**1. Общий пайплайн CV-проекта**

**1.1. Методология CRISP-DM/CRISP-DS**

* **CRISP-DM (Cross-Industry Standard Process for Data Mining)**: Это методология, состоящая из шести этапов: понимание бизнеса, понимание данных, подготовка данных, моделирование, оценка и развертывание.
* **CRISP-DS (Cross-Industry Standard Process for Data Science)**: Расширенная версия CRISP-DM, включающая дополнительные этапы, такие как управление проектом и коммуникация результатов.

**1.2. Формулирование гипотезы о решении задачи**

* **Компоненты гипотезы**:
  + **Данные**: Какие данные будут использоваться для решения задачи.
  + **Модель/алгоритм**: Какой алгоритм или модель будет применяться.
  + **Критерии оценки**: Как будет оцениваться успешность решения задачи.

**1.3. Выбор критериев оценки гипотезы**

* **Критерии оценки**: Метрики, такие как точность, полнота, F1-score, ROC-AUC и другие, в зависимости от задачи.

**1.4. Exploratory Data Analysis (EDA)**

* **Цель**: Понять структуру данных, выявить аномалии, проверить гипотезы.
* **Дисбаланс в датасетах**: Методы работы с дисбалансом включают oversampling, undersampling, использование взвешенных потерь.

**1.5. Как получать надежные оценки**

* **Разделение датасета**: Train/val/test разделение для оценки модели.
* **Стратификация**: Обеспечение равномерного распределения классов в каждом наборе данных.
* **Data leakage**: Избегание утечки данных между train и test наборами.

**1.6. Оверфиттинг, андерфиттинг**

* **Оверфиттинг**: Модель слишком хорошо подстраивается под тренировочные данные, но плохо обобщает на новых данных.
* **Андерфиттинг**: Модель недостаточно хорошо обучается на тренировочных данных.
* **Способы выявления и исправления**: Использование кросс-валидации, регуляризации, увеличения объема данных.

**1.7. Подход к моделированию**

* **Разные архитектуры/алгоритмы**: Важно пробовать разные подходы, так как разные модели могут показывать разные результаты на одних и тех же данных.

**1.8. Бейзлайн**

* **Бейзлайн**: Простая модель, с которой сравниваются более сложные модели.
* **Зачем нужен**: Для оценки улучшений, достигнутых более сложными моделями.

**1.9. Трекинг экспериментов**

* **Воспроизводимость**: Ведение записей о параметрах, метриках и результатах экспериментов для воспроизводимости.

**1.10. Структура репозитория МЛ-проекта**

* **Структура**: Четкая организация кода, данных, моделей и документации для удобства работы и воспроизводимости.

**1.11. Документирование МЛ-проекта**

* **Документация**: Описание целей проекта, методов, результатов и выводов для понимания и воспроизводимости.

**2. Техническая часть**

**2.1. Классический CV (Computer Vision)**

* **Классические методы**: Использование алгоритмов обработки изображений, таких как фильтры, гистограммы, детекторы краев и контуров.
* **Примеры**: SIFT, SURF, HOG, Haar Cascades.

**2.2. Нейросетевой фреймворк - PyTorch / TensorFlow**

* **PyTorch**: Популярный фреймворк для глубокого обучения, известный своей гибкостью и удобством для исследований.
* **TensorFlow**: Широко используемый фреймворк, поддерживаемый Google, с обширной экосистемой инструментов и библиотек.

**2.3. Библиотеки предобученных моделей**

* **Torchvision**: Библиотека для PyTorch, содержащая предобученные модели для задач компьютерного зрения.
* **TensorFlow Hub**: Репозиторий предобученных моделей для TensorFlow.
* **Hugging Face**: Платформа, предоставляющая предобученные модели для различных задач, включая компьютерное зрение.

**2.4. Docker**

* **Docker**: Платформа для контейнеризации приложений, позволяющая создавать, развертывать и управлять контейнерами.
* **Применение**: Обеспечение воспроизводимости и портативности моделей и приложений.

**2.5. ClearML**

* **ClearML**: Платформа для управления экспериментами, автоматизации и мониторинга в проектах машинного обучения.
* **Функции**: Трекинг экспериментов, управление данными, развертывание моделей.

**3. Популярные современные архитектуры**

**3.1. Детекция**

* **YOLO (You Only Look Once)**: Быстрая и точная архитектура для детекции объектов.
* **SSD (Single Shot MultiBox Detector)**: Архитектура для детекции объектов в реальном времени.
* **Faster R-CNN**: Модель для детекции объектов с высокой точностью.

**3.2. Семантическая сегментация**

* **U-Net**: Архитектура для сегментации медицинских изображений.
* **DeepLab**: Модель для семантической сегментации с использованием глубоких сверточных сетей.
* **FCN (Fully Convolutional Networks)**: Архитектура для семантической сегментации изображений.

**3.3. Инстанс-сегментация**

* **Mask R-CNN**: Расширение Faster R-CNN для инстанс-сегментации.
* **PANet (Path Aggregation Network)**: Модель для улучшения инстанс-сегментации.

**3.4. Другие задачи**

* **GANs (Generative Adversarial Networks)**: Генерация новых изображений.
* **VAE (Variational Autoencoders)**: Генеративные модели для создания новых данных.

**4. Алгоритмическое мышление**

* **LeetCode**: Платформа для решения задач по программированию.
* **Уровень medium**: Задачи средней сложности, которые помогают развивать навыки алгоритмического мышления и решения проблем.