

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**РАЗРАБОТКА КОМПЛЕКСНОЙ ПЛАТФОРМЫ ДЛЯ АНАЛИЗА
РИСКОВ, ПРОГНОЗИРОВАНИЯ И БИЗНЕС-АНАЛИТИКИ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студентки 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Мухиной Анастасии Алексеевны

Научный руководитель
доцент, к. ф.-м. н.

А. С. Иванова

Заведующий кафедрой
к. ф.-м. н., доцент

С. В. Миронов

Саратов 2026

ВВЕДЕНИЕ

В современных условиях ведения бизнеса как никогда велика потребность в инструментах, позволяющих качественно и быстро оценивать риски, строить прогнозы и анализировать ключевые показатели деятельности. Особенно ценятся программные решения, объединяющие в себе эти функции и предоставляющие пользователю удобный интерфейс для принятия управленческих решений.

Данная дипломная работа описывает разработку веб-приложения Business Analysis Platform — комплексного решения для анализа рисков, прогнозирования и бизнес-аналитики.

Цель работы – создание платформы, объединяющей инструменты оценки рисков, прогнозирования показателей и аналитики для поддержки управленческих решений.

В ходе разработки приложения необходимо решить следующие задачи:

1. Разработать комплексную систему анализа рисков и бизнес-прогнозирования.
2. Спроектировать модули расчета метрик для анализа рисков, прогнозирования и бизнес-аналитики.
3. Подготовить интерфейс пользователя для удобной загрузки данных.
4. Реализовать расчет классических показателей по загруженным данным.
5. Настроить и обучить ML-модели для анализа и прогнозирования.
6. Свести финальные расчеты в единую отчетность.

1 Описание разработки системы

Приложение VaP реализовано на клиент-серверной архитектуре. Взаимодействие между фронтендом и бэкендом осуществляется через REST API, все компоненты контейнеризуются в Docker. Серверная часть содержит бизнес-логику и ML-модели, клиентская часть отвечает за отображение данных и взаимодействие с пользователем. Архитектура системы представлена на рисунке 1.1

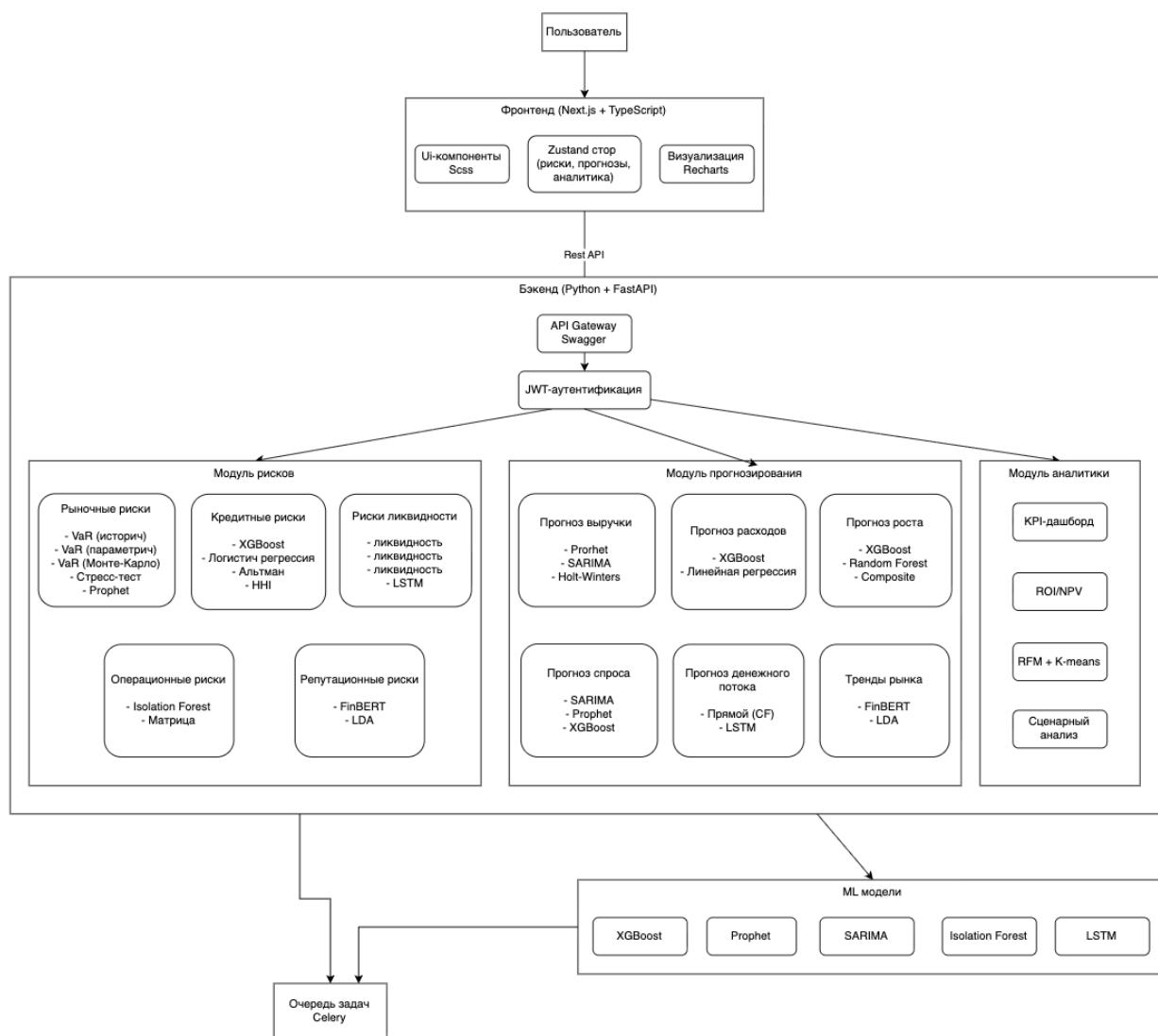


Рисунок 1.1 – Архитектура системы

Клиентская часть приложения реализована на TypeScript с использованием Next.js, стили написаны на препроцессоре SCSS. Проект организован по принципам Feature-Sliced Design (FSD), в качестве хранилища выбран стейт-

менеджер Zustand.

Серверная часть написана на Python с использованием фреймворка FastAPI, база данных — PostgreSQL, аутентификация реализована с помощью JWT-токенов.

Математические расчёты и ML-модели реализованы с использованием библиотек NumPy, Pandas, scikit-learn, XGBoost, Prophet, PyTorch и других. Для выполнения асинхронных задач используется Celery.

1.1 Подготовка данных и обучение моделей

Пользователь загружает исторические данные через CSV-файл или вводит их вручную: выручку, расходы, остатки денежных средств, финансовые показатели контрагентов. На бэкенде выполняется предобработка: удаление дубликатов, заполнение пропусков (линейная интерполяция для временных рядов, медиана для табличных данных), удаление выбросов методом Z-оценки, нормализация числовых признаков. Данные разделяются на обучающую (70%), валидационную (15%) и тестовую (15%) выборки.

Пример кода обучения Prophet:

```
1 @celery.task
2 def train_revenue_prophet(company_id):
3     df = get_historical_revenue(company_id)
4     df = preprocess_timeseries(df, 'date', 'revenue')
5     model = Prophet(yearly_seasonality=True)
6     model.fit(df.rename(columns={'date': 'ds', 'revenue': 'y'}))
7     joblib.dump(model, f'models/revenue_prophet_{company_id}.pkl')
8     return {"status": "success"}
```

1.2 Аутентификация и пользовательский сценарий

Для входа в систему создана страница авторизации, представленная на рисунке 1.2. Авторизация настроена через JWT-токены, которые хранятся 15 минут. При успешном входе сервер генерирует токен, подписанный секретным ключом, и возвращает его клиенту. Токен передаётся в заголовке Authorization для всех защищённых запросов. Также есть страница регистрации для новых пользователей, где предлагается ввести имя и контактную информацию. Для полей номера и почты используются правила, ограничивающие ввод некорректных символов.

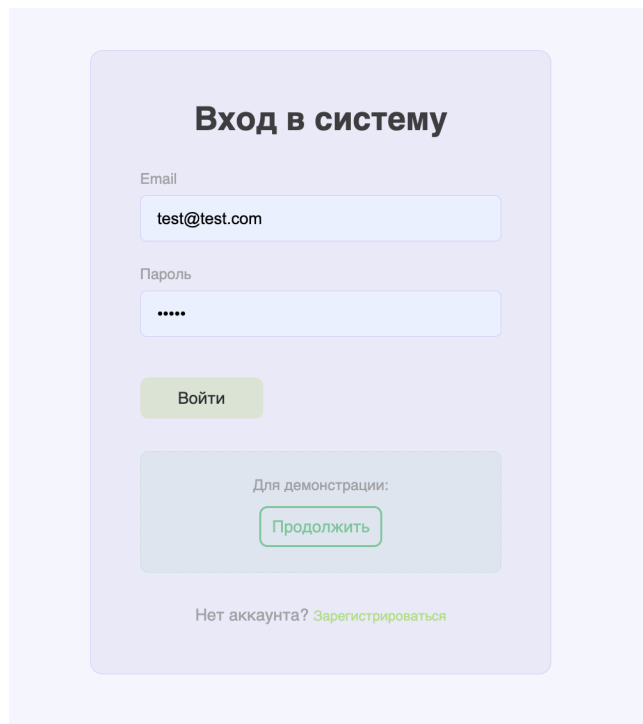


Рисунок 1.2 – Авторизация пользователя

Для авторизации реализованы эндпоинты в Swagger, а также эндпоинты для получения и хранения данных пользователя в БД. После регистрации пользователь попадает в личный кабинет, где настраивает информацию о своей компании для работы в приложении.

1.3 Проектирование базы данных

База данных реализована на PostgreSQL с использованием нормализованных таблиц и JSONB-полей для гибкого хранения данных. Основные сущности: пользователи, компании, контрагенты, результаты расчётов рисков, прогнозы и отчёты. Связи между таблицами организованы через внешние ключи, для ускорения запросов созданы индексы по часто используемым полям (`company_id`, `risk_type`, `created_at`, `date`).

Таблица `users` содержит учётные записи с хешированными паролями и ролями; `companies` – информацию о компаниях (валюта, отрасль, пороговые значения); `counterparties` – контрагентов с финансовыми показателями в JSONB-полях (`financials`, `features`), что позволяет гибко добавлять признаки. Таблица `market_data` хранит временные ряды с составным индексом (`asset_name`, `date`); `forecasts` – результаты прогнозов с метаданными (тип модели, горизонт, MAPE); `risk_predictions` – риск-метрики (PD, VaR) с привязкой к контрагентам.

Пример структуры таблицы counterparties:

```
1 class Counterparty(Base):
2     __tablename__ = "counterparties"
3     id = Column(String, primary_key=True)
4     name = Column(String(255), nullable=False)
5     financials = Column(JSON)
6     features = Column(JSON)           # дополнительные признаки для ML
7     share = Column(Float)            # доля в выручке
8     default_flag = Column(Boolean, default=False)
9     created_at = Column(DateTime(timezone=True), server_default=func.now())
```

1.4 Функциональные модули системы

1.4.1 Модуль анализа рисков

Модуль анализа рисков включает пять категорий: рыночные, кредитные, ликвидности, операционные и репутационные риски. Интерфейс модуля представлен на рисунке 1.3.

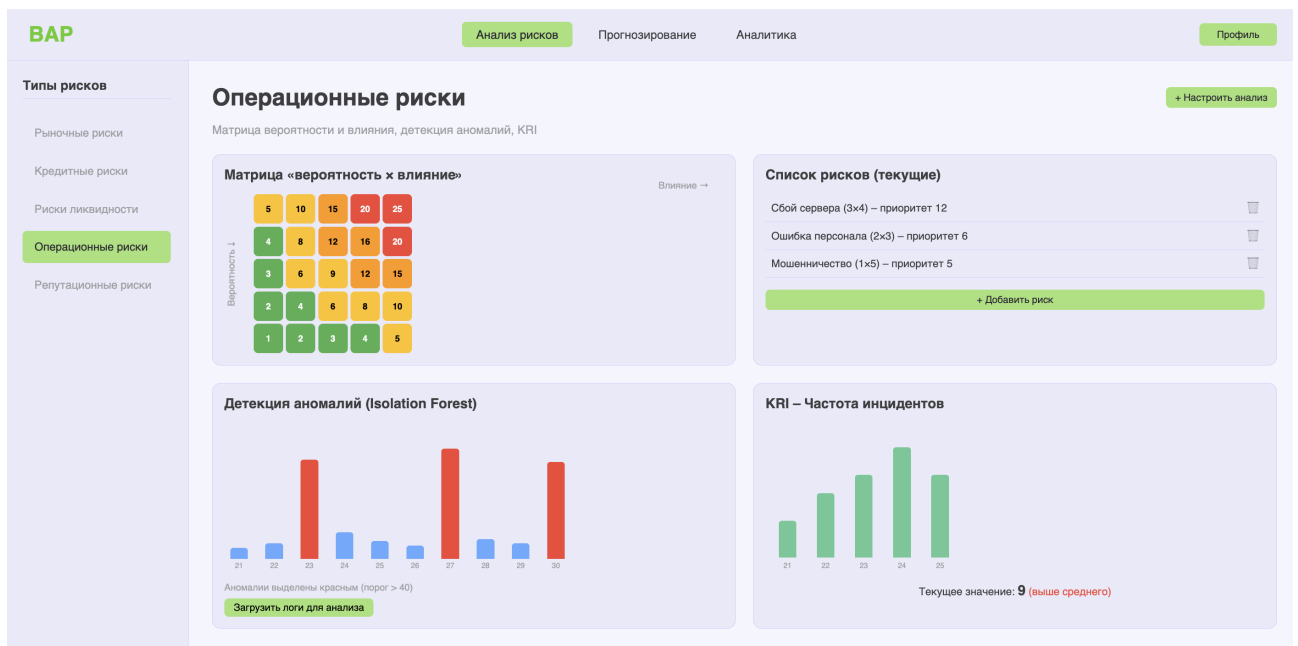


Рисунок 1.3 – Модуль анализа рисков

Рыночные риски оцениваются с помощью трёх методов VaR: исторического (квантиль прошлых доходностей), параметрического (предполагает нормальное распределение, $VaR = -(\mu + \sigma\Phi^{-1}(1 - \alpha))$) и Монте-Карло (симуляция 10 000 траекторий). Также реализованы стресс-тестирование и прогноз волатильности на основе Prophet.

Кредитные риски рассчитываются с использованием логистической регрессии и XGBoost. Входные признаки: кредитное плечо, ликвидность, рентабельность, размер компании, код отрасли. Формула вероятности дефолта:

$$PD = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n)}}.$$

Также рассчитываются модель Альтмана (Z-Score) и индекс концентрации ННП.

Риски ликвидности представлены тремя коэффициентами (абсолютной, срочной и текущей ликвидности) и прогнозом денежного потока на основе LSTM (MAPE = 9,5%). Модель LSTM с одним слоем обучена на исторических остатках денежных средств.

Операционные риски оцениваются через матрицу «вероятность – влияние» и Isolation Forest для обнаружения аномалий. Оценка аномальности:

$$s(x, n) = 2^{-\mathbb{E}[h(x)]/c(n)},$$

где $h(x)$ — глубина точки, $c(n)$ — средняя глубина. Чем ближе s к 1, тем вероятнее аномалия.

```
1 def detect_anomalies(values: list[float]) -> list[int]:
2     if len(values) < 5:
3         return []
4     X = np.array(values).reshape(-1, 1)
5     model = IsolationForest(contamination=0.1, random_state=42)
6     model.fit(X)
7     preds = model.predict(X)
8     return [i for i, p in enumerate(preds) if p == -1]
```

Репутационные риски анализируются с помощью FinBERT. Модель классифицирует тексты на негативные, нейтральные и позитивные, вычисляется индекс тональности.

1.4.2 Модуль прогнозирования

Модуль прогнозирования предназначен для построения прогнозов ключевых финансовых и операционных показателей компании на основе исторических данных. Прогнозирование в системе базируется на анализе временных рядов — последовательностей данных, фиксируемых с определённой периодичностью. Любой временной ряд раскладывается на тренд (долгосрочное изменение

среднего уровня), сезонность (периодические колебания) и случайную составляющую. Для приведения рядов к стационарному виду применяются дифференцирование, логарифмирование и Вох-Сох трансформация, стационарность проверяется тестом Дики-Фуллера. Интерфейс модуля представлен на рисунке 1.4.

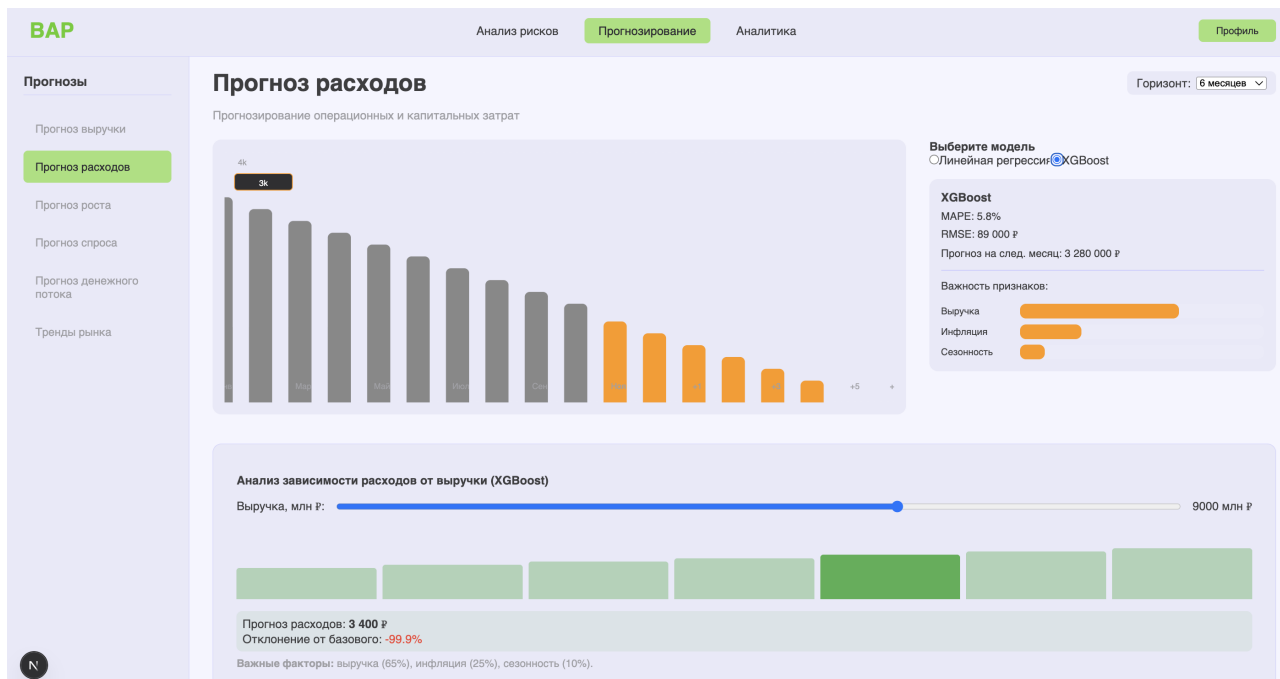


Рисунок 1.4 – Модуль прогнозирования

Модуль включает шесть направлений: прогноз выручки, расходов, роста (CAGR), спроса, денежного потока и трендов рынка. Для каждого направления реализованы несколько моделей, выбор лучшей производится автоматически на основе метрик качества (MAPE, RMSE) на тестовой выборке.

Прогноз выручки использует Prophet и Holt-Winters. Prophet раскладывает временной ряд на три компоненты:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t,$$

где $g(t)$ — тренд (кусочно-линейный), $s(t)$ — сезонность (моделируется рядами Фурье), $h(t)$ — праздничные эффекты. Модель автоматически определяет точки изменения тренда с помощью лапласовского априорного распределения. Для тестовой компании Prophet показал MAPE = 8,2%, Holt-Winters — MAPE = 11,5%. Prophet выбран как основная модель благодаря автоматическому учёту сезонности и устойчивости к пропускам.

Прогноз расходов использует линейную регрессию (бенчмарк) и XGBoost (основная модель). Признаки: выручка текущего и предыдущих месяцев, инфляция, сезонный индекс, рекламный бюджет. Анализ важности признаков показал, что главный драйвер расходов — прошлая выручка.

Прогноз среднегодового темпа роста (CAGR) основан на XGBoost с макроэкономическими признаками: текущий CAGR, прогноз ВВП, ключевая ставка, инфляция, курс валюты. Модель показала MAPE = 9,1%. Наиболее значимые факторы: ключевая ставка (0,38) и ВВП (0,32). Прогноз на следующие 3 года: базовый — 12%, оптимистичный — 18%, пессимистичный — 5%.

Прогноз спроса реализован гибридным подходом: ARIMA для выделения базового тренда и сезонности + XGBoost для учёта маркетинговых воздействий (акции, промо, изменение цены). Итоговый прогноз = прогноз ARIMA + поправка XGBoost. Гибридная модель показала MAPE = 7,8%, что лучше каждой модели по отдельности.

Прогноз денежного потока использует прямой метод (на основе прогнозов выручки, расходов, оборачиваемости дебиторской и кредиторской задолженности) и LSTM. LSTM с одним слоем обучен на исторических остатках денежных средств за последние 30 дней. MAPE = 9,5%, что позволяет выявлять кассовые разрывы за 2 месяца до их наступления.

Тренды рынка анализируются с помощью методов обработки естественного языка. Для выделения ключевых тем используется LDA (Latent Dirichlet Allocation), для анализа тональности — FinBERT. Система парсит новостные ленты и отзывы клиентов, результаты агрегируются в пузырьковой диаграмме (размер пузыря — объём упоминаний, цвет — тональность от негативной до позитивной).

1.4.3 Аналитический модуль

Аналитический модуль предназначен для преобразования сырых данных и результатов прогнозирования в наглядные показатели, помогающие принимать управленческие решения. В отличие от модулей рисков и прогнозирования, которые отвечают на вопросы «что может пойти не так» и «что будет», аналитика даёт ответ «насколько это эффективно и как можно улучшить». Модуль включает четыре ключевых раздела: ключевые показатели эффективности (KPI), оценка эффективности инвестиций (ROI/NPV), сегментация клиентов и сценарный анализ. Интерфейс модуля представлен на рисунке 1.5.

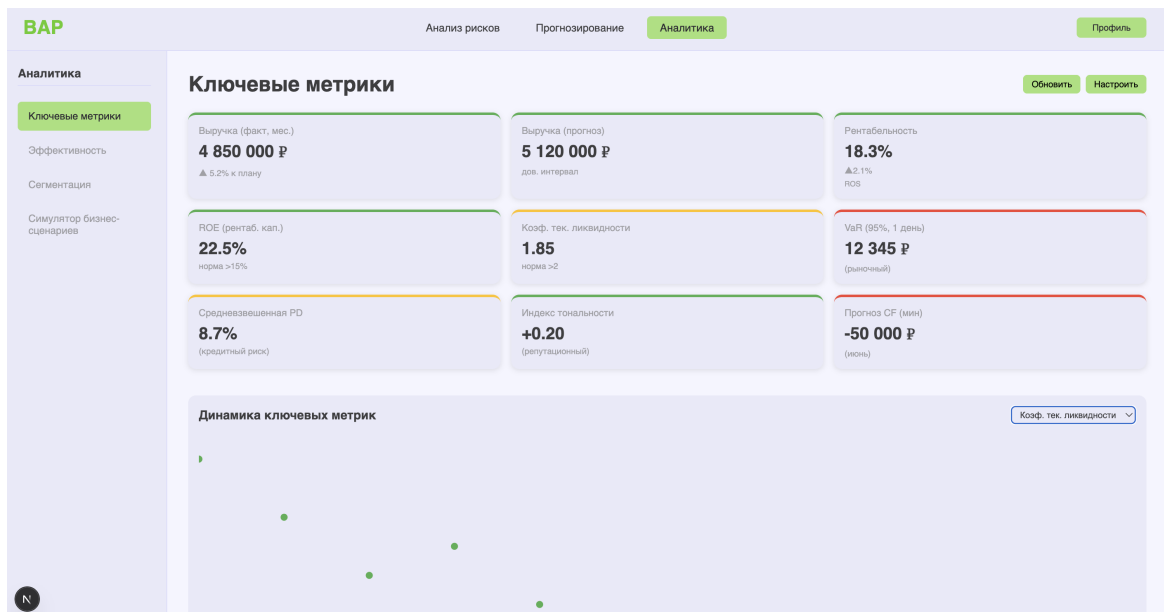


Рисунок 1.5 – Аналитический модуль

Ключевые показатели (KPI). Дашборд KPI является центральным экраном блока аналитики, где пользователь видит «здоровье» бизнеса в реальном времени. Показатели собираются из соседних модулей, а также рассчитываются производные метрики: рентабельность и оборачиваемость. Все KPI отображаются в виде карточек с цветовой индикацией: зелёный — норма, жёлтый — предупреждение, красный — критическое значение. Пользователь может настроить, какие метрики выводить на главный экран.

Оценка эффективности (ROI и NPV). Для оценки эффективности инвестиционных проектов в VAP реализованы две классические метрики: ROI (рентабельность инвестиций) и NPV (чистая приведённая стоимость). Пользователь вводит начальные инвестиции, ожидаемые денежные потоки по годам и ставку дисконтирования. Система автоматически рассчитывает ROI и NPV и выдаёт рекомендацию «Принять проект» или «Отклонить». Формулы расчёта:

$$ROI = \frac{\text{Прибыль}}{\text{Инвестиции}} \times 100\%, \quad NPV = -\text{Инвестиции} + \sum_{t=1}^n \frac{CF_t}{(1+r)^t}.$$

Если NPV положительный, проект считается экономически эффективным.

```

1 def calculate_roi_npv(
2     investment: float, cashflows: list, discount_rate: float
3 ) -> dict:
4     profit = sum(cashflows) - investment

```

```

5     roi = (profit / investment) * 100 if investment != 0 else 0.0
6     npv = -investment + sum(
7         cf / (1 + discount_rate / 100) ** i
8         for i, cf in enumerate(cashflows, start=1)
9     )
10    return {
11        "roi": round(roi, 2), "npv": round(npv, 2),
12        "profit": round(profit, 2),
13        "recommendation": "Принять" if npv > 0 else "Отклонить"
14    }

```

Сегментация клиентов. Сегментация позволяет разбить клиентскую базу на группы для персонализированного маркетинга и управления рисками. Реализованы два метода: RFM-анализ и K-means кластеризация. RFM-анализ основан на трёх факторах: давность последней покупки (Recency), частота покупок (Frequency) и суммарные траты (Monetary). Каждый клиент получает оценку от 1 до 5 по каждому фактору, после чего клиенты группируются в сегменты: «лояльные», «спящие», «новые», «потерянные». K-means выполняет автоматическую кластеризацию без жёстких правил, результаты визуализируются с помощью PCA для наглядного представления групп.

Сценарный анализ (What-if). Сценарный анализ дает возможность пользователю самому менять ключевые параметры (цена, объём продаж, процентная ставка, курс валюты) и мгновенно видеть, как изменятся прогнозы выручки, расходов и денежного потока, а также ключевые риск-метрики (VaR, PD). Интерфейс представляет собой набор ползунков: при изменении параметра система отправляет запрос на бэкенд, который пересчитывает прогнозы на основе простой эластичности.

ЗАКЛЮЧЕНИЕ

В ходе разработки была создана система сценарного планирования, анализа и прогнозирования рисков, реализовано полноценное приложение с соответствующими модулями и возможностью создания единого отчета. На примере тестовой компании показаны регистрация, заполнение данных, разделение ролей, запуск расчетов и готовый результат прогнозирования. В работе использовались формулы расчета показателей и ключевых метрик, а также модуль машинного обучения и предсказания.