Шпаргалка по базовым понятиям в ML / Концепции Cheatsheet (XeLaTeX)

Краткий справочник

Содержание

Определение Машинного Обучения (ML)

Машинное обучение — это область искусственного интеллекта, изучающая методы построения алгоритмов, способных обучаться на основе данных. Вместо явного программирования правил, ML-модели самостоятельно выявляют закономерности в предоставленных данных и используют их для решения поставленных задач (например, классификации, регрессии, кластеризации).

1 Типы Машинного Обучения

1.1. A Обучение с Учителем (Supervised Learning)

Обучение на размеченных данных

Задача: Модель обучается на наборе данных, где для каждого объекта (*примера*) заданы входные **признаки (features)** и соответствующий правильный выход (**метка класса** или **целевое значение**, label/target). **Цель:** Построить модель, способную предсказывать метку/значение для новых, ранее не виданных объектов по их признакам.

Подтипы Supervised Learning

- **Классификация (Classification):** Предсказание категориальной метки. Выход модели принадлежит к дискретному множеству классов. *Примеры:* Определение спама в письмах (спам/не спам), распознавание изображений (кошка/собака/птица), кредитный скоринг (одобрить/отклонить).
- **Регрессия (Regression):** Предсказание непрерывного числового значения. *Примеры:* Прогнозирование цены недвижимости, оценка температуры воздуха, предсказание спроса на товар.

1.2. B Обучение без Учителя (Unsupervised Learning)

Поиск структуры в неразмеченных данных

Задача: Модель обучается на данных без каких-либо меток или целевых значений. Алгоритм должен самостоятельно найти внутренние закономерности, структуру или взаимосвязи в данных.

Задачи Unsupervised Learning

- **Кластеризация (Clustering):** Разделение набора данных на группы (кластеры) схожих между собой объектов. Объекты внутри одного кластера должны быть более похожи друг на друга, чем на объекты из других кластеров. *Примеры:* Сегментация клиентов по покупательскому поведению, группировка новостных статей по темам.
- **Снижение размерности (Dimensionality Reduction):** Уменьшение количества признаков в данных при сохранении максимально возможного объема полезной информации. Используется для визуализации, сжатия данных или подготовки данных для других ML-алгоритмов. *Примеры:* Метод главных компонент (PCA), t-распределенное стохастическое вложение соседей (t-SNE).
- **Поиск аномалий (Anomaly Detection):** Выявление объектов, которые значительно отличаются от основной массы данных. *Примеры:* Обнаружение мошеннических транзакций, выявление дефектных изделий.

1.3. C Обучение с Подкреплением (Reinforcement Learning, RL)

Обучение через взаимодействие со средой

Задача: Агент (модель) учится принимать последовательность **действий** в некоторой **среде** с целью максимизации кумулятивной **награды** (reward), получаемой от среды в ответ на действия. Обучение происходит методом проб и ошибок. *Примеры:* Обучение игровых ботов (шахматы, Go, видеоигры), управление роботами, оптимизация торговых стратегий, персонализированные рекомендации.

2 II. Процесс Разработки и Разделение Данных

2.1. A Основные Этапы ML-проекта

Типичный жизненный цикл ML-модели

Процесс создания ML-решения обычно включает следующие шаги (могут итерироваться):

1. **Определение проблемы и цели:** Четкая постановка бизнес-задачи и метрик успеха.
2. **Сбор данных:** Получение релевантных данных для обучения.
3. **Анализ и предварительная обработка данных (EDA & Preprocessing):** Очистка, исследование, обработка пропусков, кодирование категорий, создание новых признаков (Feature Engineering). *(Часто наиболее трудоемкий этап).*
4. **Выбор модели(ей):** Подбор подходящих алгоритмов для задачи.
5. **Обучение модели:** Подбор параметров модели на обучающей выборке.
6. **Настройка гиперпараметров и выбор лучшей модели:** Использование валидационной выборки.
7. **Оценка качества:** Финальная оценка на тестовой выборке.
8. **Развертывание (Deployment):** Внедрение модели в рабочую среду.
9. **Мониторинг и поддержка:** Отслеживание производительности модели и ее переобучение при необходимости.

2.2. B Разделение Данных: Train / Validation / Test

Цель разделения данных

Ключевая задача ML — построить модель, способную хорошо **обобщать** (generalize), то есть давать точные предсказания на новых, ранее не виданных данных. Чтобы объективно оценить эту способность, исходный набор данных разделяют:

1. **Обучающая выборка (Train Set):**
Используется непосредственно для *обучения* модели — поиска оптимальных значений её внутренних **параметров** (например, весов в линейной модели или нейросети). Модель "видит" эти данные и подстраивается под них. *Назначение: Найти закономерности в данных.*

2. **Валидационная выборка (Validation Set):**
Используется для *настройки* **гиперпараметров** модели (параметров, которые не обучаются напрямую, а задаются до начала обучения, например, степень полинома, learning rate, параметр регуляризации λ) и для *выбора* наилучшей модели из нескольких кандидатов. Модель не обучается на этих данных, но её производительность на них используется для принятия решений о её структуре или настройках. *Назначение: Подобрать оптимальную конфигурацию модели.*

3. **Тестовая выборка (Test Set):**
Используется *только один раз* в самом конце для получения *финальной, объективной оценки* качества лучшей выбранной и настроенной модели. Эти данные модель не должна была "видеть" ни на этапе обучения, ни на этапе настройки гиперпараметров. Результат на тестовой выборке имитирует производительность модели на реальных новых данных. *Назначение: Оценить финальную производительность выбранной модели.*

Важное правило

Категорически нельзя использовать тестовую выборку для подбора гиперпараметров или выбора модели. Это приведет к "утечке" информации из теста в процесс настройки и, как следствие, к нереалистично завышенной оценке качества модели.

3 Переобучение и Недообучение

3.1. III.A Фундаментальные Проблемы Обучения

Риски при построении модели

При обучении модели существует две основные нежелательные ситуации, связанные с её сложностью и способностью к обобщению:

Недообучение (Underfitting)

Описание: Модель слишком проста для улавливания сложных закономерностей в данных. Она не способна хорошо описать даже обучающую выборку. **Характеристики:**
- Плохое качество (высокая ошибка) как на обучающей (`Train`), так и на валидационной/тестовой (`Valid/Test`) выборках.
- Модель обладает высоким **смещением (Bias)**.

Причины: Недостаточная сложность модели (e.g., линейная модель для нелинейных данных), нерелевантные признаки.

Переобучение (Overfitting)

Описание: Модель излишне сложна и "запоминает" обучающие данные, включая случайный шум и выбросы, вместо того чтобы улавливать общие закономерности. **Характеристики:**
- Отличное качество (низкая ошибка) на обучающей выборке (`Train`).
- Значительно худшее качество (высокая ошибка) на валидационной/тестовой выборке (`Valid/Test`).
- Модель обладает высоким **разбросом (Variance)**.

Причины: Слишком сложная модель (e.g., глубокое дерево решений без ограничений, многослойная нейросеть), мало данных, "шумные" данные.

Цель

Найти "золотую середину" — модель, которая достаточно сложна, чтобы уловить основные зависимости в данных, но при этом устойчива к шуму и хорошо обобщается на новые данные.

4 Дилемма Смещения-Разброса (Bias-Variance Tradeoff)

4.1. A Компоненты Ошибки Модели

Разложение ожидаемой ошибки

Ожидаемую ошибку предсказания модели на новых данных (Expected Prediction Error) можно теоретически разложить на три составляющие:

$$\mathbb{E}[\text{Error}] = \underbrace{\text{Bias}^2}_{\text{Смещение}} + \underbrace{\text{Variance}}_{\text{Разброс}} + \underbrace{\sigma^2}_{\text{Неустранимая ошибка}}$$

- **Смещение (Bias):** Ошибка, возникающая из-за неверных предположений, заложенных в модель. Отражает, насколько *в среднем* предсказания модели отклоняются от истинного значения. Высокое смещение (**High Bias**) характерно для простых моделей, неспособных уловить сложную структуру данных (приводит к **недообучению**).
- **Разброс (Variance):** Ошибка, возникающая из-за чувствительности модели к малым изменениям в обучающей выборке. Отражает, насколько сильно будут различаться модели, обученные на разных подвыборках данных. Высокий разброс (**High Variance**) характерен для сложных моделей, которые подстраиваются под шум (приводит к **переобучению**).
- **Неустранимая ошибка (Irreducible Error, σ^2):** Минимальный уровень ошибки, присущий самим данным из-за случайного шума или скрытых факторов, который не может быть уменьшен выбором другой модели.

4.2. В Суть Дилеммы

Компромисс между Bias и Variance

Существует обратная зависимость между смещением и разбросом при изменении сложности модели:
- Увеличение сложности модели (e.g., добавление признаков, увеличение глубины дерева) обычно *уменьшает смещение*, но *увеличивает разброс*.
- Уменьшение сложности модели (e.g., упрощение, добавление регуляризации) обычно *уменьшает разброс*, но *увеличивает смещение*.

Задача: Найти оптимальную сложность модели, которая минимизирует *суммарную ошибку* ($\text{Bias}^2 + \text{Variance}$) на новых данных. Этот оптимум обычно достигается при некотором компромиссном уровне смещения и разброса.

Примеры моделей:
- *Низкая сложность (Low Variance, High Bias):* Линейная регрессия, Логистическая регрессия.
- *Высокая сложность (High Variance, Low Bias):* Неограниченные Деревья решений, K-ближайших соседей (с малым k), Нейронные сети без регуляризации.
- *Баланс (часто):* Деревья с ограничениями, Ансамбли (Random Forest, Gradient Boosting), Регуляризованные модели (Ridge, Lasso), Нейросети с регуляризацией.

5 Диагностика Моделей

5.1. А Кривые Обучения (Learning Curves)

Визуальная диагностика Bias и Variance

Кривые обучения — это графики, отображающие метрику качества модели (например, ошибку MSE, Accuracy, F1-score) в зависимости от объема обучающих данных или итерации/эпохи обучения. Обычно строятся две кривые: одна для **обучающей выборки (Train)**, другая для **валидационной выборки (Validation)**. Анализ их поведения помогает диагностировать проблемы недообучения и переобучения.

Типичные сценарии анализа кривых:

1. **Признаки Недообучения (High Bias):**
- Кривая ошибки на `Train` и `Valid` стабилизируются на *высоком* уровне.
- Разрыв (gap) между кривыми `Train` и `Valid` *небольшой*.
- Качество модели неудовлетворительное, добавление новых данных в обучение почти не улучшает ситуацию.

Возможные действия:
- Использовать более сложную модель (e.g., полиномиальные признаки, больше слоев/нейронов).
- Добавить новые, более информативные признаки (Feature Engineering).
- Уменьшить силу регуляризации (если используется).

2. **Признаки Переобучения (High Variance):**
- Кривая ошибки на `Train` находится на *низком* уровне (модель хорошо подогналась под обучение).
- Кривая ошибки на `Valid` находится на *значительно более высоком* уровне.
- Существует *большой разрыв* (gap) между кривыми `Train` и `Valid`.
- Увеличение объема обучающих данных может помочь сблизить кривые и улучшить качество на валидации.

Возможные действия:
- Собрать больше обучающих данных.
- Использовать регуляризацию (L1, L2, Dropout, etc.).
- Упростить модель (e.g., уменьшить глубину дерева, количество признаков через Feature Selection).
- Использовать методы ансамблирования (особенно Bagging).

3. **Желаемый сценарий ("Хороший баланс"):**
- Кривые `Train` и `Valid` сходятся к *низкому* уровню ошибки.
- Разрыв между кривыми *небольшой и стабильный*.

Это указывает на то, что модель адекватно уловила закономерности и хорошо обобщается.