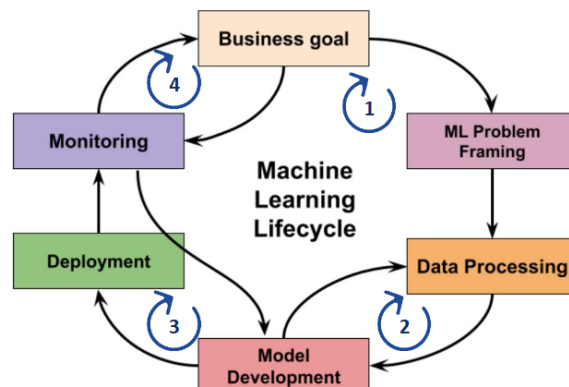


Жизненный цикл проектов в ML

Жизненный цикл в проектах в ML

Циклы:

1. Создания модели для решения бизнес-задачи
2. Итеративной разработки моделей
3. Актуализации модели во время эксплуатации
4. Бизнес-применения модели



Основные этапы

1. Создание модели для решения бизнес-задачи
2. Итеративная разработка моделей
3. Актуализация модели во время эксплуатации
4. Бизнес-применение модели

Этапы жизненного цикла Machine Learning Lifecycle

1. Определение бизнес-цели

- **Задача:** понимание бизнес-проблемы, которую нужно решить с помощью ML
- **Пример:** предсказание оттока клиентов для увеличения удержания пользователей
- **Формула:**
 - **Возврат инвестиций (ROI):**
"ROI равен (выгода от ML модели минус затраты на разработку) делённое на затраты на разработку"
- **Результат:** четко сформулированная бизнес-цель

2. Формулировка ML-проблемы (ML Problem Framing)

- **Задача:** определить тип ML-задачи: классификация, регрессия, кластеризация и т.д.
- **Пример:** задача бинарной классификации — предсказание "уйдет/останется" клиент
- **Формула:**
 - **Предсказание класса:**
"Если вероятность положительного класса при условии данных больше или равна пороговому значению, то предсказанный класс равен 1; иначе — 0"
- **Результат:** определена ML-задача с соответствующими метриками

3. Обработка данных (Data Processing)

- **Задача:** сбор, очистка и подготовка данных для модели
 - **Шаги:**
 - **Удаление пропусков:**
"Пропущенные значения заменяются средним, медианой или наиболее частым значением"
 - **Нормализация данных:**
"Нормализованное значение признака равно разности между значением признака и средним значением признака, делённой на стандартное отклонение признака"
 - **Feature Engineering:** создание новых признаков
 - **Разделение данных:**
"Данные разделяются на обучающую выборку (70%), тестовую выборку (20%) и валидационную выборку (10%)"
 - **Результат:** подготовленный набор данных для моделирования
-

4. Разработка модели (Model Development)

- **Задача:** обучение модели и оптимизация гиперпараметров
 - **Метрики качества:**
 - **Точность (Accuracy):**
"Точность равна количеству верных предсказаний, делённому на общее количество примеров"
 - **Precision и Recall:**
 - **Precision:**
"Precision равен количеству истинно положительных результатов, делённому на сумму истинно положительных и ложно положительных результатов"
 - **Recall:**
"Recall равен количеству истинно положительных результатов, делённому на сумму истинно положительных и ложно отрицательных результатов"
 - **F1-Score:**
"F1-Score равен удвоенному произведению Precision и Recall, делённому на их сумму"
 - **Результат:** обученная и оценённая модель
-

5. Деплой модели (Deployment)

- **Задача:** внедрение модели в продакшн-среду
 - **Реализация:**
 - создание API для запросов и предсказаний
 - настройка CI/CD pipeline для автоматизации обновлений модели
 - **Результат:** модель доступна для использования в бизнес-процессах
-

6. Мониторинг (Monitoring)

- **Задача:** отслеживание производительности модели
- **Метрики:**
 - **Drift данных:**
"Data Drift измеряется как расхождение между распределениями признаков в обучающем и реальном данных"

- **Ошибка модели** (например, RMSE):

"Корень из средней квадратичной ошибки (RMSE) равен квадратному корню из суммы квадратов разностей между реальными и предсказанными значениями, делённой на количество наблюдений"

- **Цель:** обновление модели при снижении качества
- **Результат:** поддержание актуальности и точности модели

Итог

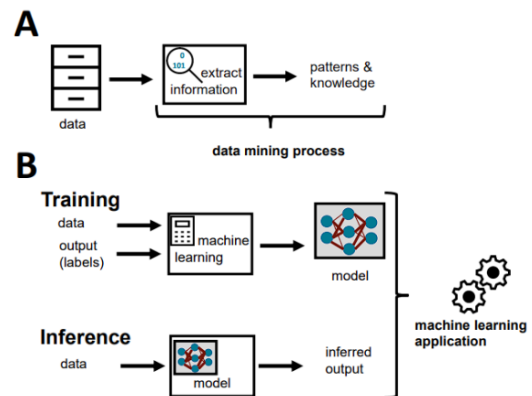
Жизненный цикл ML-модели — это итеративный процесс, включающий постоянную разработку, тестирование, мониторинг и улучшение. Основные компоненты: бизнес-цель, обработка данных, разработка модели, деплой и мониторинг

Модель машинного обучения: процесс обучения и инференса

Модель машинного обучения: процесс обучения и инференса

Модель машинного обучения – математическая модель, способная решать целевую задачу предсказания после настройки параметров. В разных контекстах под моделью понимают или алгоритм решения или набор настроенных параметров (веса модели).

- **Обучение модели** – процесс настройки параметров (весов) модели на основе обучающих данных. Для сложных моделей может требоваться много вычислительных ресурсов и специального оборудования и выполняется до применения модели.
- **Инференс модели** – применение обученной модели к данным, для которых она должна дать предсказание. По сути это производительное использование. Выполняется существенно быстрее обучения, но тоже может требовать специального оборудования.



Определение

Модель машинного обучения – это математическая модель, способная решать целевую задачу предсказания после настройки параметров. В различных контекстах модель может означать:

- алгоритм решения задачи
- набор настроенных параметров (веса модели)

Основные процессы

1. Обучение модели

- **Задача:** настройка параметров (весов) модели на основе обучающих данных
- **Особенности:**
 - Обучение требует большого количества **вычислительных ресурсов**
 - Может потребоваться **специализированное оборудование** для работы с большими данными и сложными архитектурами
 - Выполняется **один раз** до применения модели

- **Этапы:**
 1. **Подготовка данных:** очистка, нормализация и создание обучающей выборки
 2. **Выбор алгоритма:** решение задачи с помощью классификации, регрессии или других методов
 3. **Обучение и валидация:** оптимизация параметров с помощью обучающих данных
 4. **Оценка производительности:** тестирование на отложенных данных
 - **Результат:** обученная модель с настроенными параметрами, готовая к применению
-

2. Инференс модели

- **Задача:** применение обученной модели к данным для получения предсказаний
 - **Особенности:**
 - Является основным этапом **производительного использования модели**
 - Выполняется значительно **быстрее обучения**, так как веса модели уже настроены
 - Может требовать **оптимизации** для работы на устройствах с ограниченными ресурсами (например, IoT-устройства)
 - **Этапы:**
 1. Получение входных данных
 2. Применение модели для расчёта выходного значения
 3. Возврат предсказанного результата
 - **Результат:** предсказания на основе новых данных
-

Визуализация процессов

A. Data Mining Process

- Данные → **Извлечение информации** → Паттерны и знания
- **Назначение:** обнаружение структурированных закономерностей в больших объёмах данных

B. Процесс машинного обучения

1. **Training (Обучение)**
 - Входные данные и метки → **Машинное обучение** → Обученная модель
 - **Результат:** модель, способная предсказывать на основе новых данных
 2. **Inference (Инференс)**
 - Входные данные → **Обученная модель** → Предсказание
 - **Результат:** возвращаемые значения (предсказания) на основе входных данных
-

Применение машинного обучения

1. **Реальные задачи:**
 - Предсказание оттока клиентов
 - Классификация изображений
 - Анализ временных рядов для прогнозирования спроса
2. **Производительность модели:**
 - Зависит от **качества обучения** (правильная подготовка данных и выбор алгоритма)
 - Важна **оптимизация модели** для снижения времени на предсказание
3. **Обслуживание и мониторинг:**

- Регулярная переоценка точности модели на новых данных
- Адаптация модели при **изменении входных данных** или бизнес-задачи

Итоги

Обучение и инференс – два ключевых этапа в работе машинного обучения. Обучение настраивает параметры модели, а инференс использует их для предсказаний. Модели требуют обслуживания и мониторинга для поддержания качества предсказаний

Циклы создания ML-моделей

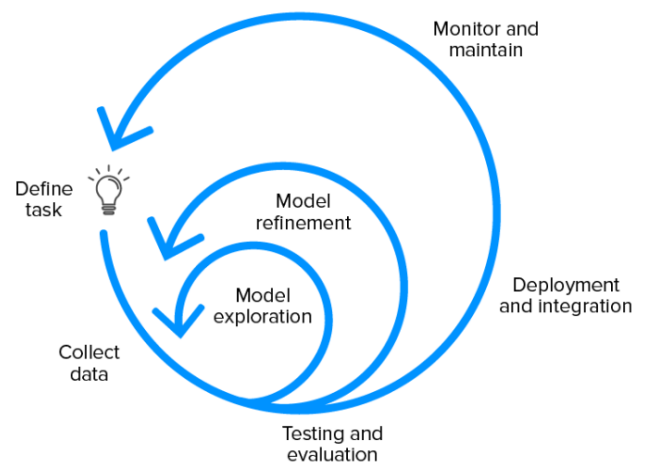
Итеративный подход к созданию моделей

1. Цикл исследования моделей

Циклы создания ML-моделей

Создание моделей – итеративный подход:

- **Цикл исследования моделей** – формирование бейслайнов, поиск SOTA-решений, быстрое прототипирование.
- **Цикл оптимизации модели** – повышение качества наиболее эффективных прототипов.
- **Тестирование и оценка** – выполняется в каждом цикле и выполняется углубленно перед поставкой.
- **Цикл промышленной поставки** – выходное тестирование модели, поставка и интеграции, мониторинг работы.



- **Задача:**
 - Формирование **бейслайнов** (базовых моделей для сравнения)
 - Поиск **SOTA** (State-of-the-Art) решений
 - Быстрое **прототипирование** моделей
- **Особенности:**
 - Исследование данных
 - Создание нескольких прототипов для проверки гипотез
- **Результат:** первоначальный набор моделей для дальнейшей оптимизации

2. Цикл оптимизации модели

- **Задача:** повышение качества наиболее эффективных прототипов
- **Шаги:**
 1. Настройка **гиперпараметров** для улучшения производительности модели
 2. Применение методов регуляризации для предотвращения переобучения
 3. **Feature engineering**: улучшение признаков данных
- **Особенности:**
 - Использование кросс-валидации для точной оценки

- Сравнение моделей по метрикам (Accuracy, F1-score, AUC и др.)
 - **Результат:** оптимизированная модель с высокой точностью и стабильностью
-

3. Тестирование и оценка

- **Задача:** проверка качества модели перед её промышленным использованием
 - **Особенности:**
 - Выполняется **на каждом этапе цикла** разработки
 - Проводится углублённое тестирование перед финальной поставкой
 - **Методы оценки:**
 - **Тестирование на отложенной выборке**
 - Анализ ошибок модели (Error Analysis)
 - Проверка устойчивости к новым данным (Robustness Testing)
 - **Результат:** подтверждение качества и готовности модели к развёртыванию
-

4. Цикл промышленной поставки

- **Задача:**
 - Финальное тестирование модели
 - Поставка, интеграция и развёртывание в производственную среду
 - **Этапы:**
 1. **Развёртывание** модели (Deployment) на сервере или облаке
 2. Интеграция с существующими системами и приложениями
 3. **Мониторинг работы:** отслеживание метрик качества и производительности
 - **Результат:** готовая к использованию модель в реальных бизнес-процессах
-

Визуализация этапов

1. **Define Task:** определение задачи и требований к модели
 2. **Collect Data:** сбор и подготовка данных для обучения
 3. **Model Exploration:** создание и исследование прототипов
 4. **Model Refinement:** оптимизация и настройка лучших моделей
 5. **Testing and Evaluation:** тестирование модели на новых данных
 6. **Deployment and Integration:** интеграция модели в производство
 7. **Monitor and Maintain:** постоянное отслеживание качества и обновление модели при необходимости
-

Итоги

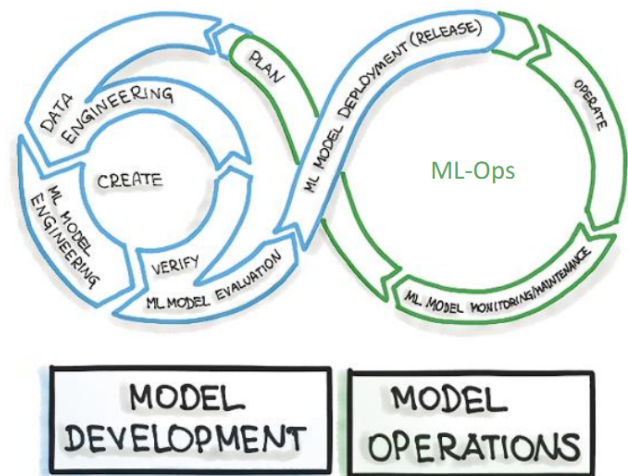
Циклы создания ML-моделей представляют **итеративный процесс**, где каждая итерация приближает модель к наилучшему результату. Ключевые аспекты включают исследование, оптимизацию, тестирование и промышленное внедрение с обязательным мониторингом производительности

ML-dev & ML-ops

ML-dev & ML-ops

ML-ops это практики эффективного развертывания и поддержки ML-моделей в промышленной среде. Основные принципы:

- **Автоматизация** – минимизация ручных операций и стандартизация процессов.
- **Непрерывность** – continuous integration, постоянный мониторинг.
- **Воспроизводимость** – версионирование, документирование, отслеживаемость изменений.
- **Масштабируемость** – горизонтальное масштабирование и эффективное использование ресурсов.
- **Надежность** – обеспечение SLA: отказоустойчивость, восстановление после сбоев.



Определение

MLOps (сокращение от *Machine Learning Operations*) – это набор практик, объединяющих разработку и эксплуатацию моделей машинного обучения для их эффективного развертывания и поддержки в промышленной среде

Основные принципы MLOps

1. Автоматизация

- **Цель:** минимизация ручных операций и стандартизация процессов.
- **Задачи:**
 - Автоматизация конвейеров обучения и развертывания.
 - Использование CI/CD для интеграции и доставки моделей.
- **Результат:** ускоренное внедрение моделей в промышленную среду.

2. Непрерывность

- **Цель:** обеспечение непрерывного мониторинга и обновления моделей.
- **Задачи:**
 - Реализация Continuous Integration (CI) – автоматическая сборка и тестирование моделей.
 - Continuous Monitoring – отслеживание метрик качества и производительности.
- **Результат:** стабильная работа моделей и обновления в реальном времени.

3. Воспроизводимость

- **Цель:** обеспечение повторяемости результатов.
- **Задачи:**
 - Версионирование данных, моделей и кодовой базы.
 - Документирование процессов и изменений.
 - Использование систем контроля версий (например, Git).
- **Результат:** возможность отката к предыдущим версиям и проверка изменений.

4. Масштабируемость

- **Цель:** эффективное использование ресурсов при росте данных и нагрузки.
- **Задачи:**
 - Горизонтальное масштабирование инфраструктуры.

- Оптимизация обучения и инференса для больших объемов данных.
- Использование распределенных систем (например, Kubernetes, Spark).
- **Результат:** стабильная работа моделей при увеличении нагрузки.

5. Надежность

- **Цель:** обеспечение отказоустойчивости и восстановления после сбоев.
 - **Задачи:**
 - Настройка SLA (Service Level Agreement) для гарантии доступности.
 - Создание механизмов резервного копирования и восстановления.
 - Мониторинг сбоев и быстрое устранение проблем.
 - **Результат:** высокая доступность и минимизация простоев.
-

Визуализация процессов

1. **Model Development** (разработка модели):
 - **Data Engineering:** сбор, очистка и подготовка данных.
 - **Create:** построение и обучение модели.
 - **Verify:** тестирование модели.
 - **ML Model Evaluation:** оценка качества модели по метрикам.
 2. **Model Operations (MLOps):**
 - **Plan:** планирование развертывания.
 - **ML Model Deployment (Release):** развертывание обученной модели в промышленной среде.
 - **Operate:** обеспечение работы модели и отслеживание производительности.
 - **ML Model Monitoring/Maintenance:** мониторинг и обновление модели.
-

Инструменты для MLOps

Для реализации практик MLOps используются различные инструменты, которые можно разделить на две категории: проприетарные решения и open-source инструменты.

Проприетарные решения

- **Azure ML:** платформа от Microsoft для разработки и развертывания моделей машинного обучения в облаке.
- **Google Cloud AI Platform:** облачная платформа от Google для создания, обучения и развертывания моделей.

Open-source инструменты

- **MLflow:** платформа для управления жизненным циклом моделей машинного обучения, включая эксперименты, развертывание и регистрацию моделей
- **DVC (Data Version Control):** инструмент для версионирования данных и моделей, интегрируется с Git для обеспечения воспроизводимости
- **Kubernetes:** система оркестрации контейнеров, используемая для развертывания и управления приложениями, включая модели машинного обучения
- **Kubeflow:** платформа для развертывания, мониторинга и управления рабочими процессами машинного обучения на Kubernetes

Выбор между проприетарными и open-source решениями зависит от специфики проекта, требований к безопасности, бюджета и других факторов

Итоги

MLOps объединяет разработку и операционную поддержку моделей машинного обучения, обеспечивая:

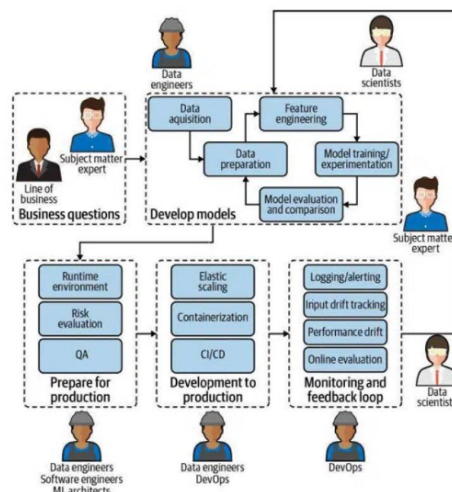
- автоматизацию процессов,
- непрерывный мониторинг и обновление,
- масштабируемость и отказоустойчивость.

Это позволяет быстрее и эффективнее разворачивать ML-модели в производственной среде, сохраняя их качество и стабильность

Команда ML-проекта

Команда ML-проекта

- **Product Owner** – определение требований, приоритизация, оценка эффекта, взаимодействие с заказчиками и стейкхолдерами
 - **Project Manager** – координация работ, планирование, обеспечение ресурсами
- **Domain Expert** – консультирование, интерпретация данных, бизнес-валидация решений
- **Data Engineer** – сбор, подготовка данных для создания моделей
 - **Data Annotator** – разметка, проверка качества данных
- **Data Scientist** – исследование данных, создание моделей и эксперименты
 - **ML Engineer** – продуктивизация моделей, построение конвейеров, оптимизация
 - **MLOps Engineer** – обеспечение ML-инфраструктуры, интеграция в ИТ-ландшафт, автоматизация конвейеров, мониторинг производительности



Ключевые роли

1. Product Owner

- **Задачи:**
 - Определение требований и приоритетов проекта
 - Оценка бизнес-эффекта от ML-решений
 - Взаимодействие с заказчиками и стейкхолдерами
- **Дополнительная роль: Project Manager**
 - **Задачи:** координация работ, планирование задач, управление ресурсами

2. Domain Expert

- **Задачи:**
 - Консультирование команды по специфике данных и задач
 - Интерпретация результатов модели с точки зрения бизнеса
 - Валидация предложенных решений и гипотез

3. Data Engineer

- **Задачи:**
 - Сбор, очистка и подготовка данных для создания ML-моделей
 - Разработка и оптимизация **ETL-процессов** (Extract, Transform, Load)
 - Создание **хранилищ данных** и обеспечение их доступности
 - **Дополнительная роль: Data Annotator**
 - Разметка данных для обучения моделей
 - Проверка и валидация качества размеченных данных
-

4. Data Scientist

- **Задачи:**
 - Исследование данных и создание прототипов моделей
 - Построение и настройка **ML-моделей** (классификация, регрессия, кластеризация)
 - Проведение экспериментов и тестирование моделей
 - Интерпретация результатов и подготовка выводов
-

5. ML Engineer

- **Задачи:**
 - Продуктивизация моделей (развертывание моделей в продакшн)
 - Построение конвейеров **обучения и инференса**
 - Оптимизация моделей для работы с большими данными и реальными нагрузками
-

6. MLOps Engineer

- **Задачи:**
 - Обеспечение **ML-инфраструктуры** и интеграция в IT-ландшафт
 - Автоматизация конвейеров **обучения и развертывания**
 - Мониторинг производительности моделей и инфраструктуры
 - Управление CI/CD-процессами (Continuous Integration и Continuous Delivery)
-

Визуализация процессов (схема на изображении)

1. **Сбор данных и подготовка**
 - **Data Engineers:** сбор данных, feature engineering, подготовка датасетов
 - **Data Annotators:** разметка и валидация данных
2. **Разработка моделей**
 - **Data Scientists:** исследование данных, обучение моделей, их тестирование
 - **Subject Matter Experts:** предоставление контекстной информации и валидация
3. **Развертывание и интеграция**
 - **ML Engineers:** продуктивизация моделей и оптимизация конвейеров
 - **MLOps Engineers:** настройка CI/CD и масштабирование моделей
4. **Мониторинг и поддержка**
 - **DevOps & MLOps Engineers:**
 - Мониторинг производительности моделей

- Управление логами, обнаружение отклонений (**performance drift**)
- Обновление и переобучение моделей по мере необходимости

Итоги

Каждая роль в команде ML-проекта играет важную функцию:

- **Product Owner** управляет требованиями и целями
- **Data Engineer** и **Data Annotator** обеспечивают подготовку данных
- **Data Scientist** разрабатывает и тестирует модели
- **ML Engineer** и **MLOps Engineer** развертывают и поддерживают модели
- **Domain Expert** вносит экспертное понимание задачи

Цели ML-проектов в бизнесе

Цели ML-проектов в бизнесе

Снижение расходов:

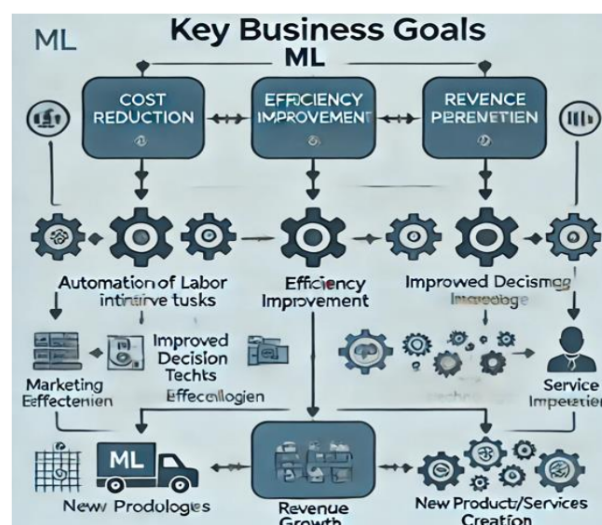
- Автоматизация трудозатратных задач
- Оптимизация процессов
- Борьба с мошенничеством

Повышение эффективности:

- Улучшение принятия решений
- Инновационные технологии

Увеличение выручки:

- Повышение эффективности маркетинга
- Повышение качества и улучшение сервиса
- Создание новых продуктов/услуг



Снижение расходов

1. Автоматизация трудозатратных задач

- Замена рутинных задач машинным обучением (например, обработка документов или ввод данных).
- Примеры:
 - Автоматическое распознавание текстов (OCR) для обработки документов.
 - Предсказание запросов в колл-центрах для автоматического ответа ботами.

2. Оптимизация процессов

- Оптимизация **цепочек поставок**, управления запасами и производства.
- Примеры:
 - Предсказание сроков поставок с использованием временных рядов.
 - Оптимизация маршрутов доставки с использованием алгоритмов машинного обучения.

3. Борьба с мошенничеством

- Обнаружение аномалий и предотвращение финансовых потерь.
- Примеры:
 - Fraud Detection в банковской сфере для выявления подозрительных транзакций.
 - Обнаружение аномалий в данных (например, необычные паттерны поведения клиентов).

Повышение эффективности

1. Улучшение принятия решений

- Использование ML-моделей для анализа данных и поддержки принятия решений.
- Примеры:
 - Предсказание спроса для оптимизации производства и продаж.
 - Рекомендательные системы для предложений товаров и услуг клиентам.

2. Инновационные технологии

- Внедрение технологий на базе машинного обучения для улучшения процессов и создания новых возможностей.
 - Примеры:
 - Компьютерное зрение для контроля качества продукции на заводах.
 - Голосовые помощники и распознавание речи для взаимодействия с пользователями.
-

Увеличение выручки

1. Повышение эффективности маркетинга

- Улучшение таргетинга и персонализация предложений.
- Примеры:
 - Прогнозирование поведения пользователей для оптимизации рекламных кампаний.
 - Анализ данных клиентов для рекомендаций персонализированных акций.

2. Повышение качества и улучшение сервиса

- Использование машинного обучения для улучшения качества обслуживания клиентов.
- Примеры:
 - Чат-боты для быстрой поддержки клиентов.
 - Анализ обратной связи и удовлетворенности клиентов с помощью NLP.

3. Создание новых продуктов/услуг

- Разработка инновационных продуктов на основе машинного обучения.
 - Примеры:
 - Автоматизированные рекомендации (например, Netflix, Spotify).
 - Анализ больших данных для создания новых аналитических инструментов.
-

Визуализация ключевых бизнес-целей (схема)

На основе предоставленного изображения, ключевые цели ML-проектов включают:

- **Cost Reduction** (снижение расходов):
 - Автоматизация рутинных задач
 - Оптимизация рабочих процессов
 - **Efficiency Improvement** (повышение эффективности):
 - Улучшение точности и скорости принятия решений
 - Внедрение инновационных технологических решений
 - **Revenue Growth** (увеличение выручки):
 - Повышение маркетинговой эффективности
 - Создание новых продуктов и услуг
 - Улучшение качества обслуживания клиентов
-

Итоги

Цели ML-проектов в бизнесе включают **снижение расходов, повышение эффективности и увеличение выручки**

Машинное обучение позволяет автоматизировать рутинные задачи, оптимизировать процессы и разрабатывать инновационные решения для бизнеса, что способствует росту и развитию компании

Примеры успешного применения ML в бизнесе:

- **Розничная торговля:** использование ML для анализа покупательского поведения и персонализации предложений, что приводит к увеличению продаж и повышению лояльности клиентов
- **Финансовый сектор:** применение алгоритмов ML для оценки кредитоспособности клиентов, что снижает риски и повышает эффективность кредитования
- **Производство:** внедрение предиктивного обслуживания оборудования с использованием ML позволяет снизить простои и затраты на ремонт
- **Медицина:** использование ML для анализа медицинских изображений и диагностики заболеваний на ранних стадиях повышает качество лечения и снижает расходы на здравоохранение

Постановка задачи ML-проекта

Постановка задачи ML-проекта

Бизнес-цель: Почему это важно для бизнеса?

Постановка задачи: Какую конкретно проблему решаем?

Метрики успеха: Какие ожидаются измеримые бизнес-эффекты и показатели качества модели?

Данные: Какие данные нужны? Где их взять?

Технические требования: Какие ключевые параметры производительности, масштабируемости и точности системы?

Границы: Каков scope, ресурсы и регуляторные аспекты?

Риски: Какие риски и допущения критичны для успеха?



Основные этапы

1. Бизнес-цель

- **Задача:** Определить, почему проект важен для бизнеса и какие бизнес-результаты ожидаются.
- **Пример:**
 - Снижение затрат на обработку данных на 30%.
 - Увеличение точности прогноза спроса для оптимизации логистики.

2. Постановка задачи

- **Задача:** Формулирование конкретной проблемы, которую решает ML-проект.
- **Пример:**
 - Прогнозировать отток клиентов на основе исторических данных.
 - Оптимизировать маршруты доставки с использованием моделей машинного обучения.

3. Метрики успеха

- **Задача:** Определить, какие измеримые бизнес-эффекты и показатели качества будут использоваться для оценки успеха проекта.

- **Примеры метрик:**
 - **Accuracy** – точность предсказаний.
 - **Recall** – полнота для задач обнаружения аномалий.
 - **Business ROI** – возврат инвестиций в проект.
-

Данные

1. **Какие данные нужны:**
 - Источники данных: исторические записи, логи, данные из CRM/ERP.
 - Формат данных: структурированные (таблицы), неструктурированные (тексты, изображения).
 2. **Где их взять:**
 - Внутренние системы компании: базы данных, аналитические платформы.
 - Внешние источники: публичные данные, API, сторонние сервисы.
 - Процесс подготовки:
 - Очистка данных (удаление выбросов, заполнение пропусков).
 - Агрегация и нормализация данных.
-

Технические требования

1. **Ключевые параметры:**
 - **Производительность:** допустимое время на обучение и предсказание.
 - **Масштабируемость:** способность модели работать с увеличивающимся объемом данных.
 - **Точность системы:** минимальный уровень метрик для принятия результатов.
 2. **Инфраструктура:**
 - Платформа для разработки: AWS, GCP, Azure или локальные сервера.
 - Фреймворки: TensorFlow, PyTorch, scikit-learn.
 - Инструменты мониторинга: Prometheus, MLflow, DataDog.
-

Границы проекта

1. **Scope:**
 - Какие задачи входят и не входят в рамки проекта.
 - Ограничения по времени и ресурсам.
 - **Пример:** Модель будет работать с текстовыми данными, но не будет включать изображения.
 2. **Ресурсы:**
 - Финансирование: бюджет на разработку и внедрение.
 - Команда: разработчики, дата-инженеры, MLOps-специалисты.
 3. **Регуляторные аспекты:**
 - Соответствие стандартам GDPR или HIPAA для работы с персональными данными.
 - Обеспечение безопасности и приватности данных.
-

Риски

1. **Какие риски критичны для успеха проекта:**
 - Недостаток данных или их низкое качество.

- Ошибки в постановке задачи и неправильное понимание бизнес-целей.
- Недостаток вычислительных ресурсов для обучения моделей.

2. Риски внедрения:

- Сложности интеграции модели в существующие системы.
- Непонимание работы модели конечными пользователями.

Пример "Правильной постановки задачи"

- **Бизнес-цель:** Повысить удержание клиентов на 10%.
- **Проблема:** Текущий показатель оттока клиентов составляет 25%.
- **Метрики успеха:** Recall $\geq 90\%$ для идентификации клиентов с высоким риском оттока.
- **Данные:** CRM-данные за последние 3 года, включая историю покупок и обращения в поддержку.
- **Технические требования:** Время предсказания ≤ 1 секунда для одного клиента.
- **Границы:** Включить клиентов с активными подписками, исключить пользователей без исторических данных.
- **Риски:** Некорректная разметка данных, низкий уровень адаптации к реальному времени.

Итоги

Постановка задачи ML-проекта включает определение **бизнес-целей**, **конкретной проблемы**, **метрик успеха**, **данных** и **технических требований**. Четкое формулирование задачи помогает избежать недопонимания, оптимизировать ресурсы и достичь измеримого успеха.

"Без внятного ТЗ – результат ХЗ" — правильная постановка задачи – основа успешного ML-проекта

Риски ML-проектов

Риски ML-проектов

Технические риски

- Недостижение целевых метрик
- Проблемы масштабирования
- Появление готовых решений

Бизнес-риски

- Отсутствие реальной ценности для бизнеса
- Сложность внедрения результата в бизнес-процессы
- Неопределенное время получения ценного результата
- Несовместимость ML-проекта с культурой организации

Ресурсные риски

- Недоступность ключевых специалистов
- Недостаток качественных данных
- Отсутствие необходимых мощностей



Технические риски

1. Недостижение целевых метрик

- Причина: Низкое качество данных или неподходящие алгоритмы.
- Решение:
 - Улучшение качества данных (удаление выбросов, заполнение пропусков).
 - Выбор и настройка более подходящих моделей и гиперпараметров.

- A/B тестирование альтернативных подходов.

2. Проблемы масштабирования

- Причина: Нагрузка на инфраструктуру при работе с большими объемами данных.
- Решение:
 - Использование распределенных систем (например, Apache Spark или Kubernetes).
 - Оптимизация моделей для ускорения предсказаний.

3. Появление готовых решений

- Причина: Конкуренты могут использовать уже существующие ML-модели или сервисы.
 - Решение:
 - Разработка уникальных алгоритмов и стратегий.
 - Адаптация и доработка готовых решений под конкретные задачи бизнеса.
-

Бизнес-риски

1. Отсутствие реальной ценности для бизнеса

- Причина: Неправильно определенные бизнес-цели или задачи.
- Решение:
 - Тесное взаимодействие с бизнес-экспертами.
 - Формулирование измеримых и реалистичных целей проекта.

2. Сложность внедрения результатов в бизнес-процессы

- Причина: Неподготовленность инфраструктуры или сопротивление изменениям.
- Решение:
 - Постепенная интеграция модели и создание пилотных проектов.
 - Обучение сотрудников работе с результатами модели.

3. Неопределенное время получения результата

- Причина: Долгий процесс обучения и тестирования моделей.
- Решение:
 - Установление промежуточных целей и регулярных отчетов.
 - Использование Agile-подхода и итеративной разработки.

4. Несовместимость с культурой организации

- Причина: Отсутствие доверия к решениям на основе машинного обучения.
 - Решение:
 - Подготовка объяснимых моделей (*Explainable AI*).
 - Демонстрация прозрачности решений и их бизнес-ценности.
-

Ресурсные риски

1. Недоступность ключевых специалистов

- Причина: Нехватка квалифицированных ML-инженеров или Data Scientists.
- Решение:
 - Привлечение сторонних экспертов или аутсорсинг.
 - Обучение и развитие внутренних специалистов.

2. Недостаток качественных данных

- Причина: Отсутствие достаточного количества или плохое качество данных.
- Решение:
 - Создание процессов **data collection** и **data annotation**.
 - Использование синтетических данных для обучения.

3. Отсутствие необходимых мощностей

- Причина: Недостаток вычислительных ресурсов для обучения больших моделей.
- Решение:
 - Использование облачных решений (AWS, GCP, Azure).
 - Оптимизация моделей и выбор менее ресурсоемких алгоритмов.

Визуализация: Кривая проекта (Research Project Curve)

На диаграмме отображаются ключевые этапы ML-проекта:

1. **Initial Enthusiasm** – начальный энтузиазм и ожидания.
2. **Trough of Sorrow** – период сложностей, неудач и возможного прекращения проекта из-за низкой производительности.
3. **Success** – успешное завершение проекта с достижением бизнес-целей.

Итоги

Риски ML-проектов можно разделить на **технические**, **бизнес** и **ресурсные**.

- Для минимизации рисков важно заранее планировать **технические требования**, ставить четкие **бизнес-цели** и обеспечивать качественную **инфраструктуру** и данные.
- Применение гибких методологий разработки (Agile, CRISP-DM) помогает эффективно управлять рисками и достигать успешных результатов

Управление ML-проектами как R&D

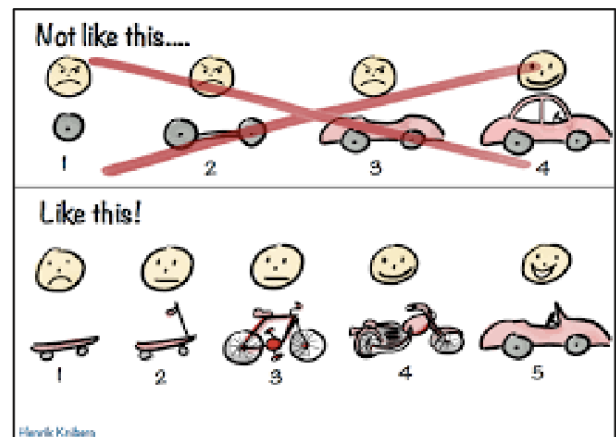
Управления ML-проектами как R&D

Специфика:

- Неопределенность результата
- Зависимость от данных
- Междисциплинарность
- Итеративность и эксперименты
- Сложность внедрения и эксплуатации

Рекомендации:

- Гибкое планирование с регулярным пересмотром
- Выделение промежуточных вех и MVP
- Акцент на исследовательской части в начале
- Параллельная работа над несколькими подходами
- Тесное взаимодействие с бизнес-заказчиком



Специфика ML-проектов

1. **Неопределенность результата**
 - Природа ML-проектов предполагает эксперименты и итеративный поиск наилучшего решения.
 - Точная эффективность модели часто неизвестна до завершения обучения и тестирования.
2. **Зависимость от данных**
 - Качество данных напрямую влияет на успешность модели.
 - Проблемы: неполные, шумные или несбалансированные данные.
3. **Междисциплинарность**

- Успех ML-проектов требует взаимодействия специалистов из разных областей:
 - Data Scientists
 - ML Engineers
 - Domain Experts
 - MLOps Engineers

4. Итеративность и эксперименты

- ML-разработка требует постоянного тестирования гипотез, оптимизации моделей и гиперпараметров.
- Важно быть гибким и готовым к изменениям в ходе проекта.

5. Сложность внедрения и эксплуатации

- Модель должна быть интегрирована в бизнес-процессы и IT-инфраструктуру.
- Эксплуатация требует постоянного мониторинга и обновления модели.

Рекомендации по управлению ML-проектами

1. Гибкое планирование с регулярным пересмотром

- Используйте **Agile-методологии** для итеративного планирования и быстрой адаптации к изменениям.
- Регулярные встречи для обновления статуса и корректировки целей.

2. Выделение промежуточных вех и MVP

- Разработка **минимально жизнеспособного продукта (MVP)** на ранних этапах для быстрой демонстрации ценности.
- Постановка промежуточных целей для измерения прогресса.

3. Акцент на исследовательской части в начале

- Начните с **исследования данных** и определения применимости ML для задачи.
- Проведение **Proof of Concept (PoC)**: демонстрация работоспособности подхода на небольшом наборе данных.

4. Параллельная работа над несколькими подходами

- Тестируйте различные модели и алгоритмы одновременно для поиска оптимального решения.
- Примеры: сравнение моделей на основе метрик (Random Forest vs XGBoost).

5. Тесное взаимодействие с бизнес-заказчиком

- Регулярная коммуникация для уточнения требований и ожиданий.
- Обратная связь на каждом этапе для корректировки вектора проекта.

Визуализация: Итеративный подход к созданию решений

"Not like this" (неправильный подход):

- Последовательное добавление частей системы без создания промежуточной ценности.
- Результат: недовольные стейкхолдеры и риски остановки проекта.

"Like this" (правильный подход):

- Постепенное добавление функциональности с регулярной доставкой ценности.

- Результат: удовлетворенность на каждом этапе и постоянный прогресс к целевому решению.

Пример процесса управления ML-проектом

- 1. **Исследовательская фаза:**
 - Анализ данных, постановка гипотез, PoC.
- 2. **Разработка и эксперименты:**
 - Обучение и тестирование моделей, оптимизация гиперпараметров.
- 3. **Интеграция и внедрение:**
 - Встраивание модели в бизнес-процессы, создание API, CI/CD конвейеры.
- 4. **Мониторинг и поддержка:**
 - Отслеживание производительности, устранение дрейфа модели и переобучение.

Итоги

Управление ML-проектами требует **гибкого подхода**, учета **неопределенности результатов** и постоянного взаимодействия с бизнес-заказчиком. Использование **итеративного процесса** с акцентом на быструю доставку ценности и регулярное тестирование гипотез позволяет минимизировать риски и повысить успешность проектов

Приемка ML-проекта

Приемка ML-проекта

Метрики качества



- Определение бизнес-метрик для решения
- Выбор технических метрик качества моделей и установка целевых значений

Методы оценки

- Расчет метрик модели на тестовой выборке
- Сравнение метрик с бейслайном
- A/B тестирование прототипов
- Пилотное внедрение

Гибкость в оценке и эффективная коммуникация

- Готовность корректировать критерии успеха
- Понятная отчетность для всех стейкхолдеров
- Регулярное информирование о прогрессе

Accuracy	Predictions/ Classifications	$\frac{\text{Correct}}{\text{Correct} + \text{Incorrect}}$
Precision	Predictions/ Classifications	$\frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$
Recall	Predictions/ Classifications	$\frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$
F1	Predictions/ Classifications	$\frac{2 * \text{True Positive}}{\text{True Positive} + 0.5 (\text{False Positive} + \text{False Negative})}$
IoU	Object Detections/ Segmentations	$\frac{\text{Pixel Overlap}}{\text{Pixel Union}}$ <div></div>

Метрики качества

- 1. **Определение бизнес-метрик для решения**
 - Устанавливаются бизнес-цели и измеримые показатели, отражающие ценность модели.
 - Пример: увеличение точности прогноза спроса на 15%.
- 2. **Выбор технических метрик качества модели**
 - Метрики зависят от типа задачи (классификация, регрессия, сегментация).
 - Примеры:
 - **Accuracy** (точность) – доля правильных предсказаний:
Accuracy = количество правильных предсказаний / общее количество предсказаний
 - **Precision** (точность) – доля верно предсказанных положительных классов:
Precision = количество истинно положительных / (количество истинно положительных + количество

ложно положительных)

- **Recall** (полнота) – доля найденных положительных примеров:
$$\text{Recall} = \frac{\text{количество истинно положительных}}{\text{количество истинно положительных} + \text{количество ложно отрицательных}}$$
 - **F1-score** – гармоническое среднее между Precision и Recall:
$$\text{F1-score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$
 - **IoU** (Intersection over Union) – используется в задачах детекции объектов:
$$\text{IoU} = \frac{\text{площадь пересечения}}{\text{площадь объединения}}$$
-

Методы оценки качества

1. Расчет метрик модели на тестовой выборке

- Используется отложенная выборка данных для проверки работы модели на ранее невидимых данных.

2. Сравнение метрик с бейзлайном

- Бейзлайн – это простая модель или эвристика, которая задает минимальный порог качества.
- Пример:
 - Для классификации: предсказание самого частого класса.
 - Для регрессии: предсказание среднего значения целевой переменной.

3. A/B тестирование прототипов

- Сравнение производительности двух или более моделей на реальных данных в условиях продакшена.

4. Пилотное внедрение

- Ограниченный запуск модели в бизнес-процессах для проверки на небольшом объеме данных.
 - Пример: модель тестируется на одной категории товаров перед масштабированием на весь ассортимент.
-

Гибкость в оценке и эффективная коммуникация

1. Готовность корректировать критерии успеха

- Метрики могут меняться в процессе разработки и тестирования модели.
- Пример: замена Accuracy на Precision при необходимости минимизировать ложные срабатывания.

2. Понятная отчетность для всех стейкхолдеров

- Регулярные отчеты о прогрессе и результатах тестирования модели.
- Визуализация метрик: графики, диаграммы, матрица ошибок.

3. Регулярное информирование о прогрессе

- Организация встреч для обсуждения текущих результатов и препятствий.
 - Использование Agile-подхода с короткими итерациями для быстрой обратной связи.
-

Визуализация метрик качества

На основе предоставленного изображения, важнейшие метрики для оценки ML-моделей включают:

1. **Accuracy**: общее соотношение правильных предсказаний.
 2. **Precision**: точность положительных предсказаний.
 3. **Recall**: полнота положительных предсказаний.
 4. **F1-score**: баланс между Precision и Recall.
 5. **IoU**: пересечение областей в задачах детекции и сегментации.
-

Итоги

Приемка ML-проекта включает определение бизнес-метрик, расчет технических метрик на тестовых данных и пилотное внедрение. Эффективная коммуникация и гибкий подход к оценке помогают своевременно адаптировать критерии успеха и довести проект до успешного результата.

Ключевые рекомендации:

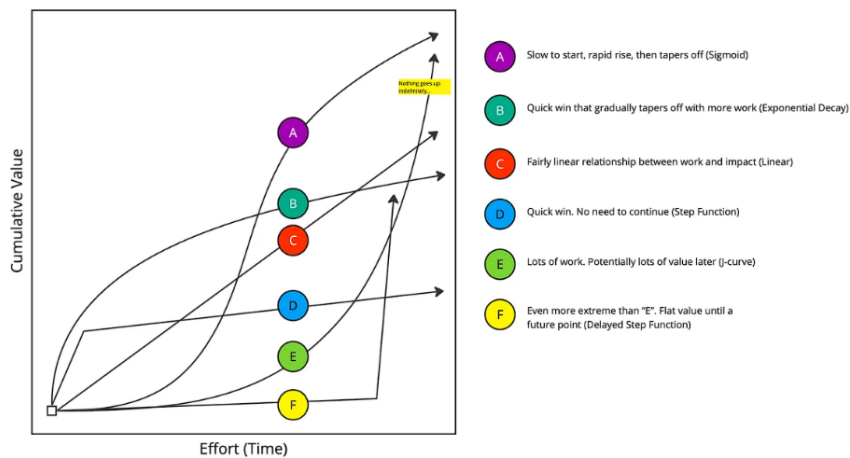
- Сравнивайте метрики с бейзлайном.
- Тестируйте модель на реальных данных (A/B тестирование).
- Постепенно внедряйте модель в бизнес-процессы через пилотные запуски.

Экономика ML-проекта

Основные концепции

Экономика ML-проекта

Кривая затраты-эффект (качество метрики)



Экономика ML-проекта включает в себя анализ вложенных ресурсов (времени и усилий) и их влияния на получаемую ценность (качество метрик). **Кривая затраты-эффект** демонстрирует, как различные стратегии разработки модели влияют на накопленную ценность с течением времени.

Кривая затраты-эффект (Качество метрики)

Обозначения на графике

1. A: Sigmoid (Сигмовидная кривая)

- **Характеристика:** медленный старт, быстрый рост в середине и постепенное замедление улучшений.
- **Пример:** сложные модели, которые требуют времени на начальную настройку, но затем показывают значительный рост качества.

2. B: Exponential Decay (Экспоненциальное затухание)

- **Характеристика:** быстрый выигрыш в начале, но дальнейшие улучшения требуют непропорционально больших усилий.
- **Пример:** простые модели или оптимизация с быстрым результатом на начальных этапах.

3. C: Linear (Линейная зависимость)

- **Характеристика:** качество модели растет равномерно с увеличением усилий.
- **Пример:** постепенная итеративная оптимизация модели без резких скачков.

4. D: Step Function (Ступенчатая функция)

- **Характеристика:** быстрый выигрыш на ранних этапах, после чего дальнейшие усилия не приносят значимого эффекта.
- **Пример:** базовая настройка модели, после которой требуется полная смена подхода для новых улучшений.

5. **E: U-curve (U-образная кривая)**

- **Характеристика:** значительный объем работы в начале, но накопленная ценность растёт значительно позже.
- **Пример:** проекты, требующие глубокого анализа данных или долгосрочных инвестиций в инфраструктуру.

6. **F: Delayed Step Function (Отложенная ступенчатая функция)**

- **Характеристика:** долгий период отсутствия прогресса, после которого наблюдается резкий рост.
- **Пример:** долгосрочные исследования и инновационные решения, требующие длительного времени на разработку и тестирование.

Анализ кривых и их применение

1. Как выбрать подход?

- **A** подходит для долгосрочных проектов с потенциалом значительного улучшения метрик.
- **B** используется для быстрых прототипов и MVP, когда требуется быстрый результат.
- **C** применим для проектов с регулярным улучшением через итерации.
- **D** целесообразен для проектов с минимальными вложениями и немедленным эффектом.
- **E** и **F** подходят для крупных инновационных проектов с отсроченным результатом.

2. Оценка стоимости и эффекта

- Сравнение трудозатрат (времени и усилий) с достигнутой ценностью позволяет оценить экономическую целесообразность выбранной стратегии.
- Наилучшие стратегии — это **B** и **D**, если необходимо быстрое внедрение.
- Для долгосрочных и масштабных решений приоритет отдается **A** и **E**.

Практические рекомендации

1. Оцените начальные и конечные затраты

- Определите, какой подход оптимален для ваших ресурсов и целей: быстрый результат или долгосрочная стратегия.

2. Мониторинг кривой прогресса

- Используйте метрики и графики для регулярного отслеживания ROI (возврат на инвестиции).

3. Распределяйте ресурсы с умом

- Для быстрых результатов начните с простых моделей (**B** и **D**).
- Для сложных и критически важных проектов инвестируйте в глубинное обучение и долгосрочные исследования (**A**, **E**, **F**).

4. Экспериментируйте и адаптируйте стратегию

- Применение Agile-подходов позволяет гибко переключаться между разными стратегиями в зависимости от получаемых результатов.

Итоги

Кривая затраты-эффект демонстрирует, как усилия по разработке ML-моделей влияют на получаемую ценность. Правильный выбор стратегии позволяет оптимизировать экономическую эффективность проекта и достичь бизнес-целей с минимальными затратами.

Ключевые выводы:

- Стратегии **В** и **Д** — лучшие для быстрых решений.
- Стратегии **А**, **Е** и **Ф** подходят для долгосрочных и инновационных проектов.
- Регулярно анализируйте кривую прогресса для адаптации стратегии и оптимизации ресурсов.