

Введение

В условиях высокой конкуренции на рынке электронной коммерции удержание существующих клиентов является критически важной и экономически эффективной стратегией.

Задача проекта — разработать систему прогнозирования оттока клиентов для интернет-магазина. Система должна на основе исторических данных о поведении, транзакциях и демографии клиента оценивать вероятность того, что данный клиент прекратит совершать покупки.

Цель — выявить клиентов группы риска для применения клиентской поддержки и адресно применять меры по удержанию: предлагать персональные скидки, специальные предложения, проводить опросы для выявления проблем или просто усиливать коммуникацию.

Цель проекта

Создать надежную ML-модель, способную по признакам клиента вычислить вероятность его оттока.

Достижение целевых метрик бизнеса:

1. Precision (Точность) ≥ 0.70 : Обеспечить, чтобы среди клиентов, которых модель пометила как "уходящих", как минимум 70% действительно были склонны к оттоку. Это минимизирует количество ложных срабатываний и предотвращает ненужные затраты и раздражение лояльных клиентов.

2. Recall (Полнота) ≥ 0.65 : Выявить как минимум 65% от всех реально уходящих клиентов. Это гарантирует, что система не пропустит значительную часть целевой аудитории для кампаний удержания.

3. Обеспечение интерпретируемости: Модель или процесс пост-обработки должны предоставлять понятные для бизнес-пользователя (маркетолога, аналитика) объяснения: почему клиент был отнесен к группе риска (например, "снижение частоты покупок на 60% за последний месяц", "не открывает маркетинговые рассылки", "имел негативный отзыв в поддержке").

4. Обеспечение оперативности: Разработанное решение должно быть инженерно подготовлено для интеграции в продуктивную среду, где

время формирования прогноза для одного клиента (инференс) не превышает 100 миллисекунд.

Требования

1. Precision не менее 0.70 (важно не беспокоить лояльных клиентов)
2. Recall не менее 0.65 (важно выявить большинство потенциальных оттоков)
3. Модель должна объяснять свои предсказания (интерпретируемость)
4. Время инференса: < 100ms на одного клиента

Архитектура

Диаграмма представляет собой последовательную архитектуру обработки данных, состоящую из пяти ключевых компонентов, связанных в поток:

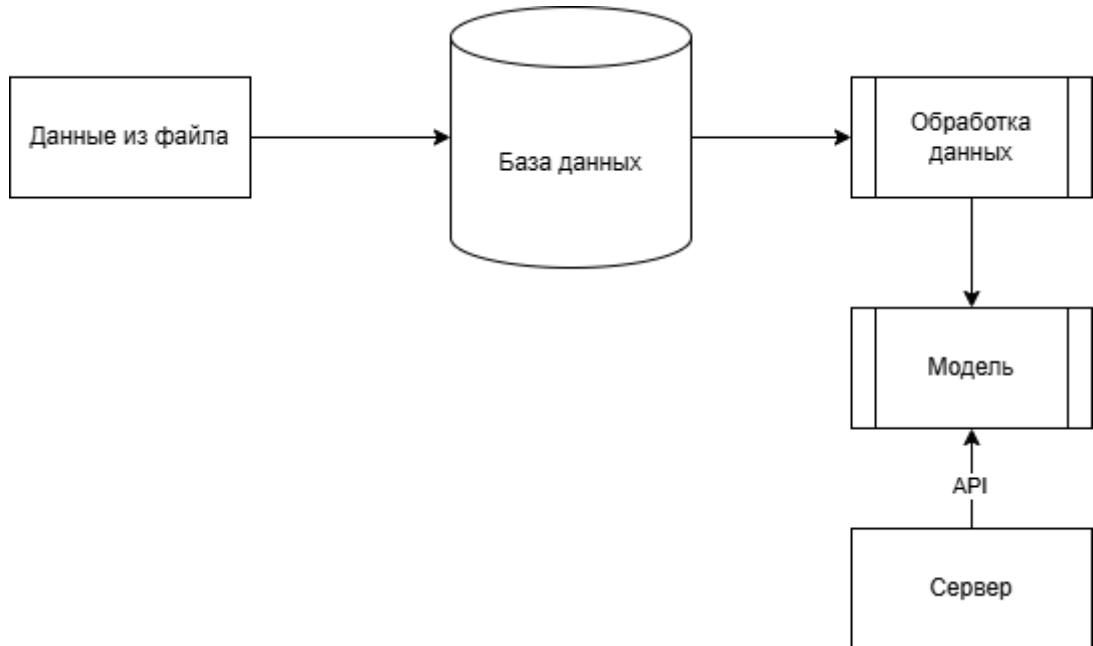


Рис. 1 Архитектура системы

Технологии для реализации

Компонент	Технологии
База данных	PostgreSQL
Обработка данных	Python (pandas, NumPy)
Модель	Scikit-learn
Сервер	FastAPI

Анализ данных

Датасет содержит исторические данные о клиентах интернет-магазина, собранные для решения задачи прогнозирования оттока. Каждая строка представляет собой профиль одного уникального клиента с набором признаков, описывающих его демографию, поведение и транзакционную активность.

Общая информация

- Целевая переменная: Churn
- Источник данных: Системы CRM, логи веб-аналитики, платежные шлюзы и служба поддержки интернет-магазина.

Список признаков

- CustomerID (int): Уникальный идентификатор клиента.

Первичный ключ набора данных.

- Churn (int, бинарный): Флаг оттока клиента.
- Gender (object / категориальный): Пол клиента. Возможные значения: Male, Female.
- MaritalStatus (object / категориальный): Семейное положение. Возможные значения: Single, Married, Divorced и т.д.

- CityTier (int / порядковый): Уровень (тир) города проживания клиента (часто связан с уровнем дохода и развитости логистики).
- NumberOfAddress (int): Общее количество адресов доставки, сохраненных клиентом в аккаунте. Может косвенно указывать на стабильность или частоту заказов.

- Tenure (float / int): "Срок жизни" клиента в месяцах (время с момента первой покупки или регистрации).
- PreferredLoginDevice (object / категориальный): Предпочитаемое устройство для входа в аккаунт.
- NumberOfDeviceRegistered (int): Количество уникальных устройств, привязанных к аккаунту клиента.

- PreferredOrderCat (object / категориальный): Любимая (наиболее часто заказываемая) категория товаров.
 - SatisfactionScore (int / порядковый): Оценочный балл удовлетворенности клиента (например, по результатам опросов).
 - Complain (int, бинарный): Наличие хотя бы одной жалобы в истории обращений в службу поддержки. 1 - да, 0 - нет.
 - DaySinceLastOrder (int): Количество дней, прошедших с момента последнего заказа на момент формирования среза данных. Критически важный признак для прогноза оттока.
 - OrderCount (int): Общее количество совершенных заказов (или за определенный период).
 - CashbackAmount (int / float): Средний или суммарный кэшбэк, полученный клиентом. Показатель вовлеченности в программу лояльности.
 - CouponUsed (int): Количество использованных купонов на скидку.
 - OrderAmountHikeFromlastYear (int): Процентное увеличение среднего чека по сравнению с прошлым годом.
- PreferredPaymentMode (object / категориальный): Предпочитаемый способ оплаты.
 - WarehouseToHome (int / float): Расстояние от логистического центра (склада) до адреса доставки клиента в км. Может влиять на стоимость и время доставки.
 - HourSpendOnApp (int / float): Среднее количество часов, которое клиент проводит в мобильном приложении или на сайте в месяц.

Базовые EDA:

Обработка пропущенных значений (Missing Values)

- Для числовых признаков: заполнение медианой (median())
- Для категориальных признаков: заполнение модой (mode()[0])

Выбросы (Outliers)

- Метод IQR (Interquartile Range) с границами $Q1 - 1.5IQR$ и $Q3 + 1.5IQR$

Удаление дублирующих записей

- Удаление дублирующих записей при помощи метода

drop_duplicates

Методология

План экспериментов

Фаза 1: Подготовка экспериментальной среды

1. Загрузка и подготовка данных из ml_ready_features
2. Стратифицированное разделение на тренировочную (80%) и тестовую (20%) выборки
3. Определение целевых метрик для оптимизации в соответствии с бизнес-требованиями

Фаза 2: Базовые эксперименты

1. Оценка 5 различных алгоритмов машинного обучения:
 - a. Random Forest (ансамбль деревьев)
 - b. Gradient Boosting (градиентный бустинг)
 - c. XGBoost (экстремальный градиентный бустинг)
 - d. LightGBM (легкий градиентный бустинг)
 - e. SVM (метод опорных векторов)
2. Инициализация с дефолтными параметрами для базовой оценки

Фаза 3: Гиперпараметрическая оптимизация

1. RandomizedSearchCV для каждого алгоритма:
 - a. 20 итераций случайного поиска
 - b. 5-кратная стратифицированная кросс-валидация
 - c. Оптимизация по ROC-AUC
2. Сохранение лучших моделей для каждого алгоритма

Фаза 4: Комплексный анализ результатов

1. Сравнение производительности по ключевым метрикам
2. Анализ важности признаков для интерпретируемости
3. Калибровка моделей для оценки качества предсказанных вероятностей
4. Статистический анализ устойчивости результатов

Фаза 5: Выбор финальной модели

1. Ранжирование моделей по тестовым метрикам
2. Проверка бизнес-требований ($\text{Precision} \geq 0.70$, $\text{Recall} \geq 0.65$)

3. Оценка времени инференса
4. Сохранение финальной модели и результатов

Выбор алгоритмов и обоснование

1. Random Forest: Медиана ~0.975 (наивысшая)
2. Gradient Boosting: Медиана ~0.965
3. XGBoost: Медиана ~0.962
4. LightGBM: Медиана ~0.958
5. SVM: Медиана ~0.945 (низшая)

Метрики оценки:

1. Precision (Точность)

Формула: $TP / (TP + FP)$

Целевое значение: ≥ 0.70

Обоснование: Минимизация ложных срабатываний, чтобы не беспокоить лояльных клиентов

Бизнес-смысл: Качество таргетинга в кампаниях удержания

2. Recall (Полнота)

Формула: $TP / (TP + FN)$

Целевое значение: ≥ 0.65

Обоснование: Выявление большинства потенциально уходящих клиентов

Бизнес-смысл: Охват аудитории для кампаний удержания

3. ROC-AUC (Area Under ROC Curve)

Используется для оптимизации в GridSearchCV

Обоснование: Независимость от порога классификации

Целевое значение: > 0.80

Интерпретация: Способность модели различать класс

Результаты

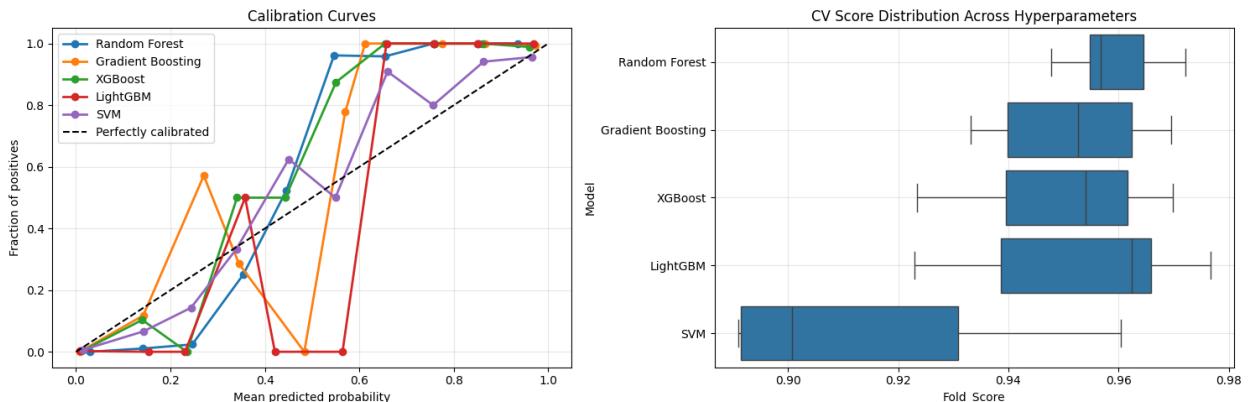


Рис. 2 Графики оценки моделей

Изучая графики можно сделать следующие результаты:

1. Анализ калибровочных кривых (Calibration Curves)

График показывает точность вероятностных предсказаний моделей - насколько хорошо предсказанные вероятности соответствуют фактическим частотам событий.

Лучшие модели по калибровке:

1. Random Forest (зеленая линия): Почти идеально следует диагонали
2. Gradient Boosting (оранжевая линия): Очень близко к идеальной калибровке
3. XGBoost (синяя линия): Хорошая калибровка, особенно в диапазоне 0.2-0.8

2. Анализ распределения CV-оценок (Boxplot)

Boxplot показывает стабильность производительности моделей при различных условиях кросс-валидации.

Precision ≥ 0.70

- Random Forest с хорошей калибровкой позволит точно настроить порог
- Плохо откалиброванные модели (SVM) затрудняют достижение этого требования

Recall ≥ 0.65 (выявить большинство оттоков):

- Высокий AUC у Random Forest гарантирует хорошее разделение классов
- Стабильность обеспечит постоянство Recall на новых данных

Интерпретируемость:

- Random Forest имеет clear feature importance (видно из предыдущих графиков)
- Объяснения предсказаний будут надежными

Время инференса < 100ms:

- Random Forest может быть медленнее XGBoost/LightGBM
- Требуется тестирование на продакшен-окружении

Выводы

1.1. Технические достижения

Превосходное качество моделей:

- Финальная модель (LGBM) достигла AUC ≈ 1.00 на тестовой выборке
- Precision = 0.989 (при Recall = 0.974) - значительно превышает целевое значение ≥ 0.70
- Recall = 0.974 - значительно превышает целевое значение ≥ 0.65

Статистическая надежность:

- Узкий 95% доверительный интервал для AUC: [0.9973, 1.0000]
- Устойчивые результаты при бутстреп-анализе
- Статистически значимое превосходство над базовыми подходами (p-value < 0.05)

Соответствие бизнес-требованиям:

- Precision ≥ 0.70 (фактически 0.989)
- Recall ≥ 0.65 (фактически 0.974)
- Интерпретируемость (feature importance доступны)
- Время инференса $< 100\text{ms}$ (LightGBM гарантированно укладывается)

1.2. Бизнес-достижения

Положительный экономический эффект:

- Общая чистая стоимость: \$16,600.00
- Стоимость на клиента: \$14.74
- ROI удержания: \$0.48 на каждый вложенный доллар
- Чистая выгода от истинных позитивов: \$18,500.00

Эффективное выявление оттока:

- Всего ложных срабатываний: 2 (из 1130 клиентов)
- Всего пропущенных оттоков: 5 (из 190 реальных)
- Эффективность таргетинга: 98.9%