AUC}_B$ (двусторонняя), или H_1 : $\text{AUC}_B > \text{AUC}_A$ (односторонняя)).
3. **Выбираем Уровень Значимости (α):** Порог для принятия решения. Обычно $\alpha = 0.05$ (5%). Это максимальная вероятность **Ошибки I рода**, которую мы готовы допустить (т.е., отвергнуть H_0, когда она на самом деле верна - "ложная тревога").
4. **Собираем данные и вычисляем Статистику Теста** (например, z-статистику, t-статистику). Это число, которое измеряет, насколько наши данные отклоняются от того, что ожидалось бы при H_0.
5. **Вычисляем p-value.**

p-value (Уровень Значимости): Вероятность получить наблюдаемые данные (или еще более экстремальные результаты), **если предположить, что Нулевая Гипотеза (H_0) верна**.
Интерпретация p-value:

- **p-value < α**: Наблюдаемые данные *очень маловероятны*, если H_0 верна. Мы **отвергаем** H_0 в пользу H_1. Результат считается **статистически значимым** на уровне α. (Есть основания полагать, что эффект реален).
- **p-value >= α**: Наблюдаемые данные *вполне совместимы* с H_0. Мы **не можем отвергнуть** H_0. Результат **не является статистически значимым**. (**Важно**: Это не доказывает, что H_0 верна!)

Аналогия p-value: Суд над H_0 ("нет разницы"). p-value — сила улик против H_0. Если улик мало (p >= α), H_0 "оправдывают" (не отвергают). Если улик много (p < α), H_0 "осуждают" (отвергают).
Ошибки:

- **Ошибка I рода (α)**: Отвергнуть H_0, когда она верна (False Positive).
- **Ошибка II рода (β)**: Не отвергнуть H_0, когда она ложна (False Negative).
- **Мощность теста (Power = $1 - \beta$)**: Вероятность правильно отвергнуть ложную H_0 (обнаружить реальный эффект).

3.4 Статистические Тесты (z, t, Mann-Whitney)

Как выбрать тест?

Выбор теста зависит от:

- **Типа данных**: Непрерывные (средние), категориальные (доли/пропорции), порядковые (ранги).
- **Цели сравнения**: Сравнение с константой, сравнение двух групп, сравнение более двух групп.
- **Зависимости выборок**: Независимые (разные группы людей/объектов) или Зависимые/Парные (одни и те же объекты до/после, **результаты моделей на одних и тех же фолдах CV**).
- **Распределения данных**: Нормальное или нет.
- **Знания о дисперсии**: Известна или нет (почти всегда неизвестна).
- **Размера выборки**.

Z-тест

- **Что проверяет**: Разницу между средними или долями (пропорциями).
- **Ключевые предположения**:
  - Дисперсия генеральной совокупности **известна** (очень редко).
  - ИЛИ размер выборки **очень большой** ($n > 30..50$), что позволяет использовать ЦПТ (Центральную Предельную Теорему) и считать выборочное среднее/долю нормально распределенными.
- **Когда использовать в ML?**:
  - **Сравнение долей (конверсий, CTR) в A/B тестах** при больших размерах выборок (тысячи пользователей).

T-тест

- **Что проверяет**: Разницу между **средними** значениями.
- **Ключевые предположения**:
  - Дисперсия генеральной совокупности **неизвестна** (оценивается по выборке).
  - Данные (или разности для парного теста) должны быть примерно **нормально распределены**. Тест довольно устойчив к небольшим отклонениям от нормальности при умеренных/больших выборках ($n > 15..30$).
  - Для независимого t-теста: гомоскедастичность (равенство дисперсий в группах). Если нет - используется **t-тест Уэлча (Welch's t-test)**, который не требует равенства дисперсий (часто используется по умолчанию в ПО).
- **Виды и Когда использовать в ML?**:
  - **Одновыборочный (One-sample t-test)**: Сравнение среднего выборки с известным значением (редко в ML).
  - **Независимый (Independent two-sample t-test)**: Сравнение средних двух **независимых** групп (например, средний чек в контрольной и тестовой группах A/B теста, если данные примерно нормальны и выборки не гигантские).
  - **Парный (Paired t-test)**: Сравнение средних двух **зависимых** (парных) измерений. **КРИТИЧЕСКИ ВАЖНО ДЛЯ ML**:
    * Сравнение метрик (например, AUC, F1) двух моделей, посчитанных на **одних и тех же фолдах кросс-валидации**. Мы смотрим на попарные разности метрик на каждом фолде и проверяем, значимо ли среднее этой разности отличается от нуля.
    * Оценка эффекта "до/после" на одних и тех же пользователях/объектах.

Тест Манна-Уитни (Mann-Whitney U test / Wilcoxon Rank-Sum test)

- **Что проверяет**: Разницу между **распределениями** двух **независимых** выборок. Часто интерпретируется как тест на различие **медиан**. Является **непараметрическим** аналогом независимого t-теста.
- **Ключевые предположения**:
  - Выборки независимы.
  - Данные как минимум **порядковые** (ordinal) или непрерывные.
  - **НЕ требует нормальности распределения!**
  - Для интерпретации как теста на медианы, предполагается, что формы распределений в группах схожи.
- **Когда использовать в ML?**:
  - Сравнение двух **независимых** групп в A/B тесте, когда данные **сильно ненормальны** (например, время на сайте, доход пользователя, количество покупок - часто имеют выбросы и скошенность) или являются порядковыми (оценки 1-5 звезд).
  - Когда предположение о нормальности для t-теста явно нарушено.
- **Примечание**: Для **парных** данных, когда нарушена нормальность, используется **Тест Уилкоксона для связанных выборок (Wilcoxon signed-rank test)**.

Итог по статистике

Понимание дисперсии, доверительных интервалов и методов проверки статистической значимости (p-value, тесты) абсолютно необходимо для корректной интерпретации результатов работы ML моделей, сравнения их между собой и оценки влияния изменений в A/B тестах. Выбор правильного инструмента зависит от данных и задачи.

Кросс-валидация (Cross-Validation, CV)

Метод оценки обобщающей способности модели и получения более надежной оценки метрики, чем на единственном тест-сплите. Помогает бороться с переобучением и оценить стабильность модели.

- **Идея**: Разделить обучающую выборку на K непересекающихся частей (фолдов). Поочередно использовать $K-1$ часть для обучения модели и 1 оставшуюся часть для валидации (расчета метрики). Повторить K раз, каждый раз меняя валидационный фолд. Итоговая оценка метрики — среднее значение по всем K фолдам. Также смотрят на стандартное отклонение метрики по фолдам для оценки стабильности.
- **K-Fold CV**: Самый распространенный вид. Данные делятся на K фолдов примерно одинакового размера (часто K=5 или K=10).
- **Stratified K-Fold CV**: Вариант K-Fold для задач **классификации**, особенно при **дисбалансе классов**. Гарантирует, что в каждом фолде сохраняется исходное соотношение (стратификация) классов. **Использовать по умолчанию для классификации!**
- **Leave-One-Out CV (LOOCV)**: Частный случай K-Fold, где $K = n$ (количество объектов). Каждый объект по очереди используется как валидационный сет. Долго, но дает почти несмещенную оценку ошибки. Используется редко, на очень маленьких данных.

Аналогия K-Fold: Подготовка к экзамену. У вас есть 5 тем (K=5). Вы 5 раз готовитесь: 1 раз учите темы 1,2,3,4 и отвечаете по теме 5; потом учите 1,2,3,5 и отвечаете по 4, и т.д. Итоговая оценка — среднее по 5 "экзаменам".

Проблема Дисбаланса Классов

Ситуация, когда объектов одного класса значительно больше, чем другого (например, 99

- **Проблема**:
  - Accuracy становится бесполезной метрикой.
  - Модель может "научиться" всегда предсказывать мажоритарный класс и иметь высокую Accuracy.
  - Стандартный K-Fold может привести к фолдам без (или с очень малым числом) объектов миноритарного класса.
- **Основные подходы к решению**:
  1. **Выбор правильной метрики**: Использовать **Precision, Recall, F1-меру, ROC AUC, PR AUC**. Анализировать **матрицу ошибок**.
  2. **Изменение выборки (Resampling)**:
     - **Undersampling**: Удаление части объектов мажоритарного класса. Риск потери информации.
     - **Oversampling**: Дублирование объектов миноритарного класса. Риск переобучения на дубликатах.
     - **SMOTE (Synthetic Minority Over-sampling Technique)** и его варианты: Генерация *синтетических* объектов миноритарного класса на основе их соседей. Часто работает лучше простого oversampling.

     Внимание! Методы изменения выборки (Under/Oversampling, SMOTE) должны применяться **только к обучающей части данных внутри каждого фолда кросс-валидации**, но **никогда** к валидационной или тестовой выборке, чтобы избежать утечки данных (data leakage).
  3. **Взвешивание классов (Class Weighting)**: Назначение большего веса объектам миноритарного класса в функции потерь модели при обучении. Многие алгоритмы (логистическая регрессия, SVM, деревья решений, градиентный бустинг) поддерживают это (например, параметр `class_weight='balanced'` или `scale_pos_weight` в scikit-learn и XGBoost/LightGBM).
  4. **Использование ансамблей**: Специальные методы ансамблирования, учитывающие дисбаланс (например, EasyEnsemble, BalanceCascade).
  5. **Использовать Stratified K-Fold** при кросс-валидации (как уже упоминалось).