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

Краткий справочник

April 2, 2025

Contents

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).

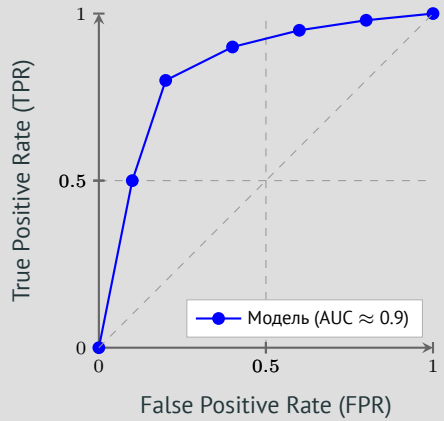
ROC AUC (Receiver Operating Characteristic Area Under Curve)

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



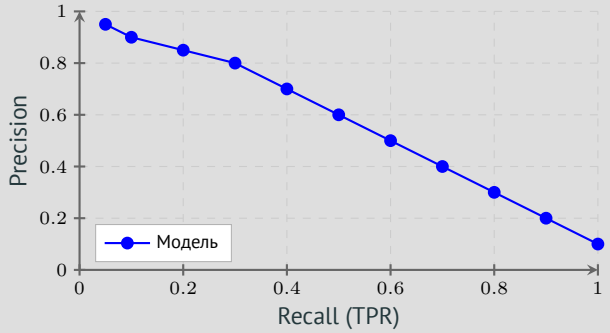
Precision-Recall AUC (PR AUC)

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 Валидация и Надежность Оценки

Статистическая Значимость [ОЧЕНЬ ВАЖНО

] Допустим, модель А дала AUC 0.85, а модель Б - AUC 0.86 на тестовой выборке. Значит ли это, что Б *действительно* лучше? Не обязательно! Различие может быть случайным из-за ограниченности тестовой выборки. Для проверки нужны стат. тесты.

- **Статистическая гипотеза**: Проверяем нулевую гипотезу H_0: "Модели А и Б имеют одинаковое качество (разница в метриках случайна, AUC $=$ AUC)". Альтернативная гипотеза H_1: "Модель Б действительно лучше (AUC $>$ AUC)".
- **p-value (Уровень значимости)**: Вероятность получить наблюдаемую (или еще большую) разницу в метриках **при условии, что нулевая гипотеза верна** (т.е., если на самом деле разницы нет).
  - **p-value < α** (часто $\alpha = 0.05$): Считаем результат **статистически значимым** на уровне α. Мы отвергаем H_0. Есть основания полагать, что модель Б действительно лучше.
  - **p-value >= α**: Результат **не является статистически значимым**. Мы не можем отвергнуть H_0. Наблюдаемая разница могла возникнуть случайно.
- **Confidence Interval (CI) / Доверительный Интервал**: Диапазон значений, который с определенной вероятностью (обычно 95

Почему это важно? Чтобы не принимать бизнес-решения (например, о внедрении новой модели, изменении продукта) на основе случайных колебаний метрик. Это **основа для интерпретации результатов A/B тестов** и сравнения моделей на offline-выборках. *Аналогия p-value*: Суд над гипотезой H_0 ("разницы нет"). p-value - это сила улик против H_0. Если улик мало (p-value большое, >= α), мы не можем "осудить" H_0 (не отвергаем). Если улик много (p-value маленькое, < α), мы "осуждаем" H_0 (отвергаем) и принимаем H_1.

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).