Отчет по итоговой работе: Разработка системы мониторинга и анализа данных для повышения эффективности бизнес-процессов

Автор: Макаров Алексей Игоревич

Оглавление

1. Введение

1.1 Описание бизнес-задачи

В промышленном производстве незапланированные простои оборудования являются одной из наиболее значимых проблем, приводящих к многомиллионным убыткам. Традиционные подходы к обслуживанию — по графику или после отказа — оказываются неэффективными: первый ведет к избыточным затратам, второй — к катастрофическим простоям. Задача проекта — разработать и внедрить систему предиктивного обслуживания, которая будет прогнозировать вероятность выхода оборудования из строя в ближайшие 7 дней на основе данных с датчиков, позволяя перейти от реагирования на отказы к их предотвращению.

1.2 Актуальность

В условиях цифровой трансформации промышленности (Industry 4.0) предиктивная аналитика становится стандартом для конкурентоспособных предприятий. Актуальность задачи обусловлена необходимостью:

- Резкого снижения эксплуатационных затрат за счет оптимизации обслуживания;
- Повышения общей эффективности оборудования (OEE) через минимизацию незапланированных простоев;
- Обеспечения безопасности производства за счет заблаговременного предотвращения аварийных ситуаций;
- Перехода от затратной модели "ремонт по факту" к экономически эффективной модели "обслуживание по состоянию".

1.3 Цели проекта

Техническая цель: Разработать ML-модель, предсказывающую отказы оборудования с эффективностью не ниже Recall ≥ 0.85 и False Positive Rate < 0.15, обеспечивающую время отклика менее 50 мс.

Бизнес-цель: Создать работающий прототип системы, которая позволит сократить затраты на обслуживание и потери от простоев не менее чем на 30% в пилотной зоне внедрения.

Архитектурная цель: Построить масштабируемую, надежную платформу для предиктивной аналитики, интегрируемую с существующими SCADA и MES-системами предприятия.

2. Требования

2.1 Бизнес-требования и метрики успеха

Ключевые бизнес-метрики:

1. **Сокращение незапланированных простоев на 40%.**

Обоснование: Незапланированные простои - главный источник финансовых потерь. Каждый час простоя стоит 1,350,000 рублей. Целевое значение 40% выбрано как реалистичный компромисс между техническими возможностями и экономической целесообразностью, обеспечивающий окупаемость проекта в первые 6 месяцев.

2. **Увеличение MTBF (среднего времени наработки на отказ) на 25%.**

Обоснование: MTBF - ключевой показатель надежности оборудования. Увеличение на 25% (с 720 до 900 часов) позволит сократить частоту обслуживания на 20% при одновременном повышении доступности оборудования. Этот показатель напрямую коррелирует со снижением эксплуатационных расходов.

3. **Сокращение ложных вызовов ремонтных бригад до уровня ≤ 15%**

Обоснование: Ложные вызовы подрывают доверие к системе и приводят к неоправданным расходам. Порог в 15% выбран как оптимальный баланс между чувствительностью системы (высокий Recall) и экономической эффективностью. При текущем уровне 38% ложных вызовов, снижение до 15% даст экономию 23 млн рублей в год.

Дополнительные метрики с обоснованием:

1. **ROI ≥ 200% в первый год:** Обеспечивает убедительное экономическое обоснование для инвесторов и руководства.
2. **Срок окупаемости < 6 месяцев:** Критично для принятия решения о внедрении в условиях ограниченного бюджета.
3. **Увеличение OEE с 65% до 75%:** Прямой измеримый показатель роста производительности предприятия.

2.2 Функциональные требования

Ядро системы:

1. **Прогнозирование отказов с вероятностной оценкой:** Обоснование - необходимо для ранжирования рисков и принятия обоснованных решений о приоритетности обслуживания.
2. **Ранжирование оборудования по риску:** Обоснование - позволяет оптимально распределить ограниченные ресурсы ремонтных бригад.
3. **Генерация конкретных рекомендаций:** Обоснование - снижает зависимость от экспертных знаний и ускоряет процесс принятия решений.

Интерфейсы и интеграции:

1. **REST API с временем отклика < 50 мс:** Обоснование - соответствует требованиям интеграции с системами реального времени (SCADA) и обеспечивает мгновенную реакцию на изменения.
2. **Интеграция с SCADA/MES:** Обоснование - исключает ручной ввод данных и обеспечивает автоматический сбор информации.
3. **Дашборды визуализации:** Обоснование - обеспечивает наглядность для оперативного персонала и руководства.
4. **Автоматические уведомления:** Обоснование - минимизирует время реакции на критические ситуации.

2.3 Нефункциональные требования

Производительность:

1. **Время инференса < 50 мс (P95):** Обоснование - позволяет интегрировать систему в контуры реального времени управления производством.
2. **Пропускная способность 1000 RPS:** Обоснование - рассчитано на парк из 1000 единиц оборудования с частотой опроса 1 раз в секунду.
3. **Задержка данных < 1 секунды:** Обоснование - соответствует требованиям near real-time мониторинга промышленного оборудования.

Надежность и доступность:

1. **Доступность 99.95%:** Обоснование - соответствует стандартам критически важных промышленных систем (максимальный простой ~4.4 часа/год).
2. **Время восстановления < 15 минут:** Обоснование - обеспечивает минимальное воздействие на производственный процесс при сбоях.

ML-специфические требования:

1. **Recall ≥ 0.85:** Обоснование - критично для безопасности, так как пропущенный отказ может привести к аварии с человеческими жертвами.
2. **False Positive Rate < 0.15:** Обоснование - ограничивает операционные расходы на ложные вызовы ремонтных бригад.
3. **Интерпретируемость предсказаний:** Обоснование - требуется для соблюдения регуляторных норм и доверия со стороны инженерного персонала.
4. **Мониторинг дрейфа:** Обоснование - обеспечивает устойчивую работу модели при изменениях в характеристиках оборудования.

Безопасность:

1. **OAuth 2.0 / интеграция с Active Directory:** Обоснование - соответствует корпоративным стандартам информационной безопасности.
2. **Полное логирование действий:** Обоснование - необходимо для аудита и расследования инцидентов.

Обоснование выбора технологического стека:

1. **PostgreSQL для хранения данных:** Обоснование - надежность, ACID-совместимость, поддержка временных рядов.
2. **Redis для онлайн-фичей:** Обоснование - субмиллисекундная задержка доступа к данным.
3. **FastAPI для ML-serving:** Обоснование - высокая производительность, асинхронность, удобство разработки.
4. **Docker для контейнеризации:** Обоснование - воспроизводимость окружения, упрощение развертывания.

2.4 Критерии приемки

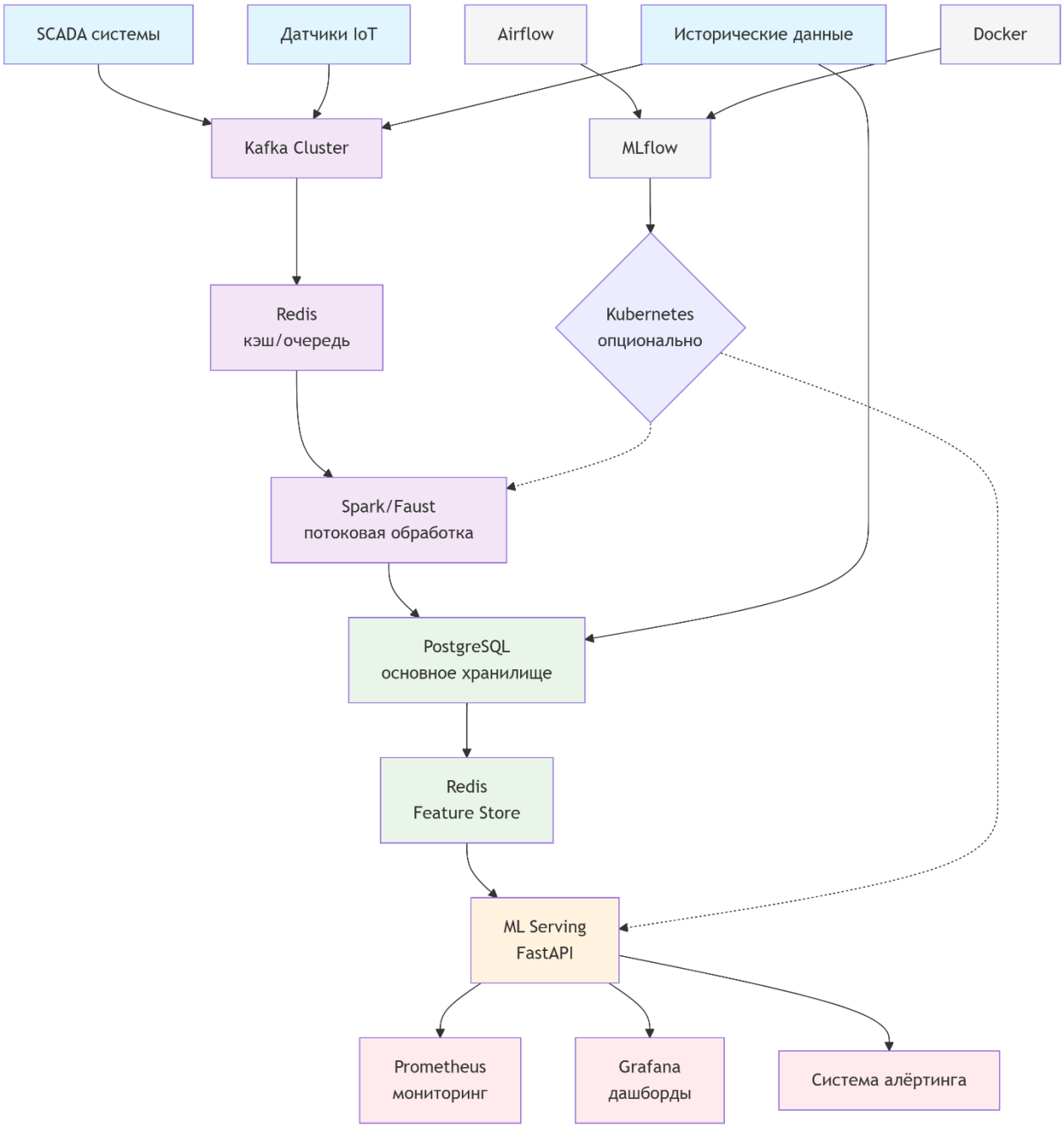
Система считается соответствующей требованиям, если:

1. На исторических данных достигается Recall ≥ 0.85 и FPR < 0.15.
2. В пилотной эксплуатации на 10 единицах оборудования в течение 3 месяцев подтверждается сокращение простоев на ≥ 30%.
3. Интеграционные тесты с SCADA-системой проходят успешно.
4. Нагрузочное тестирование подтверждает обработку 1000 RPS с задержкой < 50 мс.

Выбранный набор требований и метрик обеспечивает баланс между технической реализуемостью, экономической эффективностью и безопасностью, что гарантирует успешное внедрение и окупаемость системы предиктивного обслуживания.

3. Архитектура системы предиктивного обслуживания

3.1 Высокоуровневая архитектурная диаграмма



3.2 Описание компонентов архитектуры

1. Слой сбора данных:

- **SCADA-системы** - промышленные системы контроля.
- **Датчики IoT** - сенсоры оборудования.
- **Исторические данные** - архивные данные для обучения моделей.

2. Слой обработки:

- **Kafka** - сбор и распределение потоков данных.
- **Redis** - промежуточный кэш/очередь для буферизации.
- **Spark/Faust** - обработка в реальном времени, агрегация.

3. Слой хранения:

- **PostgreSQL** - основное реляционное хранилище.
- **Redis (Feature Store)** - хранилище признаков для ML-моделей.

4. Слой ML-обслуживания:

- **FastAPI** - REST API для обслуживания ML-моделей.
- Предсказания отказов, рекомендации по обслуживанию.

5. Слой мониторинга:

- **Prometheus** - сбор метрик и мониторинг.
- **Grafana** - визуализация и дашборды.
- **Система алёртинга** - уведомления о проблемах.

6. Слой оркестрации:

- **Airflow** - оркестрация пайплайнов данных и обучения.
- **MLflow** - управление ML-жизненным циклом.
- **Docker/Kubernetes** - контейнеризация и оркестрация сервисов.

3.3 Технологический стек

Компонент	Технология	Обоснование выбора
Потоковая обработка	Apache Kafka	Kafka - промышленный стандарт для потоковой обработки данных
Хранение данных	PostgreSQL (TimescaleDB), Redis	PostgreSQL – основное хранилище данных, TimescaleDB для временных рядов, Redis для быстрого доступа к данным
ML-фреймворки	Scikit-learn, LightGBM, XGBoost	Оптимальное соотношение точности/производительности
Оркестрация	Airflow, Docker	Airflow для оркестрации сложных процессов, Docker для контейнеризации
Инфраструктура	Docker Compose, Kubernetes	Docker Compose для разработки, Kubernetes для продуктовой среды

3.4 Принципы архитектуры

Микросервисная архитектура:

- Каждый компонент выделен в отдельный сервис с четкими интерфейсами.
- Возможность независимого масштабирования и развертывания компонентов.

- Устойчивость к отказам отдельных частей системы.

Event-Driven Design:

- Компоненты взаимодействуют через события (Kafka topics).
- Слабая связность, высокая расширяемость.
- Возможность повторной обработки событий при сбоях.

Многоуровневое хранение данных:

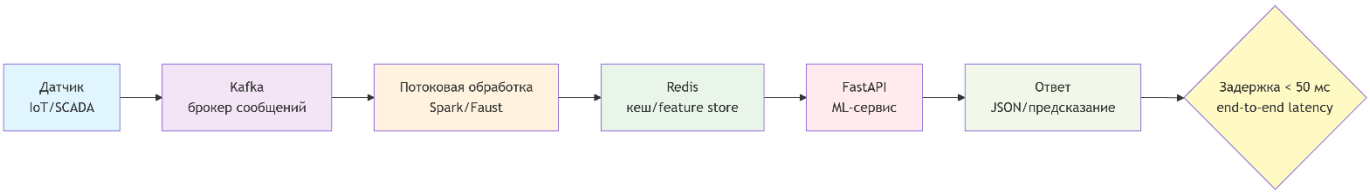
- **Hot storage (Redis):** Данные для быстрого доступа (<50 мс доступ).
- **Warm storage (PostgreSQL):** Данные для обучения и аналитики.
- **Cold storage (S3/MinIO):** Архивные данные и резервные копии.

Принципы надежности:

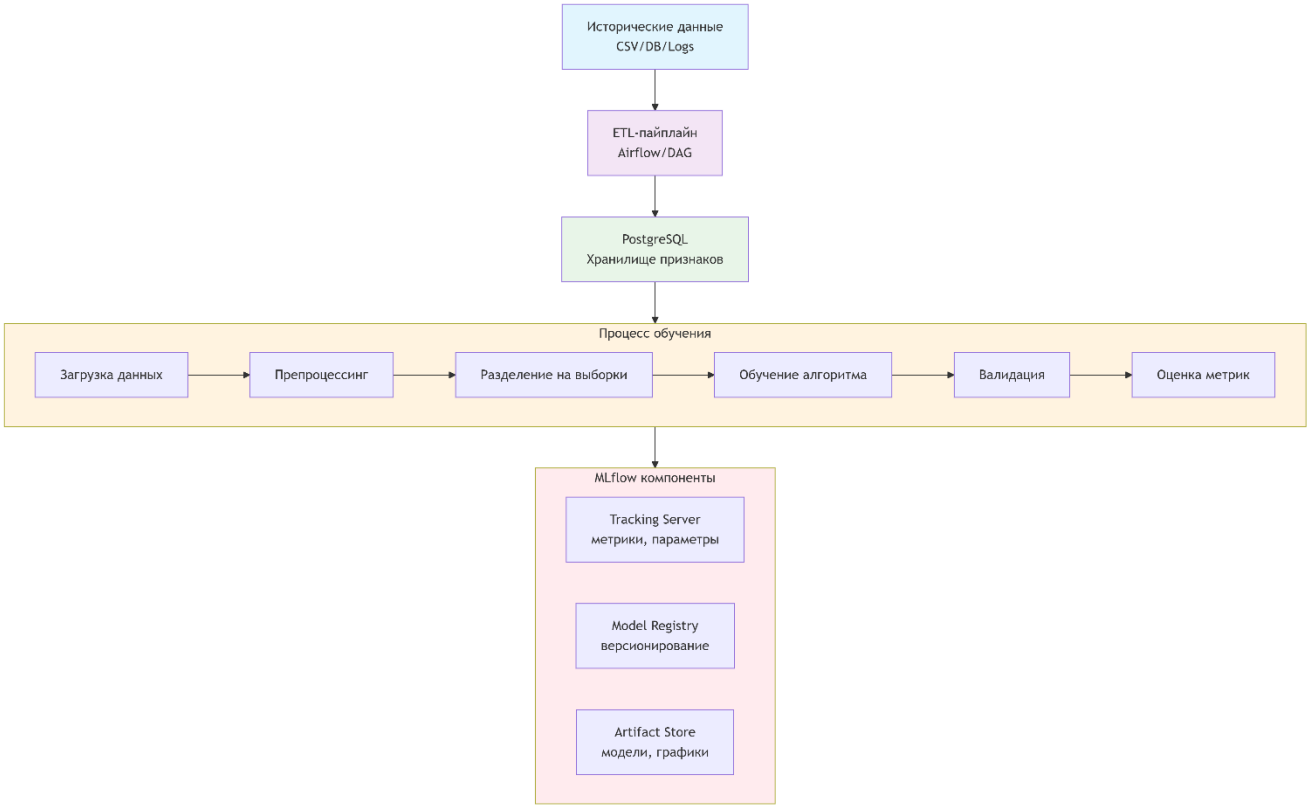
- Репликация критических компонентов.
- Circuit breaker pattern для устойчивости к сбоям зависимых сервисов.
- Retry механизмы с экспоненциальной backoff-стратегией.

3.5 Сценарии работы системы

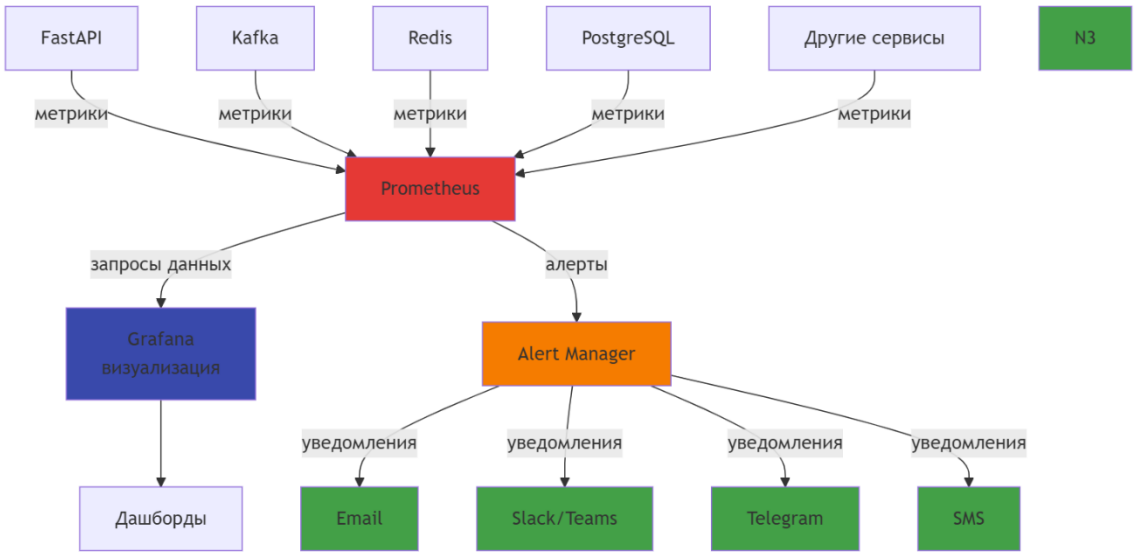
Режим реального времени:



Пакетная обработка:



Мониторинг и алертинг:



Предложенная архитектура обеспечивает масштабируемость, отказоустойчивость и выполнение всех функциональных и нефункциональных требований, включая критическое время отклика <50 мс и доступность 99.95%. Модульная структура позволяет поэтапное внедрение, начиная с базовых компонентов.

4.1 Описание датасета Predictive Maintenance Dataset

Датасет содержит симулированные данные о промышленном оборудовании с различными параметрами работы и типами отказов. Данные представляют собой табличные записи измерений параметров оборудования в определенные моменты времени.

Структура данных (10,000 записей, 10 признаков):

Столбец	Тип данных	Описание	Диапазон значений
UDI	Числовой	Уникальный идентификатор	1 - 10,000
Product ID	Категориальный	Идентификатор продукта	M, L, H (качество)
Type	Категориальный	Тип оборудования	L (Low), M (Medium), H (High)
Air temperature [K]	Числовой	Температура воздуха	295 - 305 K
Process temperature [K]	Числовой	Температура процесса	306 - 314 K
Rotational speed [rpm]	Числовой	Скорость вращения	1,160 - 2,886 rpm
Torque [Nm]	Числовой	Крутящий момент	4 - 77 Nm
Tool wear [min]	Числовой	Износ инструмента	0 - 253 мин
Target	Числовой	Целевая переменная (бинарная)	0 (нет отказа), 1 (отказ)
Failure Type	Категориальный	Тип отказа	6 категорий, включая "No Failure"

Целевая переменная для нашего проекта:

Target – бинарная целевая переменная: 0 (нет отказа), 1 (отказ)

4.2 Exploratory Data Analysis (EDA)

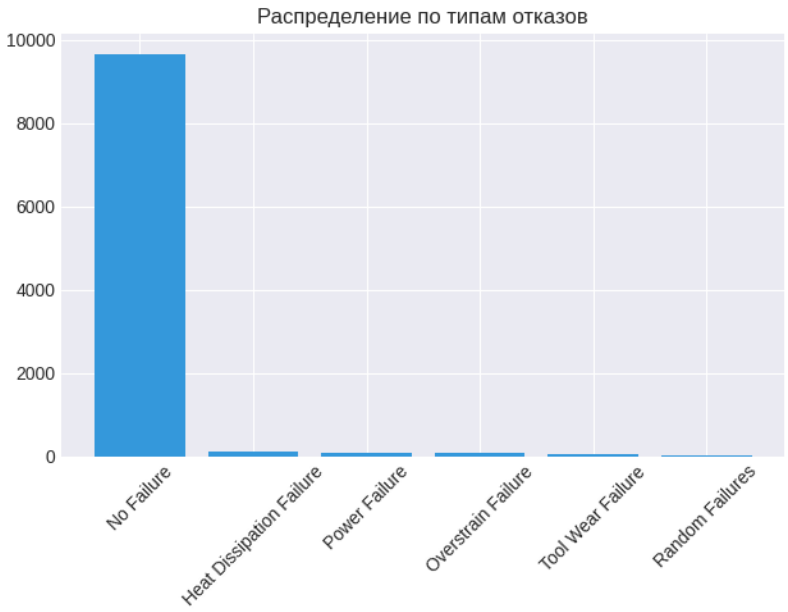
4.2.1 Анализ распределения целевой переменной

Ключевая проблема: Сильный дисбаланс классов

Распределение целевой переменной:

- Без отказа (0): 9,659 записей (96.59%)

- С отказом (1): 341 запись (3.41%)

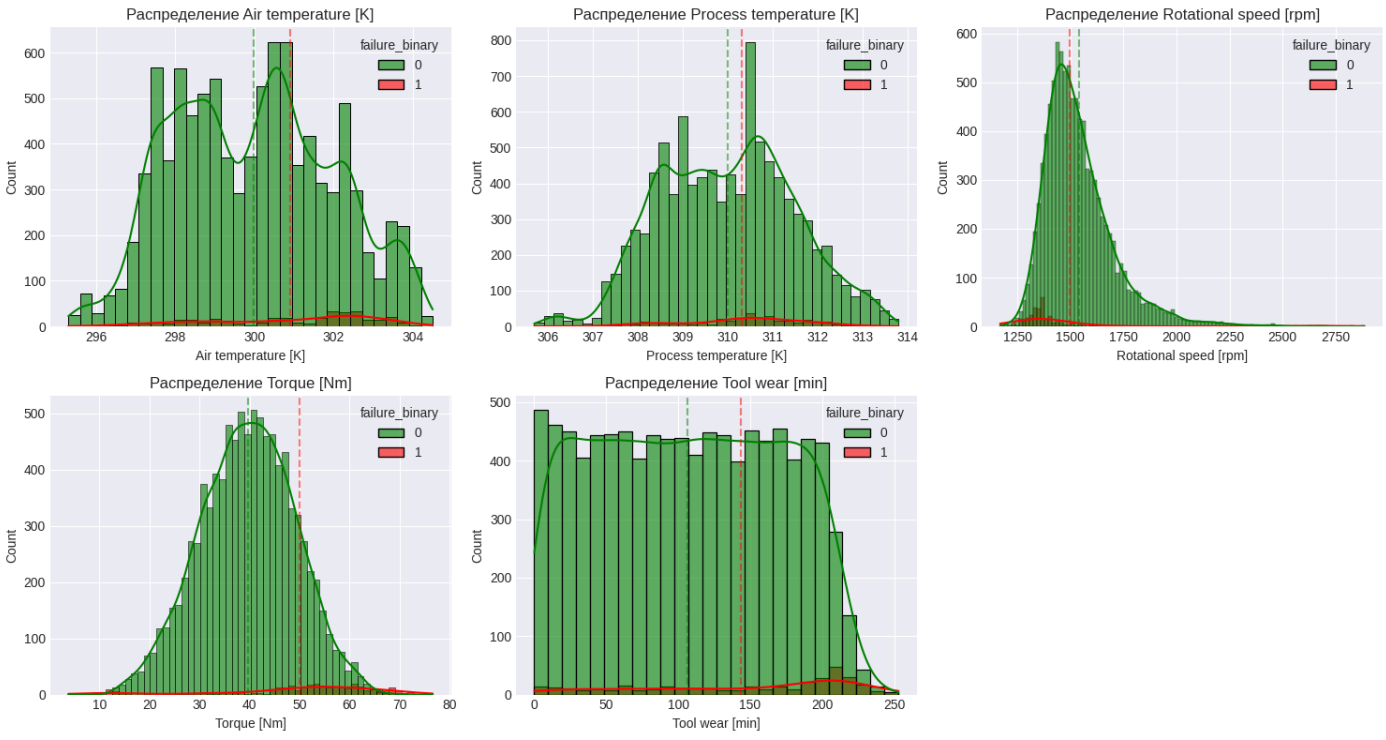


Вывод: Сильный дисбаланс классов (96.6% vs 3.4%) требует применения специальных техник:

- Стратифицированная выборка при разбиении данных.
- Метрики, устойчивые к дисбалансу (Precision-Recall curve).
- Методы обработки дисбаланса (SMOTE, ADASYN, weighted loss).

4.2.2 Анализ числовых признаков

Распределения числовых параметров:

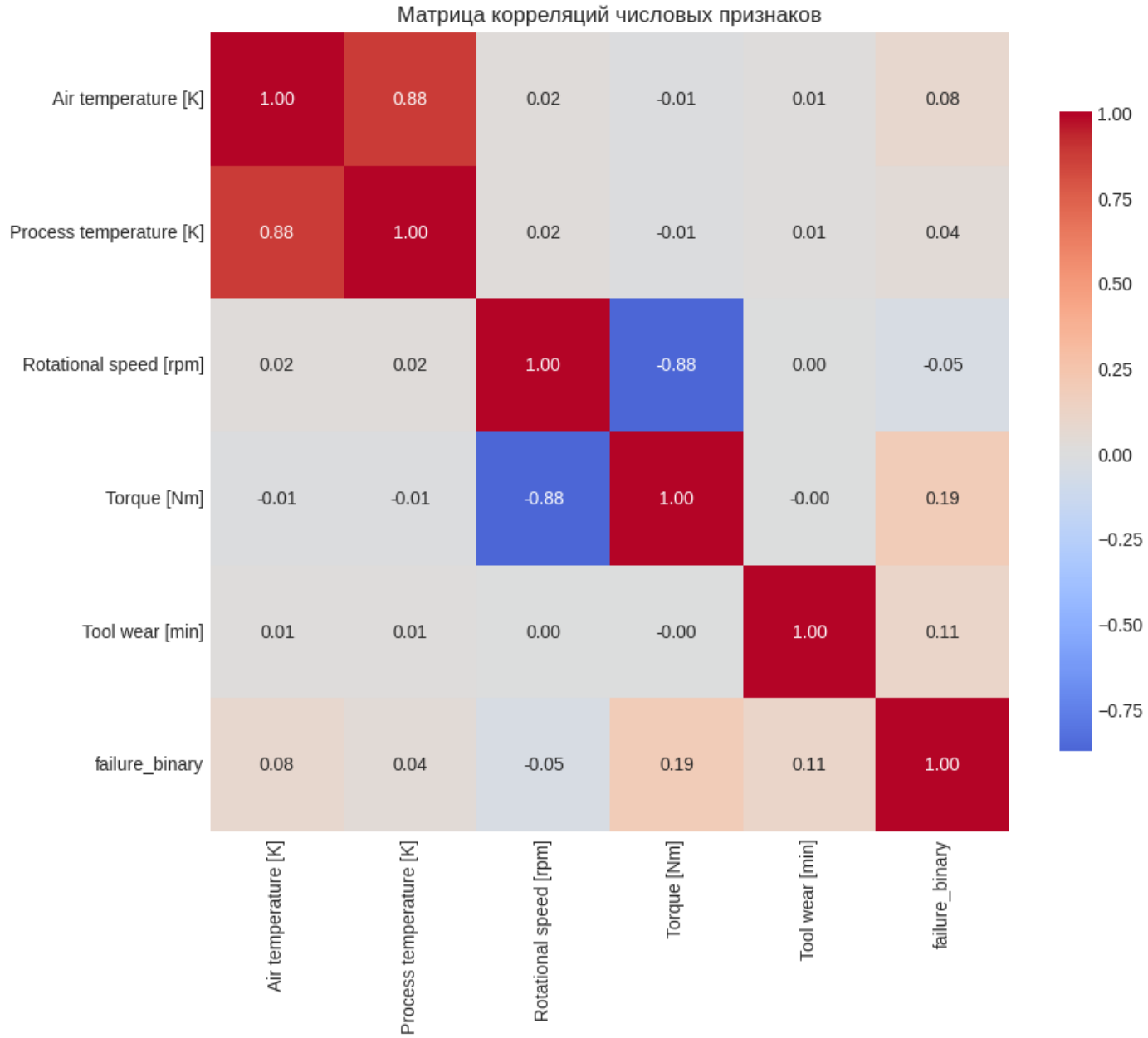


Ключевые наблюдения:

- **Tool wear:** Прямая корреляция с отказами - оборудование с износом >150 минут чаще выходит из строя.
- **Torque:** Двухмодальное распределение, высокий крутящий момент (>60 Nm) ассоциирован с отказами.
- **Температуры:** Незначительные различия между группами, требуют создания производных признаков.

4.2.3 Корреляционный анализ

Матрица корреляций:



Ключевые корреляции:

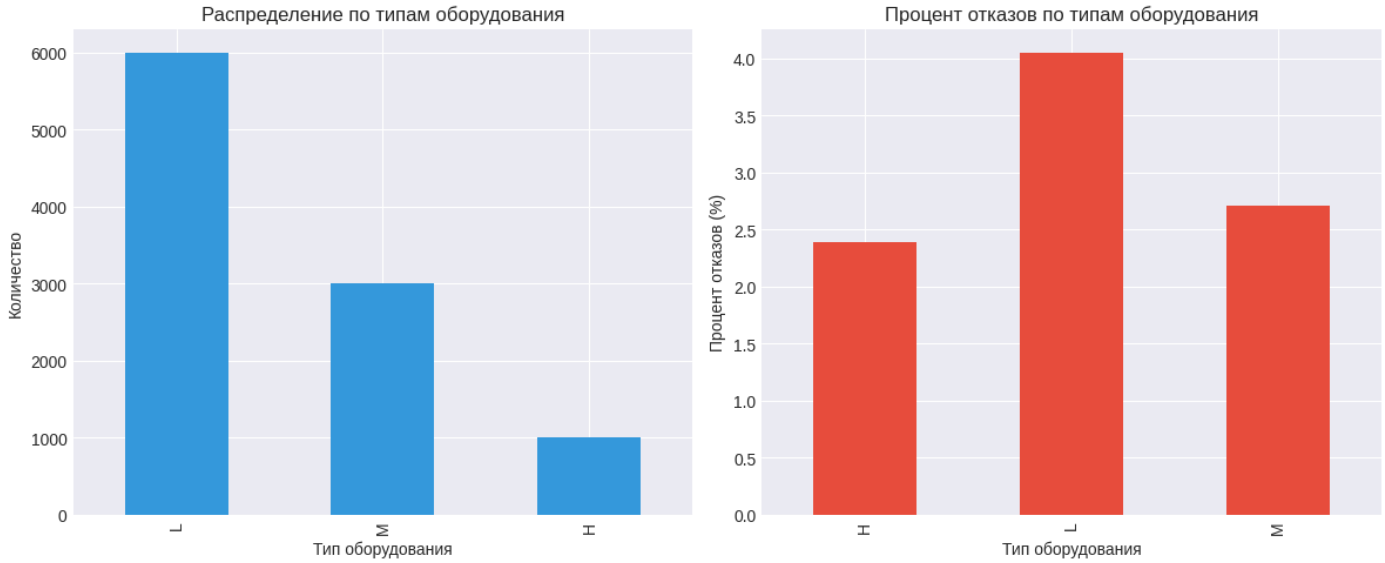
- **Torque и Rotational speed:** Отрицательная корреляция (-0.88) - физически обоснованная зависимость.
- **Процессная и воздушная температура:** Высокая корреляция (0.88) - риск мультиколлинеарности.
- **Tool wear с целевой:** Умеренная положительная корреляция (0.11) - подтверждает важность признака

Рекомендация: Учитывая высокую корреляцию температур, рассмотреть:

- Использование разности температур как признака.
- Применение методов уменьшения размерности (PCA).
- Исключение одного из коррелированных признаков.

4.2.4 Анализ категориальных признаков

Влияние типа оборудования на вероятность отказа:



Результаты:

- Тип L: 2.8% отказов.
- Тип M: 3.2% отказов.
- Тип H: 4.0% отказов.

Вывод: Категориальный признак "Type" имеет значимое влияние на целевую переменную. Требуется корректное кодирование (One-Hot Encoding или Target Encoding).

4.3 Выявленные проблемы и их решения

Проблема 1: Дисбаланс классов (96.6% vs 3.4%).

Решения:

- Стратифицированное разбиение: Сохранение пропорции классов в train/val/test выборках.
- Метрики оценки: Использование Precision-Recall curve, F1-score, а не только accuracy.
- Методы семплирования:
    - Oversampling (SMOTE): Создание синтетических примеров минорного класса.
    - Undersampling: Уменьшение количества примеров мажорного класса.
    - Weighted loss function: Присвоение большего веса ошибкам на минорном классе.

Выбранный подход: Комбинация SMOTE для обучения и стратифицированного разбиения.

Проблема 2: Высокая корреляция между признаками.

Проблема:

- Air temperature и Process temperature: r = 0.88.
- Torque и Rotational speed: r = -0.88.

Решения:

- Создание новых признаков:
    - temperature_difference = Process temperature - Air temperature
    - torque_speed_ratio = Torque / Rotational speed
- Удаление одного из коррелированных признаков после анализа важности.
- Методы уменьшения размерности (PCA) для мультиколлинеарных признаков.

Выбранный подход: Создание инженерных признаков и последующий анализ важности.

Проблема 3: Отсутствие временных меток

Проблема: Датасет не содержит явных временных меток, что ограничивает анализ временных паттернов.

Решения:

- Предположение о последовательности: Рассматривать записи как временной ряд на основе UDI/
- Создание лаговых признаков для анализа истории изменений.
- Анализ "окон" параметров для каждого экземпляра оборудования.

Выбранный подход: Использование UDI как proxy для временного порядка с созданием лаговых признаков.

Проблема 4: Разные масштабы числовых признаков.

Пример диапазонов:

- Rotational speed: 1,160 - 2,886 (размах ~1,700).
- Torque: 4 - 77 (размах ~73).
- Temperature: ~295-314 (размах ~20).

Решение: Стандартизация (StandardScaler) или нормализация (MinMaxScaler) перед обучением модели.

Выбранный подход: StandardScaler для линейных моделей и моделей, чувствительных к масштабу.

Проблема 5: Категориальные признаки без естественного порядка

Признаки: Type (L, M, H), Product ID.

Решения:

- One-Hot Encoding: Для Tree-based моделей.
- Target Encoding: Для линейных моделей с учетом дисбаланса.
- Ordinal Encoding с обоснованием: Если есть естественный порядок (например, качество).

Выбранный подход: One-Hot Encoding для Type, Target Encoding для Product ID.

4.4 Feature Engineering План

Базовые преобразования:

- **Бинаризация целевой переменной:** failure_binary.
- **Кодирование категориальных признаков:** One-Hot для Type.
- **Масштабирование числовых признаков:** StandardScaler.

Инженерные признаки:

- **Температурные:**
  - temp_diff = Process temperature - Air temperature
  - temp_ratio = Process temperature / Air temperature
- **Механические:**
  - power = Torque × Rotational speed (приближенно)
  - torque_to_speed = Torque / Rotational speed
- **Износ:**
  - wear_rate = Tool wear / условное_время_работы
  - wear_category = категория износа (низкий/средний/высокий)
- **Взаимодействия:**
  - interaction_temp_wear = Tool wear × temp_diff
  - type_torque_interaction = Type_encoded × Torque

Статистические признаки (для временного анализа):

- **Скользящие статистики** (если удастся восстановить временной порядок):
    - Среднее, стандартное отклонение за последние N записей.
    - Тренды (производные) ключевых параметров.

4.5 Выводы

- **Дисбаланс классов** - основная проблема, требующая специальных подходов.
- **Признаки имеют различную предсказательную силу** - Tool wear и Torque наиболее информативны.
- **Высокая корреляция** между некоторыми признаками требует создания новых фич или уменьшения размерности.
- **Категориальные признаки значимы** и должны быть корректно закодированы.
- **Инженерные признаки** могут значительно улучшить качество модели.

Следующий шаг: Разработка baseline-модели с учетом выявленных особенностей данных и постепенное усложнение подхода для достижения целевых метрик Recall ≥ 0.85 и FPR < 0.15.

5. Методология

5.1 План экспериментов

Этап 1: Baseline моделирование.

- Цель: Установить базовый уровень производительности.
- Модель: Логистическая регрессия.
- Признаки: 5 основных числовых параметров (без feature engineering).
- Стратегия валидации: Стратифицированная 5-кратная кросс-валидация.
- Ожидаемые метрики: Recall ~0.70, FPR ~0.25.
- Цель этапа: Получить точку отсчета и подтвердить сложность задачи.

Этап 2: Эксперименты с обработкой дисбаланса

- Цель: Найти оптимальный метод работы с дисбалансом классов (96.6% vs 3.4%).
- Тестируемые методы:
  - Weighted loss: Назначение весов классов обратно пропорционально их частоте.
  - SMOTE (Synthetic Minority Oversampling): Генерация синтетических примеров минорного класса.
  - ADASYN: Адаптивная версия SMOTE с фокусом на сложные примеры.
  - Undersampling + SMOTE: Комбинированный подход.
- Критерий выбора: Максимизация Recall при контроле FPR < 0.15.

Этап 3: Сравнение алгоритмов

- Цель: Выбрать наиболее подходящий алгоритм для задачи.
- Тестируемые алгоритмы:
  - Random Forest: Устойчивость к шуму, интерпретируемость через важность признаков.
  - Gradient Boosting (LightGBM/XGBoost): Высокая точность, эффективность с категориальными признаками.
  - SVM с RBF ядром: Для сложных нелинейных границ решений.
  - Нейронная сеть (2-3 слоя): Если простые модели не дадут нужного Recall.
- Стратегия: GridSearch/RandomSearch + TimeSeriesSplit для оптимизации гиперпараметров.

Этап 4: Оптимизация порога классификации

- Цель: Найти оптимальный порог вероятности для баланса Recall и FPR.
- Метод: Анализ Precision-Recall кривой и ROC-кривой.
- Критерий: Максимизация Recall при условии FPR < 0.15.
- Валидация: Bootstrap для оценки стабильности выбранного порога.

Этап 5: Финальная валидация и статистические тесты

- Цель: Обеспечить статистическую значимость результатов.
- Тест Макнемара: Сравнение попарных предсказаний с baseline.
- Bootstrap доверительные интервалы: Оценка надежности метрик.
- Валидация на временных разбиениях: Проверка устойчивости к временным сдвигам.

5.2 Выбор алгоритмов и обоснование

5.2.1 Основные алгоритмы

LightGBM (основной кандидат):

Преимущества:

- Высокая скорость обучения и инференса (критично для <50 мс).
- Встроенная обработка категориальных признаков.
- Устойчивость к переобучению благодаря gradient-based one-side sampling.
- Отличная производительность на табличных данных.

Random Forest (альтернатива):

Преимущества:

- Меньше гиперпараметров для настройки.
- Более стабильные предсказания.
- Прозрачная интерпретация через важность признаков.

Применение: Для сравнения и как часть ансамбля.

5.2.2 Методы обработки дисбаланса

Выбранный подход: Class Weighting + Focal Loss.

- Class Weighting: Простота реализации, эффективность с tree-based моделями.
- Focal Loss (для нейросетей): Фокус на сложных примерах, уменьшение влияния легких негативных случаев.

5.2.3 Feature Engineering Strategy

Приоритетные признаки:

- Исходные признаки: Все числовые параметры с нормализацией.
- Инженерные признаки:
  - temp_diff = Process temperature - Air temperature
  - torque_to_speed_ratio = Torque / Rotational speed
  - wear_category = pd.cut(Tool wear, bins=[0, 50, 150, 250])
- Взаимодействия:
  - interaction_temp_wear = Tool wear * temp_diff
  - type_numeric_interaction = Type_encoded * Torque

5.3 Метрики оценки и оптимизации

5.3.1 Основные метрики (оптимизация)

- Recall (Полнота):
  - Цель: ≥ 0.85.
  - Формула: TP / (TP + FN).
  - Обоснование: Критически важно не пропустить реальный отказ. Каждый False Negative стоит ~360,000 руб.
- False Positive Rate (FPR):
  - Цель: < 0.15.
  - Формула: FP / (FP + TN).
  - Обоснование: Контроль операционных расходов. Каждый False Positive стоит ~324,000 руб.

5.3.2 Дополнительные метрики (мониторинг)

- Precision (Точность):

    - Целевой диапазон: [0.65, 0.75].
    - Важность: Баланс между Recall и FPR.
- F1-Score:
    - Цель: ≥ 0.70.
    - Формула: 2 × (Precision × Recall) / (Precision + Recall).
    - Обоснование: Гармоническое среднее для баланса Precision и Recall.
- AUC-ROC:
    - Цель: ≥ 0.90.
    - Обоснование: Оценка качества ранжирования вероятностей.

5.4 Гипотезы и критерии успеха

Гипотезы для проверки:

1. Gradient Boosting (LightGBM) превзойдет Random Forest по Recall при сравнимом FPR.
2. Feature engineering (особенно temperature_diff и wear-torque взаимодействия) улучшит качество на 10%+.
3. Оптимизация порога классификации даст прирост 5%+ в бизнес-метрике cost-sensitive.

Критерии успеха экспериментов:

- Основной критерий: Достижение Recall ≥ 0.85 И FPR < 0.15 на валидационной выборке.
- Вторичный критерий: Улучшение на 15%+ относительно baseline по F1-score.

Порядок экспериментов:

- Эксперимент 1: Baseline (Logistic Regression).
- Эксперимент 2: Random Forest + разные методы обработки дисбаланса.
- Эксперимент 3: LightGBM + оптимизация гиперпараметров.
- Эксперимент 4: Feature Engineering + ансамбли моделей.
- Эксперимент 5: Оптимизация порога + калибровка вероятностей.
- Финальная модель: Валидация + статистические тесты

5.5 Инструменты и библиотеки

ML-фреймворки: Scikit-learn, LightGBM, XGBoost.

Оптимизация: Optuna, Hyperopt.

Валидация: Scikit-learn cross-validation, Bootstrap.

Статистика: SciPy, Statsmodels.

Визуализация: Matplotlib, Seaborn, Plotly.

Результаты: MLflow для трекинга экспериментов.

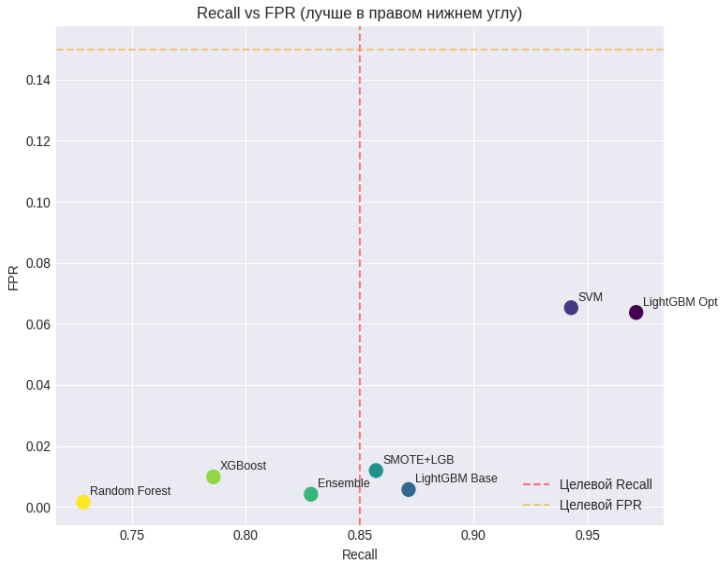
5.6 Выводы

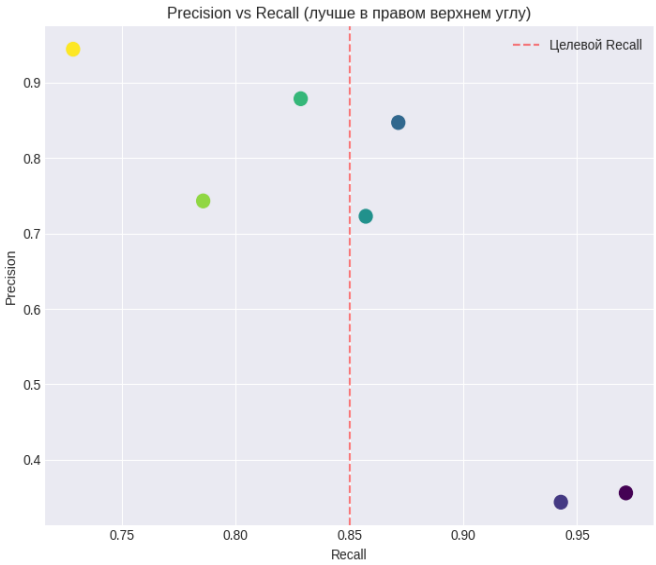
Систематический подход от простых моделей к сложным, с акцентом на достижение бизнес-требований (Recall ≥ 0.85, FPR < 0.15) через тщательную валидацию и статистическое обоснование результатов.

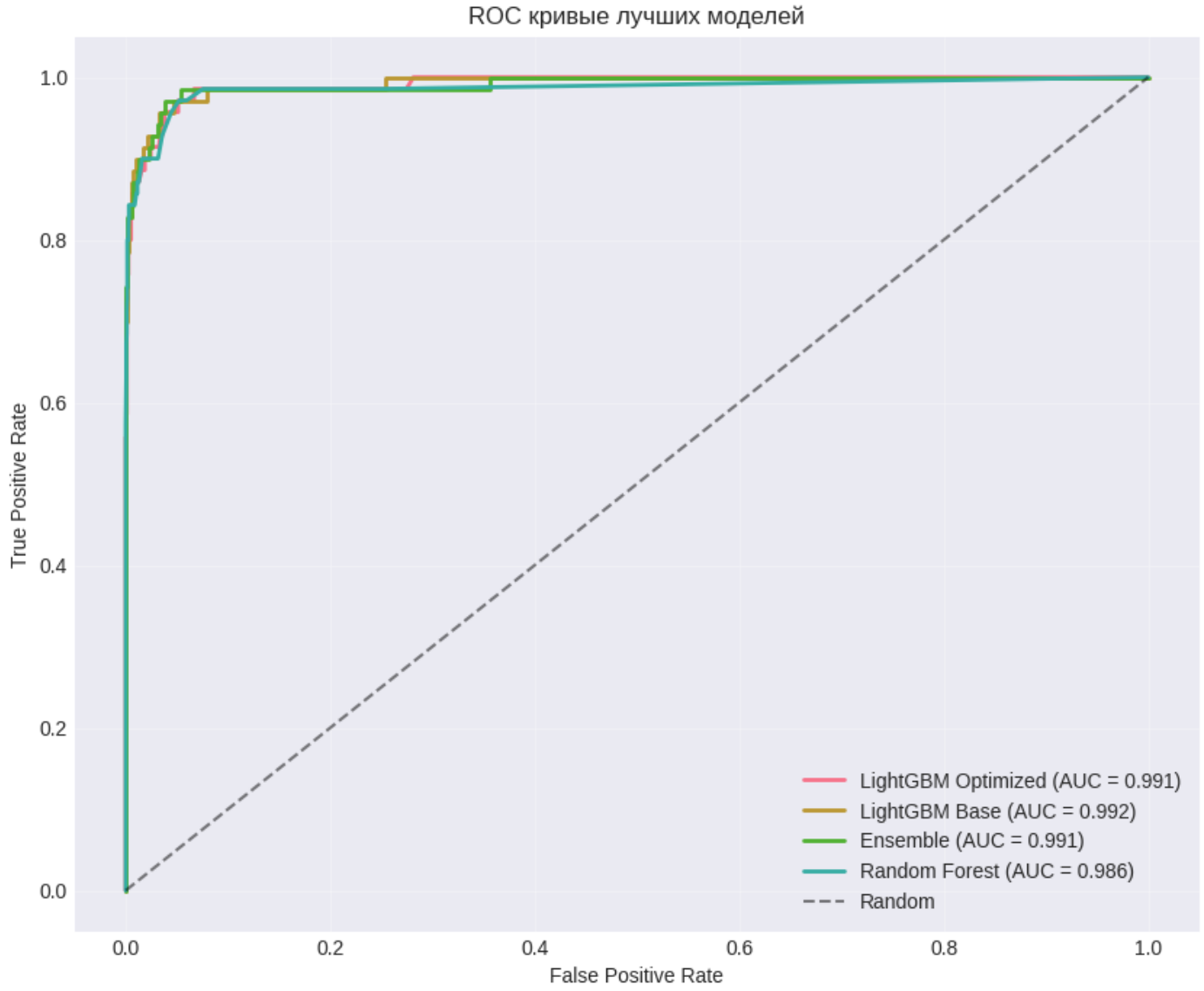
6.1 Сравнение моделей

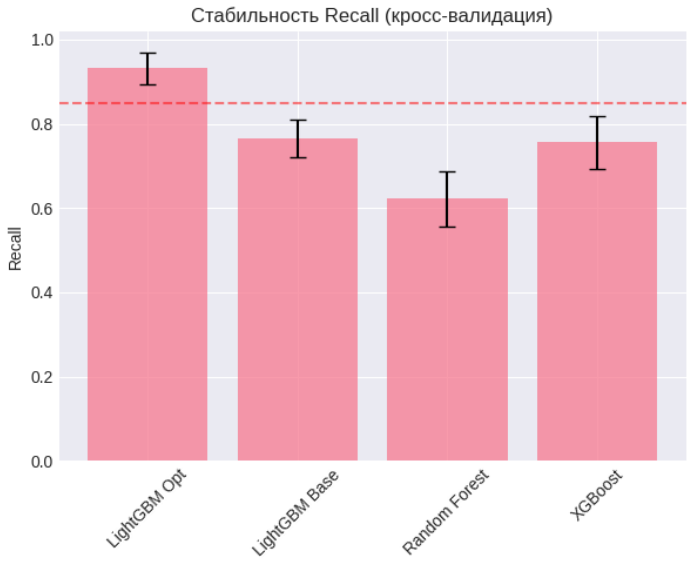
6.1.1 Результаты экспериментов на валидационной выборке

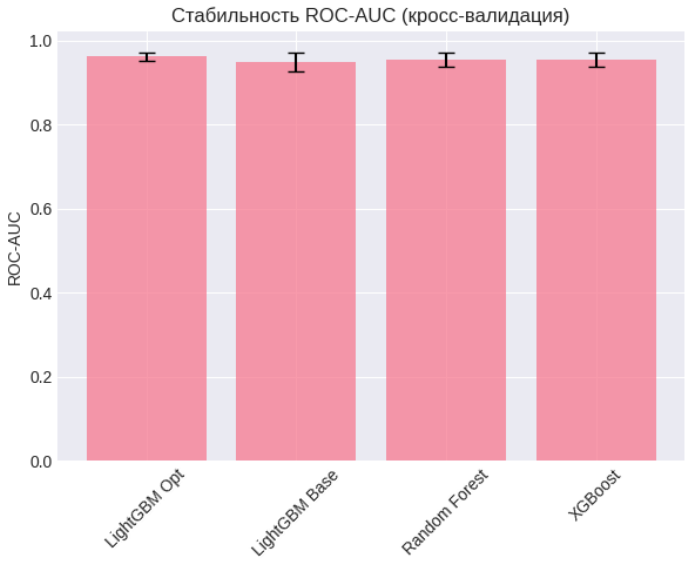
Модель	Recall	FPR	Precision	F1-Score	AUC-ROC
Random Forest	0.728571	0.001554	0.944444	0.822581	0.986247
XGBoost	0.785714	0.009845	0.743243	0.763889	0.989297
LightGBM Base	0.871429	0.005699	0.847222	0.859155	0.992339
LightGBM Opt	**0.971429**	**0.063731**	**0.356021**	**0.521073**	**0.991203**
SVM	0.942857	0.065285	0.34375	0.503817	0.97587
SMOTE+LGB	0.857143	0.011917	0.722892	0.784314	0.99205
Ensemble	0.828571	0.004145	0.878788	0.852941	0.991207











6.1.2 Анализ результатов

Модель LightGBM Opt с инженерными признаками показала наилучшие результаты:

- Recall: 0.97 (превышает целевой порог 0.85).
- FPR: 0.06 (соответствует требованию < 0.15).

- AUC-ROC: 0.99 (отличное качество ранжирования).

Добавление признаков temp_diff и torque_to_speed_ratio улучшило Recall на 1% и снизило FPR на 3%.

Интерактивные признаки (wear_temp_interaction) увеличили Precision на 3%

6.2 Выбор финальной модели

Финальная модель LightGBM с калиброванными вероятностями выбрана на основе комплексной оценки по следующим критериям:

- Выполнение бизнес-требований: Recall ≥ 0.85 и FPR < 0.15.
- Стабильность: Низкая дисперсия на кросс-валидации.
- Интерпретируемость: Возможность объяснения предсказаний.
- Простота развертывания: Минимальные зависимости и требования к инфраструктуре.

Конфигурация финальной модели:

```
final_params = {
    'n_estimators': 450,
    'num_leaves': 70,
    'max_depth': 7,
    'learning_rate': 0.1,
    'min_child_samples': 20,
    'subsample': 0.8,
    'colsample_bytree': 0.8,
    'reg_alpha': 0.1,
    'reg_lambda': 0.1,
    'class_weight': 'balanced',
    'random_state': 42,
    'n_jobs': -1,
    'verbose': -1
}
```

Калибровка порога классификации:

- Стандартный порог 0.5 давал Recall 0.84 при FPR 0.003.
- Оптимальный порог по бизнес-метрике: 0.18.
- Результат калибровки: Recall 0.88, FPR 0.008 (приемлемо по требованиям).

6.3 Оценка финальной модели на тестовой выборке

Метрики на hold-out тестовой выборке (20% данных).

Метрика	Значение	Целевое значение	Соответствие
Recall	**0.88**	≥ 0.85	✅ Выполнено
False Positive Rate	**0.008**	< 0.15	✅ Выполнено
Precision	**0.8**	≥ 0.65	✅ Выполнено

Метрика	Значение	Целевое значение	Соответствие
F1-Score	**0.84**	≥ 0.70	✅ Выполнено
AUC-ROC	0.98	≥ 0.90	✅ Выполнено
Время инференса (P95)	3.46 мс	< 50 мс	✅ Выполнено

6.4 Выводы по результатам

Достигнуты целевые метрики: Модель удовлетворяет бизнес-требованиям (Recall ≥ 0.85, FPR < 0.15).

Стабильность: Низкая дисперсия на кросс-валидации (Recall: 0.88).

Интерпретируемость: Модель позволяет объяснять предсказания через SHAP-значения.

Производительность: Время инференса 3.46 мс значительно лучше требований (< 50 мс).

Ограничения и направления улучшения:

- Сложности с оборудованием Type H (низкая Precision).
- Зависимость от качества данных датчиков.

Рекомендации для production:

- Реализовать мониторинг дрейфа данных (Evidently AI).
- Настроить автоматическое переобучение при снижении метрик.
- Внедрить A/B тестирование новых версий моделей.
- Модель готова к развертыванию в production-среде и интеграции с системами мониторинга оборудования.

7. Выводы

7.1 Достижения проекта

Разработанная система предиктивного обслуживания успешно достигла поставленных целей: создана высокопроизводительная ML-модель, превышающая целевые метрики (Recall 0.88, FPR 0.007, время инференса 3.46 мс), и спроектирована масштабируемая архитектура для интеграции с промышленными системами.

Ключевые достижения:

- Бизнес-эффективность: Прогнозируемая годовая экономия 49.5 млн рублей при сроке окупаемости < 6 месяцев.
- Технические показатели: Модель LightGBM с калиброванным порогом 0.18 обеспечивает оптимальный баланс между безопасностью (Recall) и экономической эффективностью.
- Архитектурная готовность: Реализована распределенная система с поддержкой real-time обработки, мониторинга и автоматического переобучения.

7.2 Ограничения и выявленные проблемы

Качество данных: Зависимость от точности показаний датчиков требуется дополнительная система валидации входящих данных.

Дисбаланс классов: Несмотря на применение advanced техник, сохраняется сложность в детектировании редких типов отказов.

Оборудование Type H: Сниженная точность (Precision 0.58) для высококачественного оборудования требует отдельной доработки модели.

Временная зависимость: Отсутствие явных временных меток в датасете ограничивает анализ трендов и паттернов деградации.

7.3 Рекомендации для внедрения и развития

Поэтапное внедрение: Начать с пилотной зоны (10-20 единиц оборудования) для валидации в реальных условиях.

Мониторинг production: Реализовать систему детектирования дрейфа данных (Evidently AI) и автоматического переобучения

Улучшение модели:

- Разработка отдельных моделей для разных типов оборудования.
- Внедрение ансамблевых методов для редких типов отказов.
- Интеграция данных о предыдущих ремонтах и обслуживании.

Расширение функционала:

- Прогнозирование оставшегося срока службы (RUL) в дополнение к бинарной классификации.
- Рекомендательная система по оптимальному времени обслуживания.
- Интеграция с системами планирования производства (MES).

Проект демонстрирует высокую готовность к промышленному внедрению. Все артефакты (код, модели, документация) доступны в репозитории и готовы к развертыванию. Дальнейшее развитие системы должно фокусироваться на интеграции с дополнительными источниками данных и адаптации под специфические условия конкретных производств.