2. ОСНОВЫ ML

В ходе этого этапа собеседования проверяются знания соискателя в различных областях ML, причем оценивается как их широта, так и глубина. Под широтой понимается способность кандидата предложить несколько решений типичной задачи ML, а под глубиной — способность подробно объяснить одно из них.

И хотя может показаться, что на собеседовании вас могут спросить буквально о чем угодно, связанном с ML, на деле же существует довольно ограниченный набор тем, которые подходят для содержательного обсуждения. Интервьюеры предпочитают задавать вопросы по таким темам, которые большинство соискателей либо изучали в университете, либо сталкивались с ними на практике — вне зависимости от отрасли, в которой они работали. Это делает собеседование более информативным и дает возможность экспертам при необходимости подменять друг друга. Нам же это позволяет выстроить свою подготовку вокруг относительно небольшого набора вопросов.

Вас, скорее всего, не будут спрашивать о новейших разработках в области ML, поскольку интервьюеры обычно сосредоточиваются, как вы уже наверняка догадались, на основах ML. Например, вас с большей вероятностью спросят о логистической регрессии или бустинге деревьев решений, чем о трансформерах, BERT или механизмах внимания.

В этой главе мы рассмотрим наиболее часто задаваемые вопросы по основам ML, на которые вам, возможно, придется ответить для успешного прохождения этого этапа собеседования. Мы также дадим довольно обобщенные, но содержательные ответы на основные теоретические вопросы. Рекомендуем дополнить приведенные ответы собственными знаниями и/или практическим опытом, подробнее затрагивая те темы, которые вам особенно близки. Также в этой главе мы укажем ряд дополнительных ресурсов, к которым вы можете обратиться для более глубокого изучения проблем.

В некоторых ответах для лучшего понимания материала приводятся изображения и математические формулы. На собеседовании воспроизводить их не требуется.

ВОПРОСЫ ДЛЯ ПОДГОТОВКИ

Мы собрали список часто задаваемых вопросов на этапе собеседования, посвященного основам ML. Он охватывает все ключевые области знаний в сфере ML, с которыми вам следует ознакомиться перед собеседованием.

Проверьте себя: выберите несколько вопросов из приведенного ниже списка. Попробуйте сначала ответить на них самостоятельно и только потом загляните в раздел с ответами!

Датасеты

2.1. Как собрать данные и подготовить обучающий датасет?

2.2. Какие проблемы могут возникнуть при сборе данных для обучения модели?

2.3. Как определить, подходит ли собранный датасет для моделирования?

2.4. Какие существуют способы обработки несбалансированных меток в датасете?

2.5. Что делать, если часть данных не размечена?

Признаки

2.6. Опишите различные типы входных признаков и их значения.

2.7. Что такое отбор/важность признаков?

2.8. Как выполняется отбор признаков?

2.9. Как бы вы поступили с отсутствующими значениями признаков?

Моделирование

2.10. Какие распространенные алгоритмы моделирования вы знаете? Назовите три алгоритма и сравните их.

2.11. Подробнее расскажите об одном из алгоритмов. Объясните, как он работает.

2.12. Что такое функция потерь? Как она оптимизируется?

2.13. Возможно ли ускорить оптимизацию алгоритма? Каковы компромиссы?

2.14. Какие существуют методы оптимизации градиентного спуска?

2.15. Как настроить гиперпараметры?

2.16. Что такое переобучение? Как определить, что модель переобучена?

2.17. Как бороться с переобучением?

2.18. Что такое регуляризация? Назовите два подхода и сравните их.

2.19. В чем разница между линейной и логистической регрессией?

2.20. Сравните несколько функций активации в нейронных сетях.

2.21. Объясните разницу между деревом решений, случайным лесом и градиентным бустингом деревьев решений (GBDT).

2.22. В чем разница между бустингом и бэггингом?

2.23. Какие существуют методы обучения без учителя? Как они работают?

2.24. Что такое метод k-средних? Как осуществляется инициализация? Каков критерий остановки?

2.25. Какие существуют методы обучения с частичным привлечением учителя (semi-supervised learning)? Как они работают?

Оценка

2.26. Какие наиболее распространенные функции потерь вы знаете? Сравните их.

2.27. Как определить, является ли функция потерь выпуклой?

2.28. Как оценить модель классификатора? Назовите три метрики и опишите, когда их следует использовать.

2.29. Как оценить качество модели регрессии?

2.30. Почему не стоит оптимизировать модель непосредственно по метрикам оценки?

2.31. Если ваша модель демонстрирует низкую эффективность, как провести отладку и повысить ее качество?

ОТВЕТЫ

Датасеты

Ответ 2.1. Как собрать данные и подготовить обучающий датасет?

Подготовка датасета — это ключевой этап в разработке системы ML. Сюда относится:

1. **Сбор данных.** Источником данных, необходимых для обучения модели, могут служить уже существующие потоки, генерируемые вашей системой, например информация о пользовательском поведении, либо новые данные, собранные в результате проведения пользовательских исследований. Необходимо использовать подходящие **методы семплирования** — например, случайную или стратифицированную выборку — в зависимости от задачи. Решите, что подходит вашей модели лучше — потоковая передача данных или пакетная обработка.
2. **Очистка данных.** Проверьте данные на наличие пропущенных или дублирующихся записей, определите выбросы, удалите нерелевантную информацию и исправьте ошибки. Если значения признаков отсутствуют, такие точки данных могут быть исключены или восстановлены с помощью, например, замены на среднее значение либо более сложных техник.
3. **Разметка данных.** В задачах, относящихся к обучению с учителем, данным необходимо присвоить метки, например, посредством сбора информации о пользовательском взаимодействии или с помощью специалистов-разметчиков.
4. **Разделение данных.** Разделите данные на обучающую, валидационную и тестовую выборки. Обучающая выборка используется для обучения модели, валидационная — для оценки ее эффективности в процессе обучения (например, для ранней остановки или настройки гиперпараметров), а тестовая — для оценки эффективности модели после завершения обучения. При кросс-валидации обучающая и валидационная выборки могут быть объединены в одну. Тестовая выборка обеспечивает независимую оценку качества модели и может использоваться для сравнения различных подходов к моделированию.
5. **Предобработка данных.** Проведите необходимую предобработку данных, например их нормализацию, масштабирование или преобразование в подходящий для модели формат.
6. **Проверка сбалансированности.** Дисбаланс в данных возникает, когда один из классов представлен значительно бóльшим количеством семплов по сравнению с другими. Это может вызвать проблемы при обучении, поскольку модель будет уделять внимание большему классу и игнорировать меньшие. При работе с несбалансированными данными сначала попробуйте обучить модель на исходном распределении. Если модель демонстрирует хорошую обобщающую способность, скорее всего, проблема некритична и дополнительных мер не требуется.

7. **Перемешивание данных.** Перемешайте данные, чтобы уменьшить смещение (предвзятость) и предотвратить обучение модели на порядковых закономерностях, если только это не является целью обучения.

Хотя построение датасета для машинного обучения обычно представляет собой последовательность определенных шагов, этот процесс не всегда линеен. В некоторых ситуациях может потребоваться возврат к предыдущим этапам или корректировка уже выполненных шагов. Например, разделить данные можно уже после этапа разметки. Или, другой вариант, вам не требуется балансировать набор данных перед его разметкой.

Ответ 2.2. Какие проблемы могут возникнуть при сборе данных для обучения модели?

В предыдущем ответе был изложен общий порядок сбора данных. Ниже представлены возможные проблемы, с которыми вы можете при этом столкнуться.

1. **Сбор данных.** Следует учитывать смещения, которые могут возникнуть в процессе сбора данных, например смещение сервинга (serving bias), и применять соответствующие меры по их нейтрализации, такие как методы «многорукого бандита». Некорректное семплирование может привести к узкому или однородному датасету, что ухудшит обобщающую способность модели и снизит разнообразие ее прогнозов.
2. **Очистка данных.** Методы очистки, включая обнаружение выбросов, дубликатов и заполнение пропущенных значений, могут повлиять на релевантность данных, особенно если их отсутствие не случайно. Это может негативно сказаться на качестве обучения модели.
3. **Разметка данных.** Нечеткие инструкции по разметке или несогласованность между разметчиками может сделать данные шумными. Чтобы сократить количество ошибок, рекомендуется использовать такие методы, как взвешенное голосование (weighted voting) или валидация разметки (annotation validation). Подробнее о данном подходе см. в *Callison-Burch, 2009*[1].
4. **Разделение данных.** Корректное разделение позволит избежать утечки данных между обучающей, валидационной и тестовой выборками. Так, например, можно разделить данные по временны́м меткам: обучающая выборка содержит все данные до определенной даты, а тестовая — после. См. также *Data Split Example / Machine Learning, 2022*.

---

[1] Список дополнительных ресурсов приведен в конце книги. — *Примеч. ред.*

5. **Предобработка данных.** Неправильная нормализация может повлиять на полезность некоторых признаков. Например, нормализация счетных признаков, у которых большинство значений — нули, нарушит процесс обучения модели.
6. **Проверка сбалансированности.** Распределение классов в данных влияет на эффективность модели, которая может медленно сходиться, быть смещенной в сторону мажоритарного класса и плохо распознавать миноритарные классы. Для решения этой проблемы можно использовать методы расширения выборки, или оверсемплинга (oversampling), и сокращения выборки, или андерсемплинга (undersampling)[1]. Однако вы можете столкнуться и с другими сложностями, например с плохой калибровкой выходных вероятностей. Один из способов улучшить калибровку — увеличить вес класса с меньшим числом семплов.

Дополнительную информацию о работе с несбалансированными датасетами см. ниже в ответе 2.4 «Какие существуют способы обработки несбалансированных меток в датасете?».

Ответ 2.3. Как определить, подходит ли собранный датасет для моделирования?

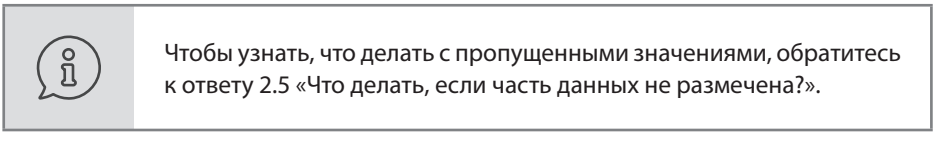
Первым шагом в создании обучающего датасета является сбор необработанных (сырых) данных, которые будут использоваться для обучения и валидации модели. Такие данные могут поступать из различных источников, например из логов, баз данных, веб-скрейпинга или даже пользовательских исследований. При сборе сырых данных необходимо учитывать несколько важных моментов.

- **Объем** имеет значение, так как напрямую влияет на эффективность модели. Большой объем данных позволяет модели лучше выявлять закономерности и эффективнее обобщать. Как правило, размер датасета

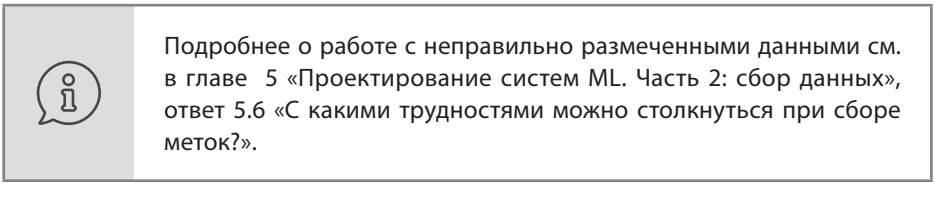
[1] В литературе по ML и анализу данных встречаются различные переводы терминов upsampling, downsampling, oversampling и subsampling, в том числе прямой калькой с английского. Во избежание путаницы именно такой вариант принят в данной книге: апсемплинг (увеличение выборки, повышающая дискретизация); даунсемплинг (уменьшение выборки, понижающая дискретизация), оверсемплинг (расширение выборки, перевыборка, повышающая выборка) и сабсемплинг (выборка подмножества, субдискретизация, подвыборка). — *Примеч. ред.*

должен быть на порядок больше числа параметров модели. Простые модели, обученные на больших датасетах, зачастую показывают лучшие результаты, чем сложные модели, обученные на небольших выборках. Так, компания Google добилась успеха в обучении простых моделей линейной регрессии на больших датасетах (см. The Size and Quality of a Data Set / Machine Learning, 2022).

- **Качество** не менее важно, поскольку недостоверные данные могут существенно снизить эффективность модели. Причины недостоверности датасета могут включать:
  - Отсутствующие значения как в признаках, так и в метках. Это может быть вызвано различными факторами, например ошибками сбора данных или повреждением данных.



  - Дублирующиеся строки в датасете могут исказить результаты и привести к переобучению.
  - Некорректные значения признаков могут появиться, если распределение признаков в обучении и в инференсе различается. По возможности используйте один и тот же код в пайплайне обучения и пайплайне сервинга. См. *Zinkevich, 2023*.
  - Некорректные метки, которые могли возникнуть из-за неправильной разметки или ошибок в процессе сбора данных.



- Выбор правильной стратегии семплирования важен для создания качественного датасета. Стратегия должна соответствовать целям и метрикам модели. Например, если цель — построение модели ранжирования, то случайной выборки может оказаться недостаточно. Гораздо лучше

подойдет выборка, отражающая метрики ранжирования, например на уровне запросов.

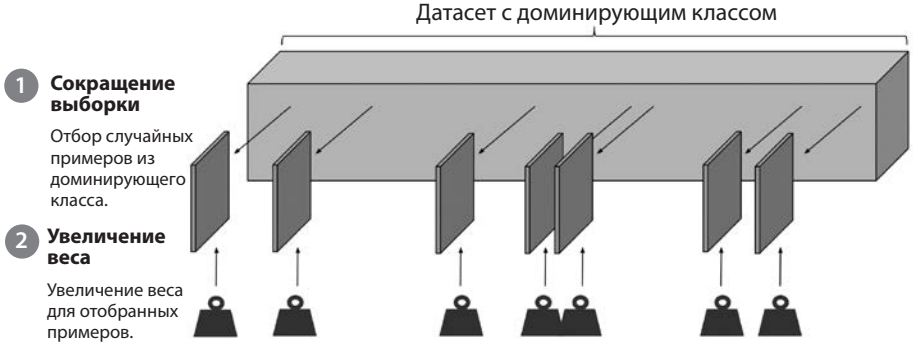
Аналогично, сильный дисбаланс классов в датасете может привести к медленной сходимости или низкому качеству прогнозов.

Дополнительную информацию о работе с дисбалансом меток см. в ответе 2.4 «Какие существуют способы обработки несбалансированных меток в датасете?».

Ответ 2.4. Какие существуют способы обработки несбалансированных меток в датасете?

Устранение дисбаланса меток в датасете позволит предотвратить смещение модели в сторону мажоритарного класса, что особенно важно, если у вас есть классы, представленные крайне малым числом семплов. Вот несколько подходов к решению данной проблемы:

1. **Ресемплирование**:
   - Расширение выборки (oversampling) миноритарного класса — дублирование семплов из миноритарного класса для достижения баланса между классами.
   - Уменьшение выборки (downsampling) мажоритарного класса — удаление части семплов из мажоритарного класса для достижения баланса между классами, что может привести к ускорению сходимости модели и снижению требований к объему хранилища данных.



Сокращение выборки (downsample) и последующее увеличение веса (upweight). Источник: *Imbalanced Data / Machine Learning, 2022*.

Вам также может понадобиться увеличить вес мажоритарного класса после сокращения выборки, что обеспечит корректную калибровку модели в процессе обучения.

2. **Генерация синтетических данных**: создание новых синтетических семплов для миноритарного класса с целью балансировки датасета, например, с помощью техники SMOTE.
3. **Обучение с учетом стоимости ошибок** (cost-sensitive learning): присвоение различного веса семплам из разных классов, чтобы скорректировать дисбаланс через функцию потерь, используемую при обучении модели.
4. **Ансамблевые методы**: использование нескольких моделей с различными стратегиями семплирования или ансамбля моделей, обученных на разных подвыборках данных с применением бустинга или бэггинга.
5. **Моделирование с акцентом на точность** (precision-oriented modeling): некоторые модели лучше подходят для оценки по метрикам, чувствительным к дисбалансу классов, таким как precision или precision@k. Например, в деревьях решений и их производных, таких как случайные леса, градиентный бустинг и др., можно проредить пути с высокой энтропией или с малым числом семплов. Деревья также можно разбить на списки решений. В некоторых задачах понимания естественного языка (NLU), например в классификации намерений с ограничениями по типу данных, даже основанные на правилах модели могут показывать высокую эффективность. А модели, выдающие вероятностные прогнозы, например логистическая регрессия, могут использовать пороговые значения для достижения желаемых результатов.

   Выбор способа устранения дисбаланса меток зависит от конкретной задачи и структуры датасета. Кроме того, можно комбинировать разные подходы для достижения наилучшего результата. Тем не менее важно помнить, что иногда наилучшей стратегией является отсутствие каких-либо изменений, особенно если у вас уже есть необходимый объем данных и модель обладает достаточной мощностью.

Содержательное исследование по данной теме см. в *López et al., 2013*.

Ответ 2.5. Что делать, если часть данных не размечена?

С отсутствующими метками в датасете можно справляться несколькими способами.

1. **Разметка данных**: самый простой способ получить корректные метки, однако довольно дорогостоящий и трудоемкий.
2. **Удаление неразмеченных данных**: такой подход применим лишь в случае незначительного числа пропущенных меток, поскольку может привести к потере информации.
3. **Импутация (восполнение) меток**: замена отсутствующих значений рассчитанными на основе доступных данных, например средним значением или модой. Так, в сильно несбалансированных датасетах (например, кликах на веб-страницах) можно восполнить метки мажоритарного класса. Однако такой подход может привести к смещению и, следовательно, к снижению качества модели.
4. Прогнозирование отсутствующих меток с использованием моделирования (**индукция**): обучение на размеченных данных одной (самообучение) или нескольких моделей (совместное обучение) с последующим их использованием для прогнозирования пропущенных меток. Минус данной стратегии заключается в том, что в обученных моделях могут закрепиться ошибочные прогнозы.
5. Прогнозирование отсутствующих меток с использованием разбиения (**трансдукция**): вместо обучения модели только на размеченных данных используются как размеченные, так и неразмеченные данные. Это может быть реализовано с помощью: (а) кластеризации с частичным привлечением учителя; (б) графовых методов, таких как алгоритмы распространения меток (label propagation algorithm, LPA); (в) обучения на базе многообразий (manifold learning), при котором близкие точки в пространстве низкой размерности получают схожие прогнозы.
6. Если данные отсутствуют на уровне отдельных инстансов, но доступны агрегированные метки на уровне групп, можно применить техники для построения классификатора на уровне инстансов с использованием групповых меток. Подробнее см. в главе 7 «Вопросы продвинутого уровня», раздел «Обучение без меток».