

Web:

Streamlit-приложение описано в `src/app.py`.

Основные функции:

Основные блоки:

1. Выбор клиента

- в сайдбаре:
 - ввод `client_id`;
 - выпадающий список с небольшим диапазоном ID для быстрого выбора;
- при выборе клиента фронтенд вызывает метод `api.predict(client_id)`.

2. Прогноз дохода

- отображается основной показатель «Ожидаемый доход» (`income_prediction`);
- дополнительно рассчитывается и выводится доверительный интервал $\pm 8\%$;
- рядом показывается сегмент клиента (low/medium/high).

3. Объяснение прогноза (SHAP-подобный блок)

- вкладка «Объяснение прогноза (SHAP)»;
- строится горизонтальный bar-chart (Plotly) по `shap_explanation`:
 - ось Y - русские названия признаков (`feature_mapping` из `data/feature_mapping.json` или из дефолтного словаря);
 - ось X «важность»/влияние;
 - цвет бара показывает знак влияния (положительное/отрицательное).

4. Персональные рекомендации

- вкладка «Персональные рекомендации»;
- выводится список рекомендованных продуктов (ипотека, кредитные карты, накопительные счета), зависящий от уровня дохода и сегмента.

5. What-if анализ

- вкладка «Что если?»;
- три контролируемых параметра:
 - слайдер изменения зарплаты за 6–12 месяцев;
 - слайдер изменения оборотов по счетам;

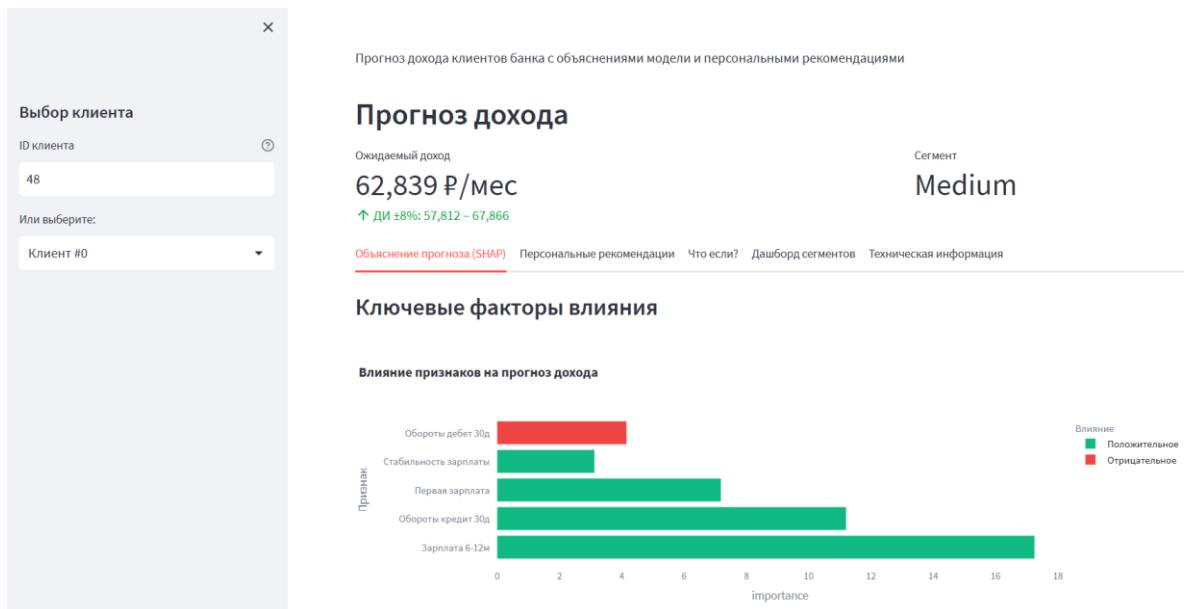
- слайдер «стабильности» зарплаты;
- по нажатию кнопки вызывается `api.what_if(client_id, changes)`;
- отображаются:
 - дельта дохода (`metric delta_income`);
 - новый прогноз дохода;
 - краткое резюме внесённых изменений.

6. Дашборд сегментов

- вкладка «Дашборд сегментов»;
- запрос к `api.segments()`;
- круговая диаграмма (pie chart) распределения клиентов по сегментам;
- таблица солями сегментов и общее число клиентов.

7. Техническая информация

- таблица с индивидуальными предсказаниями трёх моделей;
- строка с метрикой качества модели.



Прогноз дохода

Ожидаемый доход
105,885 ₽/мес
↗ ДИ ±8%: 97,414 – 114,356

Сегмент **Stable High Income**

Объяснение прогноза (SHAP) Персональные рекомендации Что если? Дашборд сегментов Техническая информация

Рекомендуемые продукты

1. Ипотека (от 7.5%)
2. Инвестиции в фонды
3. Platinum карта

Логика подбора: прогноз дохода + SHAP-анализ + сегментация клиента

Made with Streamlit

Прогноз дохода клиентов банка с объяснениями модели и персональными рекомендациями

Прогноз дохода

Ожидаемый доход
105,885 ₽/мес
↗ ДИ ±8%: 97,414 – 114,356

Сегмент **Stable High Income**

Объяснение прогноза (SHAP) Персональные рекомендации **Что если?** Дашборд сегментов Техническая информация

Анализ сценариев

Зарплата 6-12m, ₽ Обороты счетов, ₽ Стабильность зарплаты

Пересчитать сценарий

Изменение дохода
6,668,600 ₽
↗ 6668600.0

Новый прогноз
6,774,485 ₽/мес

Изменения: salary_6to12m_avg: +7170, turn_cur_cr_avg_act_v2: -3836, salary_stability: +0

Прогноз дохода

Выбор клиента

ID клиента: 5

Ожидаемый доход: 105,885 ₽/мес

Сегмент: Stable High Income

Объяснение прогноза (SHAP) | Персональные рекомендации | Что если? | Дашборд сегментов | Техническая информация

Технические детали

	Предсказание, ₽
catboost	102,716
lightgbm	115,724
xgboost	105,218

Качество модели: WMAE: 41183

Прогноз дохода

Выбор клиента

ID клиента: 5

Ожидаемый доход: 105,885 ₽/мес

Сегмент: Stable High Income

Объяснение прогноза (SHAP) | Персональные рекомендации | Что если? | Дашборд сегментов | Техническая информация

Распределение клиентской базы

Сегменты клиентов

73,214 клиентов

Сегмент	Доля, %
0	94.6
1	0.5
2	4.9

Прогноз дохода

Ожидаемый доход: 105,885 ₽/мес

Сегмент: Stable High Income

↑ ДИ ±8%: 97,414 – 114,356

Объяснение прогноза (SHAP) | Персональные рекомендации | Что если? | Дашборд сегментов | Техническая информация

Технические детали

	Предсказание, ₽
catboost	102,716
lightgbm	115,724
xgboost	105,218

Качество модели: WMAE: 41183

FastAPI:

IncomePrediction API 1.0.0 OAS 3.1

/openapi.json

Прогнозирование дохода

default

GET /health/ Health Check

POST /predict Predict Client

POST /what_if What If Analysis

GET /segments Get Segments

GET /segments/dashboard Segments Dashboard

Все ручки описаны в `src/main.py`.

1. GET /health/

- Назначение: проверка работоспособности сервиса и базовой статистики.
- Ответ: статус сервера, версия моделиб аптайм, количество загруженных клиентов, флаг обучения сегментатора, значение wmae_validation.

2. POST /predict

Вход (JSON):

- { "client_id": int }

Логика:

- по client_id из X_test поднимается строка клиента
- вызывается prediction, формирующий:
 - единое числовое значение дохода;
 - три предсказания (catboost, lightgbm, xgboost);
 - топ-5 признаков с SHAP-подобными значениями;
 - строку с качеством модели;
- дополнительно определяется сегмент клиента и список рекомендаций.

Ответ (JSON, схема PredictResponse):

- client_id
- income_prediction (float)
- individual_predictions: {catboost, lightgbm, xgboost}

- shap_explanation: список из 5 объектов {feature, shap_value, importance}
- model_quality: строка вида "WMAE: 0.0000"
- segment: строковый код сегмента
- recommendations: список текстовых рекомендаций.

3. POST /what_if

Вход (JSON):

- {
 "client_id": int,
 "feature_changes": { "feature_name": float, ... }
}

Логика:

- пересчитывает новый прогноз дохода, исходя из суммы изменений feature_changes;
- возвращает новое значение дохода и дельту относительно базового прогноза.

Ответ (JSON, схема WhatIfResponse):

- client_id
- new_income_prediction
- delta_income
- changes_applied (повтор feature_changes).

4. GET /segments

Назначение: агрегированное распределение клиентов по сегментам для дашборда.

Ответ: объект вида {"segments": {"segment_name": count, ...}}.

5. GET /segments/dashboard

Назначение: расширенные данные для дашборда сегментации.

Ответ включает:

- распределение по сегментам,
- общее число клиентов,
- пример важности признаков для сегментации,
- пороговые значения дохода для границ сегментов.

Структура проекта

Корень репозитория:

data/

- исходные и производные данные (train/test, X_test_final, parquet/CSV и служебные файлы);

logs/

- логи работы сервисов;

models/

- финальные ML-артефакты, в том числе final_3models_with_shap.pkl (ансамбль + SHAP explainer);

notebooks/

- model_income_pipeline_v2.ipynb - основной ноутбук с анализом данных, построением фич и финального ансамбля;
- upgraded_model_income_v2.ipynb - дополнительные эксперименты над моделью;

src/

- main.py - FastAPI backend (текущая версия с заглушками);
- app.py - Streamlit интерфейс для сотрудников банка;
- model.py - класс для интеграции ансамблевой модели в backend (каркас, ждет доработки);
- preprocessing.py - предобработка данных и функции подготовки одного клиента к инференсу;
- recommendations.py - генерация текстовых рекомендаций по продуктам на основе дохода, сегмента и SHAP-факторов;
- segment.py - логика кластеризации клиентов и формирования сегментов/дашбордов;
- logging_config.py - конфигурация логирования приложения;

README.md - описание проекта и инструкция по запуску;

requirements.txt - список зависимостей;