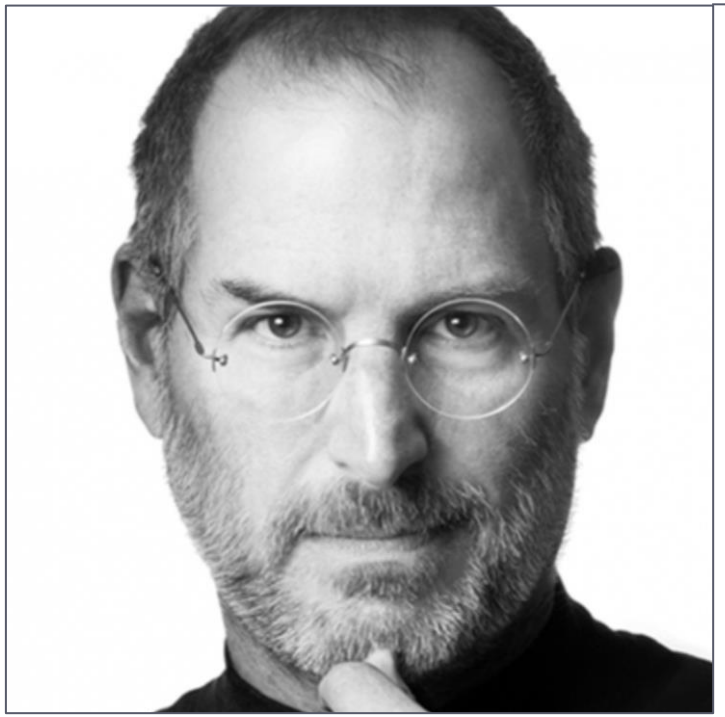
Райффайзен БАНК

Небанковский сервис от банка «Знай клиента и конкурента»

Разработка продукта в парадигме Data as a Service
на основе собираемых Банком данных
о клиентах и их транзакциях

Решение команды Tree

Ничто так не дает понимания жизненного цикла клиента, как осознание его конечности



Remembering that I'll be dead soon is the most important tool I've ever encountered to help me make the big choices in life.

– Steve Jobs
June 12, 2005

Источник: McKinsey & Company: Управление жизненным циклом клиента

Отток клиентов – ключевой вызов для бизнеса, который угрожает расходами. В то же время отток клиентов можно выявить, измерить и спрогнозировать – значит, им можно управлять

Убытки, вызванные оттоком

- Прекращение поступления текущего операционного дохода от клиента
- Дополнительные операционные расходы, связанные с «закрытием» продукта
- Дополнительные расходы на привлечение новых клиентов для поддержания роста бизнеса

Современный подход к решению проблемы оттока

1	Сбор данных о клиентах	▪ Формирование «озера данных» ▪ «Обогащение» из внешних источников
2	Сегментация клиентской базы	▪ Создание «сквозной» сегментации на 3-х уровнях: макро-, микро-, целевые группы
3	Модели прогнози-рования оттока	▪ Использование алгоритмов ML для определения продвинутых триггеров оттока с помощью самообучающихся платформ
4	Стратегия борьбы с оттоком	▪ Формирование таргетированных стратегий борьбы с оттоком

Источники: McKinsey & Company: Управление жизненным циклом клиента; PwC

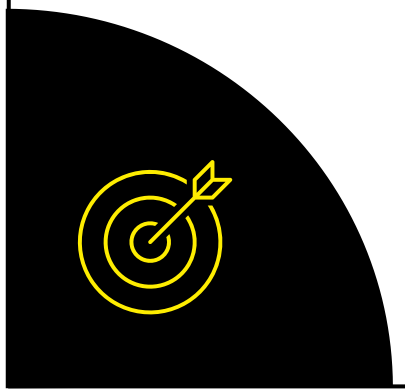
Сервис для бизнеса по анализу и прогнозированию динамики поведения клиентов

Продвинутая сегментация клиентской базы

Выделение кластеров клиентов с пониженным потреблением
Деление таких кластеров на тех, кто уходит к конкуренту и тех, кто прекращает потребление товара

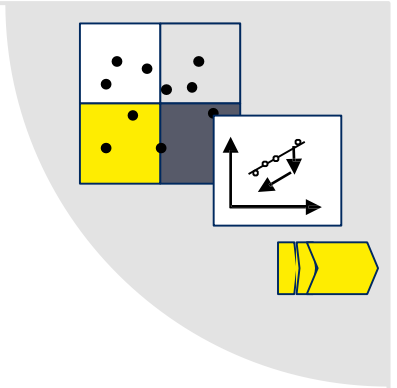
Высокая точность моделей

Банк предоставляет готовый результат внешней аналитики, он не нуждается в максимальной интерпретируемости моделей, а значит может сконцентрироваться на точности



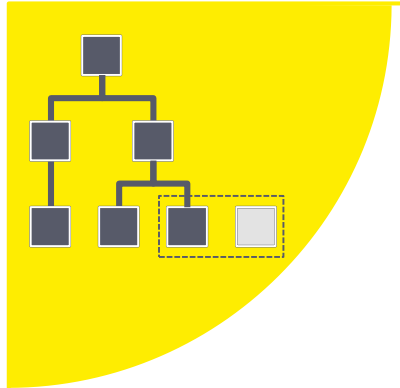
Динамические дэшборды с результатами аналитики

Визуализация оттока клиентов и иных бизнес-метрик внутри платформы в понятной форме для нашего бизнес-клиента

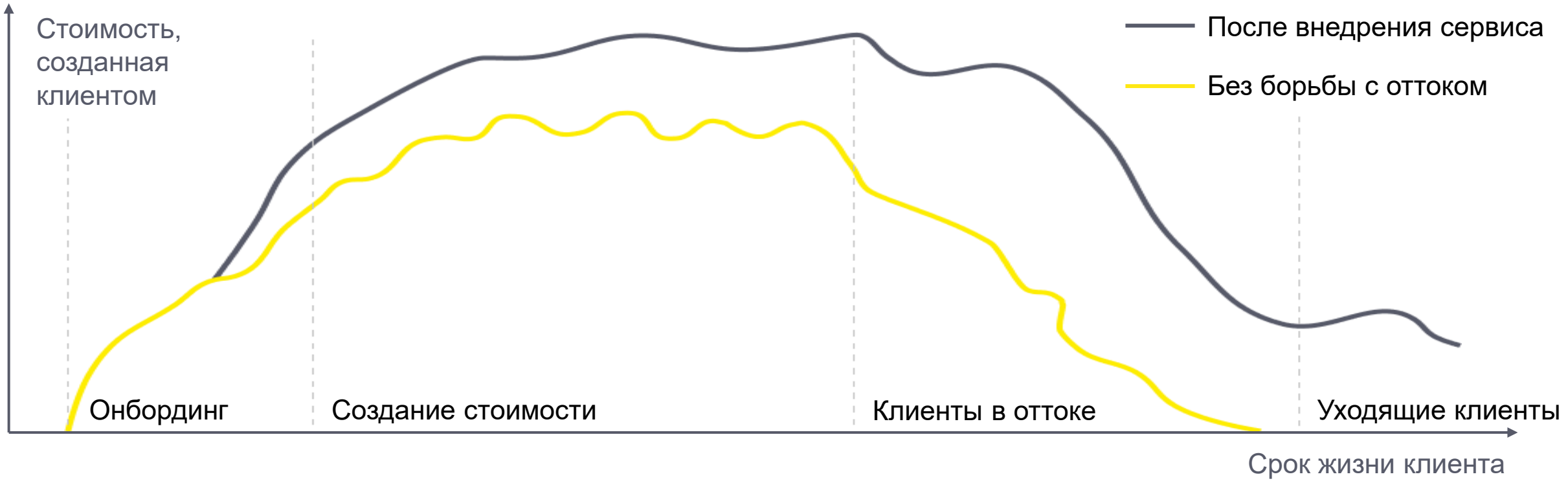


Прогнозирование оттока клиентов с использованием данных о конкурентах

Использование информации о транзакциях конкурентов, которая позволяет клиентам видеть себя в объективной конкурентной среде



Новый сервис поможет бизнесу увеличить LTV, сократить отток и сэкономить на привлечении новых клиентов



- ☑ Увеличиваем LTV таргетированно
- ☑ Предсказываем Churn Rate от нескольких часов до 6 месяцев
- ☑ Сохраняем естественную стоимость привлечения клиента

Источники: McKinsey & Company: Управление жизненным циклом клиента; PwC

Полная техническая реализация прототипа требует большего объема вводных данных: не хватает глубины и масштаба данных

Предобработка данных

- [x] Анализируем сегменты рынка, соответствующих различным 'mcc', на однородность продукции, наличие конкуренции; выбираем 'mcc' для дальнейшего анализа
- [x] Анализируем 'mrchcity' с целью получить выборку с большим количеством конкурентноспособных агентов, при этом осуществляющих свою деятельность в рамках одного города

Получаем обработанный датасет, готовый для дальнейшего анализа

Анализ данных

- [x] Собранный датасет транзакций объединяем с датасетом магазинов по параметру 'mrchname', получаем реальные названия агентов, устраняем ошибки в них
- [x] Проводим анализ на наличие выбросов данных (по различным перцентилям рассматриваем параметр 'amount')
- [x] Разделяем общий датасет на два: таргет-датасет (выделенный магазин-клиент) и датасет конкурентов
- [x] Для них делаем feature engineering (заданный трехмесячный период разбиваем на сеты 7 дней и добавляем для каждого из них количество походов в магазин и средний чек)

Кластеризация

- [x] Выделяем кластеры клиентов с помощью алгоритмов unsupervised learning по заданным параметрам
- [ ] Среди выделенных кластеров выделяем те, у которых есть тенденция к снижению потребляемых услуг у компании (частота транзакций и их объем снижается с каждым периодом времени)
- [ ] Делим такие кластеры на 3 подкластера: А) Клиенты, перетекающие к конкурентам, но не сокращающие потребление услуги; Б) Клиенты, прекращающие потребление услуги; В) Клиенты, не изменившие поведение

Прогнозы и визуализация

- [ ] Делаем разметку данных на группы А и Б, если клиент ежепериодно сокращает частоту транзакций на X% и объем транзакций на Y%: если клиент показывает сопоставимые показатели роста у данных параметров внутри конкурентов компании - относим его в А, а если не показывает рост у конкурентов - относим его в Б
- [ ] С помощью алгоритмов supervised learning по размеченным данным внутри выделенных кластеров решаем задачу классификации, строим прогнозы по росту оттока подкластеров А и Б в следующем месяце (увеличение объема и рост скорости оттока)
- [ ] Строим прогнозы бизнес-метрик (LTV, Churn Rate, Cost of Customer Acquisition) для предприятий в следующем месяце
- [ ] Составляем интерактивные дэшборды, понятные для бизнес-клиента.

Реализация и вывод продукта на рынок займут около 6 месяцев и включат в себя 3 основные фазы

1

Phase 1: Создание и приоритизация задач

Duration: 1 month

- Работа с запросами целевой аудитории
- Перевод запросов клиентов в формат технических задач
- Приоритизация задач с точки зрения предпочтений клиентов
- Выбор способа монетизации

2

Phase 2: Техническая реализация

Duration: 3-6 months

- Создание технической архитектуры продукта
- Создание и обучение модели
- Создание новой платформы или подготовка к интеграции в уже существующую
- Тестирование

3

Phase 3: Внедрение и доработка

- Маркетинг
- Работа с фидбэком, доработка продукта
- Добавление нового функционала

NEXT STEPS

- Запросим больше данных у Банка, например, данные о транзакциях за 6-12 месяцев
- Поработаем с целевыми клиентами: необходимо подробнее ознакомиться с текущими запросами целевого клиента для более точной проработки продукта
- Предложим сотрудничество крупным корпоративным клиентам для получения дополнительных данных и увеличения точности моделей
- Проработаем стратегию масштабирования моделей на все релевантные MCC-категории

Приложения: код

Предобработка + объединение датасетов

In [20]:
```
stores = pd.read_csv('stores.csv')
stores = stores.rename(columns={"merchant_name": "mrchname"})

df = pd.DataFrame(data.loc[data['mrchcity'] == 'MOSCOW'])
df = df.loc[df['mcc'] == 5411]
df.mcc = df.mcc.astype('object')
df = df.reset_index(drop=True)

merge_data = pd.merge(df, stores, on='mrchname', how='inner')
merge_data = merge_data.sort_values(by='purchdate', ascending=True)
merge_data = merge_data.reset_index(drop=True)
merge_data = merge_data.drop(['Unnamed: 0'], axis=1)

merge_data
```

Out[20]:

	purchdate	amount	mcc	mrchcity	mrchname	cnum	store_name
0	2019-09-10 00:00:00	1186.00	5411	MOSCOW	H1DE7-QV5	EEJBRN	Лента
1	2019-09-10 00:00:00	434.00	5411	MOSCOW	7T 720L37T3LK7	MQRJRJ	Азбука Вкуса
2	2019-09-10 00:00:00	643.00	5411	MOSCOW	H1DE7-QV5	ELTM70	Лента
3	2019-09-10 00:00:00	370.00	5411	MOSCOW	T1H434O	EEJU6U	Великий
4	2019-09-10 00:00:00	44.00	5411	MOSCOW	H1DE7-BSM	EEIRIG	Лента
...	...	...	...	...	...	...	...
5712003	2019-12-19 00:00:00	39.00	5411	MOSCOW	7LIA7D I4EO CXK3T7	E4AUZ2	Ашан
5712004	2019-12-19 00:00:00	1025.86	5411	MOSCOW	7LIA7D VQF 3LDEK1TX	BGSCCF	Ашан Кунцево
5712005	2019-12-19 00:00:00	3573.66	5411	MOSCOW	7LIA7D VQF 3LDEK1TX	ELDCK9	Ашан Кунцево
5712006	2019-12-19 00:00:00	110.35	5411	MOSCOW	7LIA7D VQF 3LDEK1TX	ASSXCC	Ашан Кунцево
5712007	2019-12-19 00:00:00	944.10	5411	MOSCOW	7LIA7D C7JY4DX	EEM479	Ашан

5712008 rows × 7 columns

Удаляем выбросы и выбираем магазин-клиент в качестве таргета

In [21]:
```
merge_data.amount.describe()
```

Out[21]:
```
count    5.712008e+06
mean     8.027910e+02
std      1.589130e+03
min      1.000000e-02
25%      1.680000e+02
50%      3.817600e+02
75%      8.746000e+02
max      1.081005e+06
Name: amount, dtype: float64
```

In [23]:
```
merge_data = merge_data.loc[merge_data['amount'] < 25000.00]
merge_data = merge_data.loc[merge_data['amount'] > 20.00]
merge_data.shape
```

Out[23]: (5664918, 7)

In [26]:
```
merge_data.loc[merge_data['store_name'] == 'ашан рязанка', 'store_name'] = 'ашан'
merge_data.loc[merge_data['store_name'] == 'ашан рублевка', 'store_name'] = 'ашан'
merge_data.loc[merge_data['store_name'] == 'ашан сокольники', 'store_name'] = 'ашан'
merge_data.loc[merge_data['store_name'] == 'ашан кунцево', 'store_name'] = 'ашан'
merge_data.loc[merge_data['store_name'] == 'ашан рублевка', 'store_name'] = 'ашан'
merge_data.loc[merge_data['store_name'] == 'ашан гагаринский', 'store_name'] = 'ашан'
merge_data.loc[merge_data['store_name'] == 'ашан сокольники', 'store_name'] = 'ашан'
merge_data.loc[merge_data['store_name'] == 'ашан кунцево', 'store_name'] = 'ашан'
merge_data.loc[merge_data['store_name'] == 'ашан гагаринский ', 'store_name'] = 'ашан'

merge_data.loc[merge_data['store_name'] == 'глобус красногорск', 'store_name'] = 'глобус'
merge_data.loc[merge_data['store_name'] == 'метро стор 1017', 'store_name'] = 'metro cash&carry'
merge_data.loc[merge_data['store_name'] == 'метро стор 1356', 'store_name'] = 'metro cash&carry'
merge_data.loc[merge_data['store_name'] == 'метро стор 1014', 'store_name'] = 'metro cash&carry'
merge_data.loc[merge_data['store_name'] == 'метро стор 1073', 'store_name'] = 'metro cash&carry'
merge_data.loc[merge_data['store_name'] == 'метро стор 1019', 'store_name'] = 'metro cash&carry'
merge_data.loc[merge_data['store_name'] == 'метро стор 1061', 'store_name'] = 'metro cash&carry'
merge_data.loc[merge_data['store_name'] == 'метро стор 1077', 'store_name'] = 'metro cash&carry'
merge_data.loc[merge_data['store_name'] == 'метро стор 1048', 'store_name'] = 'metro cash&carry'
merge_data.loc[merge_data['store_name'] == 'метро стор 1050', 'store_name'] = 'metro cash&carry'
merge_data.loc[merge_data['store_name'] == 'метро стор 1318', 'store_name'] = 'metro cash&carry'
merge_data.loc[merge_data['store_name'] == 'метро стор 1322', 'store_name'] = 'metro cash&carry'
merge_data.loc[merge_data['store_name'] == 'метро стор 1011', 'store_name'] = 'metro cash&carry'
merge_data.loc[merge_data['store_name'] == 'метро стор 1049', 'store_name'] = 'metro cash&carry'
merge_data.loc[merge_data['store_name'] == 'метро стор 1012', 'store_name'] = 'metro cash&carry'
merge_data.loc[merge_data['store_name'] == 'метро стор 1018', 'store_name'] = 'metro cash&carry'
merge_data.loc[merge_data['store_name'] == 'метро стор 1049', 'store_name'] = 'metro cash&carry'

merge_data.loc[merge_data['store_name'] == 'ав экспресс меню', 'store_name'] = 'азбука вкуса'
merge_data.loc[merge_data['store_name'] == 'азбука вкуса daily', 'store_name'] = 'азбука вкуса'
```

In [30]:
```
array1 = merge_data.store_name.value_counts().index.tolist()
array2 = merge_data.store_name.value_counts().tolist()
array3 = []
for i, j in zip(array1, array2):
    couple = i + ' ' + str(j)
    array3.append(couple)
array3
```

Out[30]:
```
['Пятерочка 1173115',
 'Перекресток 977340',
 'Вкусвилл 712524',
 'Азбука Вкуса 457181',
 'Магнит 344221',
 'Магнолия 237137',
 'Дикси 212835',
 'Мясновъ 161927',
 'SPAR 139304',
 'Мираторг 124378',
 'Ашан 124240',
 'Fix Price 82901',
 'Окей 70796',
 'Metro Cash&Carry 56695',
 'Лента 48762',
 'Глобус Красногорск 43635',
 'Глобус 37021',
 'Виктория 35695',
 'Атак 33401',
 'Ярче! 30747',
```

Приложения: код

Разбиваем даты транзакции по сетам из 7 дней

In [55]:
```
from datetime import datetime
from datetime import timedelta

merge_data.purchdate = pd.to_datetime(merge_data.purchdate)

future = merge_data.purchdate[0] + timedelta(days=7)
week = []
j = 1

for i in range(0, 5636824):
    if merge_data.purchdate[i] < future:
        week.append(j)
    else:
        j = j + 1
        week.append(j)
        future = future + timedelta(days=7)

merge_data['week'] = week
merge_data.week.value_counts()
```

In [60]:
```
merge_data
```

Out[60]:

	amount	mcc	mrchcity	mrchname	cnum	sets	store_name
0	788.80	5411	MOSCOW	1LJXKW7J 1NKWJ1KK	PTMKRN	2	eurospar
1	621.14	5411	MOSCOW	T3LKT4HH M56B 5	2CDFHG	2	вкусвилл
2	452.76	5411	MOSCOW	T3LKT4HH M56B 5	BGYOGO	2	вкусвилл
3	458.00	5411	MOSCOW	T3LKT4HH M56B 5	2CDCFY	2	вкусвилл
4	196.00	5411	MOSCOW	T3LKT4HH M56B 5	0OYSXY	2	вкусвилл
...	...	...	...	...	...	...	...
5636819	2098.85	5411	MOSCOW	WO7E1JXIA37	ELEEKS	14	пятерочка
5636820	423.91	5411	MOSCOW	WO7E1JXIA37	EL4S4S	14	пятерочка
5636821	588.75	5411	MOSCOW	WO7E1JXIA37	KSNCSY	14	пятерочка
5636822	91.33	5411	MOSCOW	WO7E1JXIA37	ELOR8F	14	пятерочка
5636823	94.00	5411	MOSCOW	T3LKT4HH PQG M	ELEI5Z	14	вкусвилл

5636824 rows × 7 columns

Приложения: код

Пример разделения основного датасета на таргет-датасет с выделением признаков на две недели

```
clients = data.loc[data['store_name'] == 'лента']
clients = clients.drop(['purchdate', 'mcc'], axis=1)
clients = clients.reset_index(drop=True)
clients_final = pd.DataFrame()
clients_final['cnum'] = clients.cnum.value_counts().index

# 2
tmp = pd.DataFrame(clients.loc[clients['sets'] == 2]).groupby('cnum').agg(
    cl_ava_2 = ('amount', lambda x: sum(x) / len(x)),
    cl_freq_2 = ('amount', len)
)
clients_final = pd.merge(clients_final, tmp, on='cnum', how='left')
# 3
tmp = pd.DataFrame(clients.loc[clients['sets'] == 3]).groupby('cnum').agg(
    cl_ava_3 = ('amount', lambda x: sum(x) / len(x)),
    cl_freq_3 = ('amount', len)
)
clients_final = pd.merge(clients_final, tmp, on='cnum', how='left')

clients_final = clients_final.fillna(0)
```