Санкт-Петербургский государственный университет
Кафедра технологии программирования

Выпускная квалификационная работа бакалавра

Благов Артём Анатольевич

Оценка надежности клиентов финансовой организации по их профилям в социальных сетях

Направление Прикладная математика и информатика

Заведующий кафедрой
к.ф.-м.н., доцент Блеканов И.С.

Научный руководитель
старший преподаватель Малинина М.А.

Санкт-Петербург
2018г.

Содержание

1. Введение

В сфере микрокредитования решение о выдаче займа принимается на основании анкеты клиента и кредитной истории гражданина. Процессы ручного рассмотрения каждой заявки уже сейчас начинают уходить на второй план. До появления систем авто-скоринга андеррайтеры занимались обработкой всего трафика заявок и принимали свое субъективное решение. Большие компании стремятся заменить своих сотрудников алгоритмическими системами [1]. Уже сейчас большой процент заявок на кредит (если быть точнее, то 90%) в микрофинансовой организации, для которой компания Devim разрабатывает и внедряет алгоритмы оценки, обрабатываются автоматически, а остальные уходят на ручное рассмотрение андеррайтерам соответствующей мфо (микрофинансовая организация). И многие заявки возвращаются клиентам в статусе «Отказано», что негативно сказывается и на жизни гражданина, и на прибыли микрофинансовой организации. Предугадать кредитную историю можно попробовать из общедоступных данных о клиенте из социальных сетей. В социальных сетях пользователи легко делятся своей приватной и публичной информацией, которую можно добавлять к данным анкеты, не указанным клиентом. Так как при подаче заявки в микрофинансовую организацию клиент обязуется указывать свои паспортные данные, о чем сказано в договоре, то несовпадение этих данных с информацией из социальной сети может трактоваться как попытка мошенничества.

Пока что сервисы скоринга не обучены звонить клиентам, оценивать их эмоциональное и физическое состояние, задавать косвенные вопросы и нечетко анализировать данные и поведение пользователя в социальных сетях. Google показала на своем фестивале разработчиков Google I/O 2018 [2], что их Ассистент движется в этом направлении – понимании общего смысла устной речи [3]. Внедрение таких алгоритмов общения с клиентами вместо андеррайтеров очень рискованная и трудозатратная задача. В такой ситуации человек обладает преимуществом – он может позвонить и лично пообщаться с клиентом, подробнее спросить его про заявку. А для получения каких-то дополнительных данных они просматривают профили клиентов из социальных сетей, а также профили друзей и родственников. Автоматизация обхода профилей и их анализ облегчит монотонную работу андеррайтера и ускорит принятие решение, время на вынесение которого может быть ограничено политикой компании.

Бывают ситуации, когда в займе было отказано, так как клиент не имеет кредитной истории. Клиентами без кредитной истории могут быть молодые люди, которые в силу своего возраста не могли брать займы, и мигранты, кредитную историю которых затруднительно получить в силу языковых,

географических и технологических барьеров. Например, пока что сложно интегрироваться с бюро кредитных историй каждого государства, даже если оно имеет собственное API для таких целей. В таких случаях предсказывающие модели могут давать некорректные решения. Оценка кредитоспособности для таких клиентов зависит от конкретной системы, её архитектуры и встроенных алгоритмов. Так, заявки без кредитной истории могут отклоняться в 100% случаев или перенаправляться андеррайтерам. Анализ профиля социальной сети в некоторых случаях может помочь заполнить пробелы в анкете клиента для корректировки входящих данных в модель.

2. Постановка задачи

Сейчас многие финансовые организации стремятся получить как можно больше информации о клиенте. Открытым полем для извлечения дополнительной информации являются социальные сети, данные из которых организации собирают самостоятельно силами компании или обращаясь к data mining фирмам, которые за определенную плату продают проанализированный профиль гражданина. Принципы работы data mining компаний, собирающих все данные подряд, иногда достаточно агрессивные: пытаются обходить специальные ограничения социальных сетей, создают армии ботов, фейковые приложения, вводящие в заблуждение пользователей и требующих раскрыть свою приватную информацию. Последнее время в новостных заголовках все чаще появляются сообщения об утечках данных и нарушениях приватности [4]. Например, социальная сеть ВКонтакте недавно заявила о возможности сотрудничества с НБКИ [5], но уже в середине мая отвергла это предложение [6], объявнив его противоречащим принципам Вконтакте.

Задачей данной работы является автоматизация использования общедоступных данных из социальных сетей для оценки надежности клиентов микрофинансовой организации. Принципы, по которым собираются данные и выставляется оценка, должны быть достаточно абстрактными для дальнейшей возможности масштабирования на различные социальные сети и другие платформы, например мессенджеры. Например, на платформе Twitch.TV [7] пользователи обычно не используют свои реальные имена, но оставляют сообщения и комментируют трансляции своих любимых стримеров [8], что является общедоступными для сбора и анализа данными. В данной работе целевой платформой будет социальная сеть ВКонтакте, так как по информации из базы данных, которую предоставила Devim [9], из 450000 клиентов ссылку на свой профиль vk.com, Одноклассники, Instagram, Мой Круг и Facebook указали 31000, 14400, 600, 8200 и 4500 человек соответственно. А для выявления закономерностей и анализа требуются достаточные объемы данных.

Необходимо найти закономерности между данными профиля и целевыми переменными микрофинансовой организации, которые влияют на решение авто-скоринг системы или андеррайтера. Следующим этапом будет построение предсказывающих моделей на основании анализа. При получении положительного результата точности предсказания переменных методом кросс-валидации у модели на тестовом множестве нужно внедрить эту модель в сервис, а при отрицательном – попробовать выделить граничные случаи, которые оказывают влияние на решение андеррайтера, например несовпадение указанного в заявке и профиле пола гражданина. Также нужно встроить в сервис алгоритмические

проверки профиля, которые в данный момент делаются сотрудниками при рассмотрении заявки вручную, тем самым уменьшив время на принятие решения, автоматизировав монотонные действия и облегчив работу андеррайтеров.

3. Обзор литературы

Для начала работы с VK API и понимания основных принципов взаимодействия и ограничений хорошо подходит статья "VK API на Python"[10]. Из неё, а точнее её второй части [11] можно понять, что разработчики VK достаточно сильно ограничивают доступ к публичным данным пользователя, которые не указываются в самом профиле.

Первое погружение в специфику микросервисной архитектуры описывается в книге Сэма Ньюмена "Создание микросервисов"[12]. Основная идея в таком подходе – это уменьшение связности, которая вносит колоссальные проблемы при внедрении в систему новых сервисов, промежуточных звеньев и миграциях с одной технологии на другую. Также маленькие отдельные сервисы позволяют облегчить масштабирование и отказоустойчивость системы и, иногда, убирают узкие места [13], связанные с долго-работающей бизнес-логикой сервисов.

Кроме написания domain logic [14] для развертывания на производственное окружение, к сервису предъявляются дополнительные требования, которые формируют production-ready микросервис. Эти требования описывается в книге Susan J. Fowler "Production-Ready Microservices: Building Standardized Systems Across an Engineering Organization"[15] и диктуются бизнес и техническими требованиями проекта.

4. Сбор данных

4.1. База данных мфо

Первым этапом в анализе данных является сбор самих данных. Из базы данных мфо будут собираться значения целевых переменных, которые оказывают влияние на бизнес составляющую. В базе данных многие из указанных переменных не содержатся в явном виде и вычисляются косвенно. Ниже указан список переменных, требующихся для анализа и которые записаны в объекты `Client`.

- "*vk_url*": ссылка на социальную сеть ВКонтакте вида `vk.com/id` или `vkontakte.ru/id`.
- "*is_approved*": решение об одобрении системы авто-скоринга или андеррайтера. 0 - отказано, 1 одобрено.
- "*is_payed*": вернул ли клиент займ. 0 - не вернул, 1 - вернул.
- "*cost_all*": стоимость кредита в рублях, которую обязался выплатить заемщик на момент подписания договора потребительского микрозайма.
- "*really_payed*": реальная стоимость кредита, которую заплатил клиент для закрытия договора. Так, если клиент просрочил выплаты по займу, то итоговая сумма будет больше "Cost all а если вернул раньше срока – меньше.
- "*is_pay_by_order* "*pay_by_order_delta* "*pay_by_order_percent*": Факт, разность и относительный процент того, что клиент заплатил по договору соответственно. Для факта сравниваются значения *cost_all* и *really_payed*, для разницы – из *really_payed* отнимается значение *cost_all*, а для *pay_by_order_percent*: $\frac{really\_payed}{cost\_all}$
- "*is_prolongation* "*prolongation_amount* "*prolongation_days*": Факт пролонгации, количество пролонгаций и суммарное количество дней пролонгации соответственно.
- "*is_delayed* "*delayed_days* "*all_delayed_days*": Факт задержки выплат, официально просроченные дни и всего просроченных дней. *is_delayed* равен единице, если значение *delayed_days* больше нуля. Точно так же как и

all_delayed_days, *delayed_days* увеличивается на единицу за каждый просроченный день, но сбрасывается до нуля если клиент взял пролонгацию договора.

Для клиентов, которым было отказано в займе, значения остальных целевых переменных не определены и поэтому такие клиенты не будут учитываться в построении, обучении и валидации моделей предсказания переменных, кроме целевой переменной `is_approved`. Всего было собрано 4500 уникальных пользователей с различными значениями целевых переменных. В выборку попали и клиенты, которым отказали в кредите, и которые вернули досрочно, день в день по договору или просрочили выплаты и были отданы коллекторам. Большинству клиентов было отказано, что усилило бы вероятность переобучения модели, поэтому с признаком $is\_approved = True$ было выбрано 1500 клиентов, а остальные не были включены в итоговую выборку.

4.2. Социальная сеть ВКонтакте

Социальная сеть ВКонтакте предоставляет API [16] для получения практически всех общедоступных данных (которые может получить любой зарегистрированный пользователь через web-версию vk.com), но и вводит искуственные барьеры. Во первых, для приложений введены различные виды токенов доступа [17], что уже разграничивает способы получения информации. Во вторых есть ограничения на вызовы всего API и отдельных методов [18], что напрямую связано с объемом трафика заявок, которые можно обработать. Итак, из вконтакте для собранных из базы данных микрофинансовой организации клиентов были собраны такие поля и записаны в объекты `User`:

- *vk_url*: поле `domain` объекта Пользователь [19].
- *vk_id*: поле `id` объекта Пользователь.
- $first\_name$: поле `first_name` объекта Пользователь.
- *last_name*: поле `last_name` объекта Пользователь.
- *sex*: поле sex объекта Пользователь.
- *hidden*: скрыл ли пользователь свой профиль.
- *deactivated*: является ли профиль заблокированным или удаленным.

- *counter_audios*: число аудиозаписей, которые пользователь добавил в раздел "мои аудиозаписи".
- *counter_photos*: число фотографий пользователя.
- *counter_about_lists*: число элементов в описательных списках. Описательными списками являются поля `movies`, `music`, `tv`, `books`, `games`, `interests`.
- *counter_friends*: число друзей, которых не скрыл пользователь.
- *counter_feeders*: число источников информации, которые появляются в ленте новостей пользователя, кроме друзей и рекламных объявлений. Вычисляется как сумма числа групп, числа подписок и числа интересных страниц.
- *counter_wall_posts*: число постов на стене пользователя, в том числе и репостов.
- *counter_shared*: число репостов на стене пользователя.
- *is_in_relations*: состоит ли пользователь в отношениях, истина для значений 2, 3, 4, 8 возвращаемого поля `relation` метода *users.get*.
- *counter_share_contacts*: количество контактов, которыми поделился пользователь. Под контактами подразумевается поля объекта `user`, которые могут быть интерпретированы как средства связи с пользователем – номер телефона, ссылка на личный вебсайт и другие.
- *is_share_contacts*: факт того, что пользователь поделился своими контактными данными.
- *is_wall_open*: содержит ли стена пользователя записи других пользователей.
- *counter_career_all*: количество смен мест занятости, а именно суммарное количество элементов из полей школы, университеты, места работы и места службы.
- *counter_career*: количество смен мест работы.
- *is_career*: указано ли что-либо в полях школы, университеты, места работы и места службы.
- *personal_political*: поле `personal.political` объекта Пользователь.

- *personal_people_main*: поле `personal.people_main` объекта Пользователь.
- *personal_life_main*: поле `personal.life_main` объекта Пользователь.
- *personal_smoking*: поле `personal.smoking` объекта Пользователь.
- *personal_alcohol*: поле `personal.political` объекта Пользователь.
- *counter_langs*: количество указанных языков, которыми пользователь владеет.
- *is_military*: указаны ли места службы пользователя.
- *friends_ids*: список id друзей.
- *groups_ids*: список id групп, сообществ, подписок.
- *langs*: языки, которыми владеет пользователь.
- *describes*: список выражений из полей `about` и `quotes`.

Отдельно хочется отметить, что данные, которые собиралась из социальных сетей не содержат таких данных о пользователе, как например возраст, которые оказывают огромное влияние на целевую переменную [35], так как эти данные уже указываются в анкете при подаче заявки на займ и авто-скоринг системы умеют с ними работать.

5. Анализ данных

Анализ является одним из этапов реализации систем интеллектуальной обработки данных по методологии CRISP-DM [22], которая помогает формализовывать бизнес задачи.

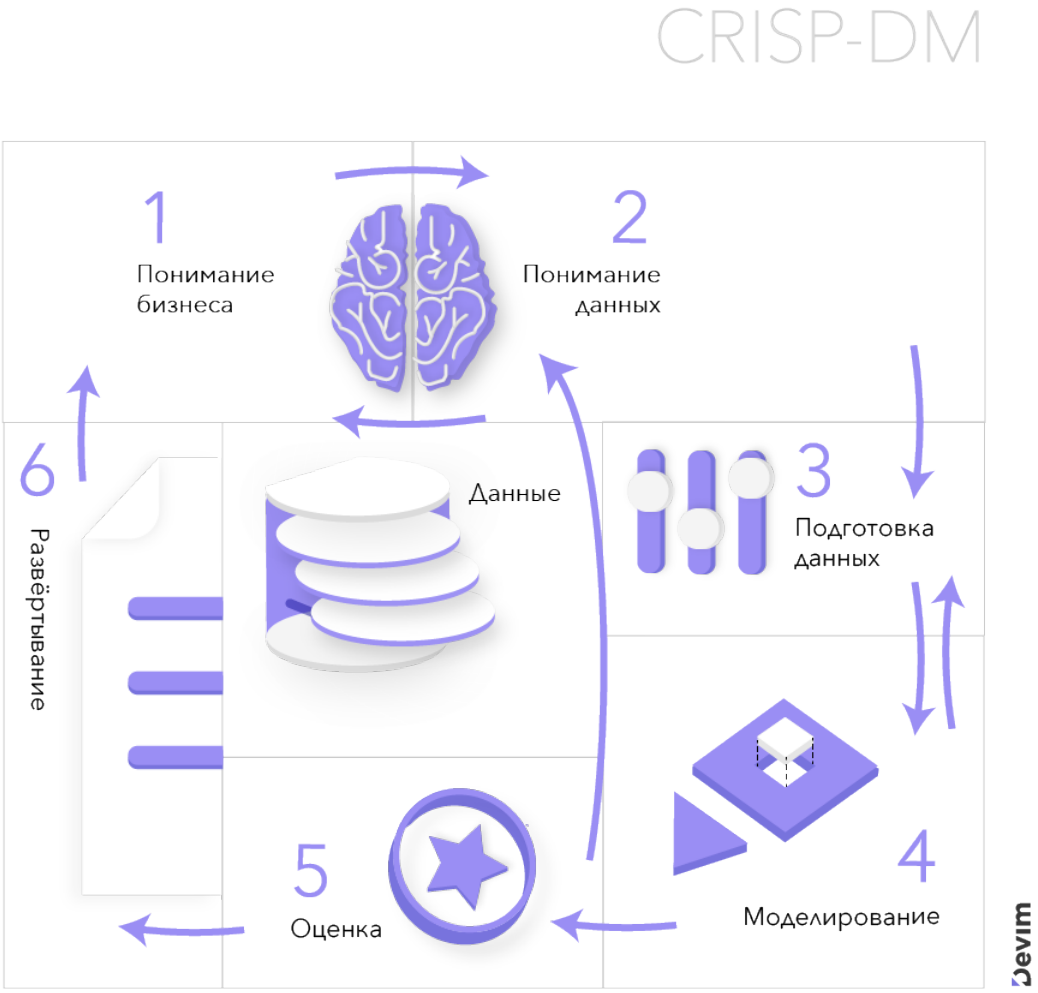

Рис. 1. CRISP-DM

Признаки из базы данных мфо и vk были объединены в одну таблицу по правилу совпадения url страницы ($User.vk\_url == Client.vk\_url$). Выбросы в значениях были убраны по правилу трех сигм в предположении, что все признаки нормально распределены (a_i, σ_i^2). После была построена таблица корреляции Пирсона при помощи функции $corr()$ из python библиотеки pandas и визуализирована тепловой картой корреляции, представленной на рис. 2. Как можно заменить, значения на главной диагонали соответствуют значению 1.0, так как в данных ячейках находится значение корреляции признака самим с собой. В матрице выделяются две подматрицы на главной диагонали, которые соответствуют таблицам корреляции признаков отдельно из базы данных мфо и базы данных вконтакте соответственно. Но в данном случае интересными для исследования являются подматрицы A_1 и A_2 с координатами $(11; 0) : (32; 10)$ и $(0; 11) : (10; 32)$, причем $A = A_1 = A_2^T$.

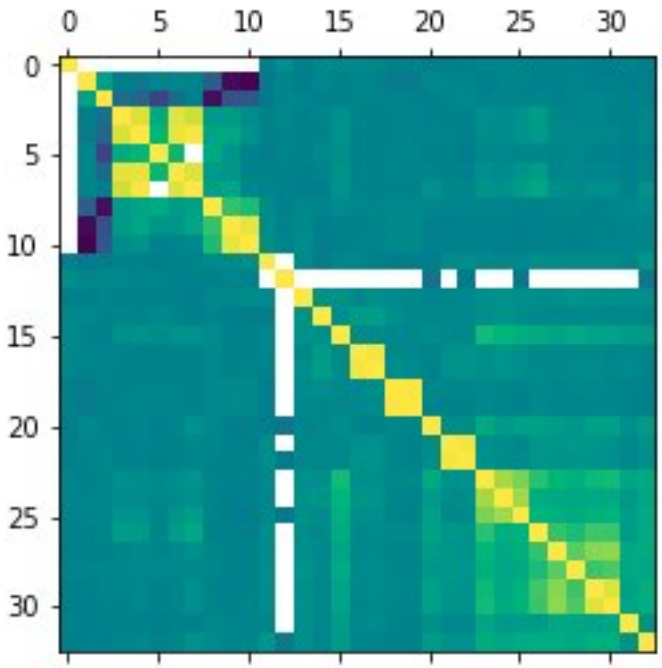

Рис. 2. Корреляция всех признаков

Цвет ячеек задается линейно в диапозоне от темно-синего до желтого цветов, которые соответствуют значениям корреляции -0.08 и 1.0. Белым цветом обозначены ячейки для которых невозможно вычислить корреляцию (NaN).

Если рассмотреть матрицу A ближе (рис. 3), то можно заметить, что все значения в ячейках не более 0.3 и не менее -0.08. Значение 0.3 достигается в ячейке $(15; 7)$, которая соответствует корреляции `User.personal_political` и `Client.prolongation_days`. Так как `User.personal_political` является перечислительным признаком, то такую корреляцию не имеет смысла рассматривать, так как, скорее всего, это является аномалией в данных. Значения корреляции остальных признаков находятся около нуля и интерпретируются как независимость исследуемых данных.

Для обучения модели предсказания были взяты несколько методов машинного обучения классификации – метод опорных векторов [24], метод стохастического градиентного спуска [25] и реализация многослойной нейронной сети на основе персептрона [26]. Перед обучением пропущенные ячейки были заполнены наиболее частыми значениями (most frequent), непрерывные признаки нормализированы и выбраны лучшие признаки по принципу линейной независимости и уменьшения ранга матрицы корреляции, реализованными в библиотеке sklearn [23] методами.

Наилучшие параметры для каждого шага были подобраны алгоритмом grid-поиска по указанной таблице для получения наилучшей точности классификации, а также метод кросс-валидации для проверки точности обученной модели на разных частях данных. Результаты кросс валидации показали, что общая точность предсказания для любой целевой переменной не более 0.65, причем точность модели в случае бинарного классификатора 1.0 интерпретируется как полное совпадение, 0.0 – как всё не верно, а 0.5 эквивалентно подбрасыванию монетки, то есть использованию случайного классификатора. Без дальнейшего анализа результатов предсказания на тестовом множестве можно сделать вывод, что модель классификации не подходит для использования и требуется изменить набор собираемых данных из социальной сети.

6. Архитектура текущей системы

В этой главе, в отличии от предыдущих, слово "клиент"будет подразумевать сущность из модели общения Server-Client [27], если контекст предложения не говорит обратного.

Вся текущая система (рис. 4) построена на принципах микросервисной архитектуры – каждая бизнес сущность выделяется в отдельный микросервис со своим API (application program interface). Основные бизнес процессы сосредоточены в Core, имеющий специальные точки входа (Gateways) для конкретных клиентов, и сервисы которого закрыты для прямого доступа. Поток заявок идет напрямую в Core, где, по модели общения *Publisher – Subscriber* (далее Pub/Sub) [28] через Events доставляются заявки клиентам. Компонент Analytics занимается скорингом и перенаправляет неоднозначные случаи в компонент Underwriters. Компонент Collectors является системой работы коллекторов. Благодаря модели Pub/Sub и микросервисному подходу, всю архитектуру системы можно описать достаточно абстрактно, не раскрывая конкретных реализаций отдельных компонентов. Кроме того, такой подход позволяет прозрачно менять реализации компонентов или встраивать готовые решения, не изменяя и не нарушая работоспособность других частей и проводить Blue-Green Deployment [29], который существенно уменьшает риски связанные с переходом. Поставленная задача требует написания и внедрения такого микросервиса, реализация которого в контексте данной системы не увеличивала связности и служила вспомогательным инструментом при принятии решения, а при его отказе остальная система могла бы продолжать свою работу.

7. Сервис

Для достижения требуемых целей было решено использовать две внутренних сущности: и . Сущность будет содержать информацию о пользователе, собранную из социальной сети, дату последенй обработки и ссылки на других пользователей - друзей и родственников, а также ссылку на уникальный идентификатор клиента из компонента Core. Id пользователя ВКонтакте является уникальным, поэтому было принято решение создавать сущности для всех друзей и родственников, но не заполнять их. один к одному связана с заявкой из Core и содержит дату создания, ссылку на Пользователя, флаг завершенности обработки и список Assertions. Каждый Assertion это небольшой объект, содержащий в себе вид утверждения и комментарий, описывающий результат анализа профиля пользователя.

Архитектура самого сервиса разбита на слои, каждый из которых предназначен для решения конкретной задачи разработчика.

7.1. Транспортный слой

Транспортный слой сервиса обособляет код, предназначенный для взаимодействия микросервиса и внешних сервисов при помощи транспортных протоколов сети. Также он содержит код, использующий достоинства конкретных протоколов и различные оптимизации передачи данных, например сжатие.

Для Server-Client модели общения используется протокол gRPC [30], у которого пропускная способность больше, чем у классического для интернета протокола HTTP. Кроме того, протокол gRPC обладает большим количеством полезных подключаемых расширений от самих разработчиков и сообщества [31]. В том числе и подключаемый HTTP прокси [32], который перенаправляет HTTP запросы на gRPC точки точки доступа.

Для общения с внешними сервисами микросервис предоставляет один API метод Get, предназначенный для пользовательских интерфейсов для отображения результатов и статуса обработки профиля. Второй точкой входа будет событие направления заявки системой авто-скоринга на ручное рассмотрение андеррайтерам в Core, при котором данный сервис будет выступать Subscriber, а Core – Publisher.

7.2. Бизнес логика

Бизнес логика содержит все процессы микросервиса, для реализации которых он был создан – алгоритмы взаимодействия бизнес-сущностей по требу-

емым правилам. Слой бизнес логики помещает разработчика в рамки domain моделей, которые не должны содержать частей из транспортного слоя или базы данных, что позволяет избежать смешения понятий и идей из разных слоев архитектуры.

На данный момент сервис может предоставить 4 вида сообщений:

- Положительно (good) – проанализированная информация содержит значения, которые оказывают положительное влияние на пользователя социальной сети как клиента мфо. Например, верифицированная учетная запись (поле $verified$ из ответа на запрос https://api.vk.com/method/users.get) будет обозначать, что пользователь дорожит своим аккаунтом и информацией на нём.

- Нейтрально (neutral) – обозначает какие-то факты о пользователе, которые нельзя интерпретировать однозначно и должны трактоваться в каждом конкретном случае отдельно.

- Уведомление (warning) – профиль содержит косвенные признаки недобросовестности гражданина.

- Предупреждение (critical) – указанная в профиле информация может указывать на ненадежность клиента и, с большой вероятностью, профиль не принадлежит клиенту. Например, если пол пользователя не совпадает с указанным в анкете.

В Таблице 1 приведен список алгоритмических проверок, основанных на эвристических предположениях автора, которые могут быть расширены или отключены в любой момент, а значения параметров изменены со временем или после сбора статистических данных с пометками экспертов.

Под понятием «сильно различаются» подразумевается значение расстояния Жаро-Винклера [20], которое позволяет оценить схожесть слов и показывает хорошие результаты для имен собственных. Перед вычислением расстояния заглавные буквы слова меняются на прописные, “ъ” заменяется на “ь”, а “ё” – на “е” и производится попытка замены уменьшительно-ласкательного варианта имени на его полный аналог через специально подготовленный словарь частовстречающихся соответствий, составленный вручную на основании публичной базы данных зарегестрированных имен и фамилий в популярных социальных сетях [21]. Предварительно имена не ставятся в именительный падеж так как в анкете указываются паспортные данные, которые пишутся в именительном падеже, а ВКонтакте позволяет указать падеж, в котором требуется вернуть имя и фамилию пользователя.

Таблица 1. Список проверок и соответствующих Assertions

Описание проблемы	Вид сообщения
Профиль пользователя заблокирован или удален	critical
Профиль пользователя скрыт	neutral
Дата рождения в профиле и заявке разные	warning
Имена пользователя и клиента сильно* различаются	warning
Фамилии пользователя и клиента сильно* различаются	warning
Профиль пользователя верифицирован	good
Пользователь и клиент разных полов	critical
Профиль был активен не более, чем 3 дня назад	good
Профиль был активен не более, чем неделю назад	neutral
Профиль был активен не более, чем месяц назад	warning
Профиль последний раз был в сети более месяца назад	critical
Количество друзей, которым займ был одобрен, больше тех, кому отказали	good
Количество друзей, которым займ был одобрен, меньше тех, кому отказали	warning
Количество родственников, которым займ был одобрен, больше тех, кому отказали	good
Количество родственников, которым займ был одобрен, меньше тех, кому отказали	warning

7.3. Middlewares

Middlewares или промежуточные слои, это участки кода, отвечающие за промежуточные действия, производимые в рамках запроса. Middlewares могут знать о специфике domain или транспортного слоя, но в тоже время быть максимально независимы. Работоспособность самого сервиса также не должна зависеть от промежуточных слоев, так как они не должны менять поток данных. В случае, если промежуточный слой меняет поток данных, это должно производиться прозрачно для разработчика. Например, слой JWT [33] авторизации является подключаемым звеном цепи запроса, который не просто меняет, а принимает решение о доступе или отказе. Это является изменением потока данных, которое очевидно для разработчиков, и поэтому может использоваться как middleware.

Список middlewares, которые разработчики подключают к своим сервисам, определяет разницу между production-ready и сырым микросервисом и чаще всего определяется бизнес требованиями и ведущими разработчиками или

техническим директором каждой конкретной компании. Сервис реализует следующие промежуточные слои.

- Слой логирования
  Позволяет отслеживать входящие и выходящие данные, а также сообщения об ошибках на стороне сервера перед отправкой клиенту.

- Слой валидации
  Проверяет входящие данные на корректность с точки зрения domain моделей и отказывает в доступе при их неправильности. Например, день месяца не может быть больше тридцати одного. И в случае поступления таких данных на сервис слой валидации должен отклонить запрос. В реализации данного сервиса происходит валидирование указанного в анкете URL на принадлежность социальной сети ВКонтакте.

- Слой трассировки
  Аналогично определению из геометрической оптики, трассировкой сервиса [34] является подход к изучению и отслеживанию направлений движения запросов в распределенной информационной системе. У системы есть точка начала запроса и промежуточные звенья – сервисы. В точке появления запроса создается корневой span, метаданные которого передаются от сервиса к сервису, каждый из которых проставляет отметку о прохождении чекпоинта и прикрепляет свой span к предыдущему. Таким образом получается цепь из span-ов, каждый из которых содержит информацию о времени и месте создания. Вся эта информация отправляется в агрегатор трассировок распределенной системы. При помощи такой базы данных трассеров можно отследить время выполнения запроса для каждого звена из цепи микросервисов, направления движения запросов и визуализировать архитектуру всей системы в виде карты сервисов при условии возможности подключения механизмов трассировки к сервисам.

- Слой метрик
  В контексте информационной системы, метрики – это некие абстрактные числовые значения, характеризующие протекающие процессы. Например, наиболее характерной для интернета и часто используемой метрикой является время выполнения запроса. Слой метрик позволяет собирать статистические данные системы и проводить мониторинг различных показателей, например для уведомления системных администраторов датацентров при значительном увеличении среднего времени ответа. Реализованный сервис собирает такие метрики, как среднее время обработки

запроса, количество запросов в минуту и количество работающих потоков системы.

Описанные слои не являются обязательными для реализации и могут варьироваться не только в пределах компании, но и в пределах одного приложения. Так, методы, являющиеся частью процесса авторизации пользователя в веб приложении, могут не содержать проверкок доступа.

8. Интеграция

В силу особенностей построения микросервисной архитектуры и необязательности проверки профилей пользователей, а также возможности гибкого расширения функциональности на другие социальные сети было предложено внедрить сервис в подсистему Underwriters.

Сервис отслеживает статус заявки и, если заявка отправляется на ручное рассмотрение андеррайтерам, запускает анализ профиля социальной сети. При данном подходе сервис не зависит от частей группы Underwriters, а сервисы Underwriters ничего не знают о нем, что уменьшает связность системы и добавляет отказоустойчивости. По аналогичному принципу могут быть внедрены сервисы обработки других социальных сетей (рис. 5).

Для использования полученной информации на веб интерфейсе андеррайтерами будет добавлен раздел "Социальные сети"(рис. 6) в котором будут колонка со списком социальных сетей, колонка со ссылками для прямого доступа к профилю, в случае, если андеррайтер хочет самостоятельно посмотреть профиль, и колонка со списком Assertions. При наведении курсора на отдельный пункт отчета будет появляться полная информация предупреждения, а так же дополнительная информация или ссылки, если такие подразумеваются контекстом Assertion. В дополнении к комментарию и окраске, будут отображаться кнопки оценки правильности Assertion с точки зрения сотрудника для сбора статистических данных и корректировки алгоритмов анализа.

При переходе на указанную вкладку, браузер отправляет запрос напрямую в сервис или через специальный gateway-агрегатор, собирающий информацию со всех микросерисов-анализаторов социальных сетей, в ответе на который система возвращает описанные выше отчеты. В условиях данной архитектуры допускается задержка в обработке информации из социальной сети, поэтому для получения данных могут потребоваться дополнительные запросы.

Такая организация позволит достаточно прозрачно добавлять новые анализы различных социальных сетей, не прикладывая больших усилий при изменении бизнес приоритетов, популярности и доступности внешних сервисов.

9. Заключение

В данной работе была поставлена задача использования публичных данных пользователей социальных сетей в задаче оценки надежности клиентов микрофинансовой организации. Был разработан production-ready микросервис, готовый к внедрению в текущую систему, а также прототип изменений дизайна веб интерфейса информационной подсистемы андеррайтинга. В свою очередь, сервис реализует алгоритмические проверки пользователя и сообщает сотруднику о потенциальном случае мошенничества.

Разработанные методы анализа и проверки клиента в основном направлены на ускорение взаимодействия андеррайтеров с указанными в заявке данными пользователя и определение случаев мошенничества, что ускоряет принятие решений.

Предварительный анализ числовых данных, собранных из социальной сети ВКонтакте по базе данных микрофинансовой организации не оправдал ожидаемых и приемлемых результатов точности предсказания целевых переменных и требует дальнейших исследований взаимосвязей поведения пользователя в интернете и его кредитного рейтинга. Под поведением подразумеваются оставляемые комментарии, их эмоциональная окраска, частота и тематичность.

Алгоритмические проверки тоже могут быть улучшены и дополнены в различных направлениях, таких как автоматическое определение ложных (фейковых) страниц, анализ интересов пользователя через подписки, лайки и группы или предложение пройти опрос через мессенджеры или на точке продаж для оценки психологического состояния клиента на момент рассмотрения заявки. Анализ друзей и социальных графов пользователя также может дать положительные результаты в предсказании финансового положения клиента.

Использование данных из большого количества различных социальных сетей является дополнительным неисследованным информационным полем, что поможет построить финансовый профиль каждого человека.

Ни для кого не секрет, что описанные выше направления исследований и разработки требуют колоссальных трудозатрат от специалистов совершенно различных направлений деятельности. Для анализа поведения человека в интернете и его закономерностей требуется большое количество исследований в этом направлении, для проведения которых требуются большие объемы данных. А для подготовки и анализа такого объема данных требуются вычислительные мощности

Список литературы

[1] Роботы вместо лучших сотрудников: машинное обучение по ответам экспертов / Блог компании Devim / Хабр [Электронный ресурс] // habr. URL:https://habr.com/company/devimteam/blog/348092/ (дата обращения: 17.05.18).

[2] Google I/O 2018 [Электронный ресурс] URL:https://events.google.com/io/ (дата обращения: 17.05.18).

[3] Google IO 2018: all of the highlights and news from the keynote [Электронный ресурс] // techradar. URL:https://www.techradar.com/news/google-io-2018 (дата обращения: 17.05.18).

[4] How Trump Consultants Exploited the Facebook Data of Millions [Электронный ресурс] // The New York Times. URL:https://www.nytimes.com/2018/03/17/us/politics/cambridge-analytica-trump-campaign.html (дата обращения: 17.05.18).

[5] Как «ВКонтакте» разрешила анализировать данные пользователей [Электронный ресурс] // РБК. URL:https://www.rbc.ru/technology_and_media/02/04/2018/5abe534d9a7947350e3a7dfa (дата обращения: 17.05.18).

[6] «ВКонтакте» не будет сотрудничать с Национальным бюро кредитных историй [Электронный ресурс] // Коммерсант.ru. URL:https://www.kommersant.ru/doc/3630050 (дата обращения: 17.05.18).

[7] Twitch.TV [Электронный ресурс] URL:https://www.twitch.tv/ (дата обращения: 17.05.18).

[8] Стримеры — о том, как заработать на трансляциях компьютерных игр [Электронный ресурс] // The Village. URL:http://www.the-village.ru/village/people/people/233345-go-stream (дата обращения: 17.05.18).

[9] Devim URL:https://devim.com/ (дата обращения: 17.05.18).

[10] VK API на Python: часть 1, выгружаем все фото из альбома [Электронный ресурс] // proglib.io URL:https://proglib.io/p/python-vk-api/ (дата обращения: 17.05.18).

[11] VK API на Python: часть 2, узнаем, что лайкал пользователь [Электронный ресурс] // proglib.io URL:https://proglib.io/p/python-vk-api-2/ (дата обращения: 17.05.18).

[12] Sam Newman Building Microservices // O'REILLY.

[13] bottleneck Определение в кембриджском словаре английского языка [Электронный ресурс] URL:https://dictionary.cambridge.org/ru/\T2A\cyrs\T2A\cyrl\T2A\cyro\T2A\cyrv\T2A\cyra\T2A\cyrr\T2A\cyrsftsn/\T2A\cyra\T2A\cyrn\T2A\cyrg\T2A\cyrl\T2A\cyri\T2A\cyrishrt\T2A\cyrs\T2A\cyrk\T2A\cyri\T2A\cyrishrt/bottleneck (дата обращения: 17.05.18).

[14] What is domain logic? [Электронный ресурс] // Enterprise Craftsmanship. URL:https://enterprisecraftsmanship.com/2016/08/25/what-is-domain-logic/ (дата обращения: 17.05.18).

[15] Susan J. Fowler Production-Ready Microservices: Building Standardized Systems Across an Engineering Organization // O'REILLY URL:https://www.goodreads.com/book/show/33252815-production-ready-microservices

[16] API methods | Developers | VK [Электронный ресурс] URL:https://vk.com/dev/methods (дата обращения: 17.05.18).

[17] Getting a Token | Developers | VK [Электронный ресурс] URL:https://vk.com/dev/access_token (дата обращения: 17.05.18).

[18] Streaming API and Restrictions on Search API | Developers | VK [Электронный ресурс] URL:https://vk.com/dev/data_limits (дата обращения: 17.05.18).

[19] User | Developers | VK [Электронный ресурс] URL:https://vk.com/dev/objects/user (дата обращения: 17.05.18).

[20] William W. Cohen Pradeep Ravikumar Stephen E. Fienberg A Comparison of String Metrics for Matching Names and Records [Электронный ресурс] URL:https://www.cs.cmu.edu/afs/cs/Web/People/wcohen/postscript/kdd-2003-match-ws.pdf (дата обращения: 17.05.18).

[21] База данных имён и фамилий [Электронный ресурс] URL:https://mydata.biz/ru/catalog/databases/names_db (дата обращения: 17.05.18).

[22] The CRISP-DM model: the new blueprint for data mining, Journal of Data Warehousing, Vol. 5, pp. 13-22.

[23] scikit-learn: Machine Learning in Python [Электронный ресурс] URL:http://scikit-learn.org (дата обращения: 17.05.18).

[24] sklearn.svm.SVC [Электронный ресурс] URL:http://scikit-learn.org/stable/modules/classes.html#module-sklearn.svm (дата обращения: 17.05.18).

[25] sklearn.linear_model.SGDClassifier [Электронный ресурс] URL:http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html#sklearn-linear-model-sgdclassifier (дата обращения: 17.05.18).

[26] sklearn.neural_network.MLPClassifier [Электронный ресурс] URL:http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier (дата обращения: 17.05.18).

[27] What is a Client? What is a Server? And What is a Host? [Электронный ресурс] // LearnTomato. URL:https://learntomato.com/what-is-a-client-what-is-a-server-what-is-a-host/ (дата обращения: 17.05.18).

[28] Publish/Subscribe | Microsoft Docs [Электронный ресурс] URL:https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff649664(v=pandp.10) (дата обращения: 17.05.18).

[29] Using Blue-Green Deployment to Reduce Downtime and Risk [Электронный ресурс] // Cloud Foundry Documentation. URL:https://docs.cloudfoundry.org/devguide/deploy-apps/blue-green.html (дата обращения: 17.05.18).

[30] grpc [Электронный ресурс] URL:https://grpc.io/ (дата обращения: 17.05.18).

[31] gRPC Ecosystem [Электронный ресурс] // github.com URL:https://github.com/grpc-ecosystem (дата обращения: 17.05.18).

[32] grpc-gateway [Электронный ресурс] // github.com URL:https://github.com/grpc-ecosystem/grpc-gateway (дата обращения: 17.05.18).

[33] JSON Web Tokens [Электронный ресурс] URL:https://jwt.io/ (дата обращения: 17.05.18).

[34] opentracing.io [Электронный ресурс] URL:http://opentracing.io/ (дата обращения: 17.05.18).

[35] John M. Chapman Commercial Banks and Consumer Instalment Credit [Электронный ресурс] URL:http://www.nber.org/chapters/c4732.pdf (дата обращения: 17.05.18). Chapter 5. P 119

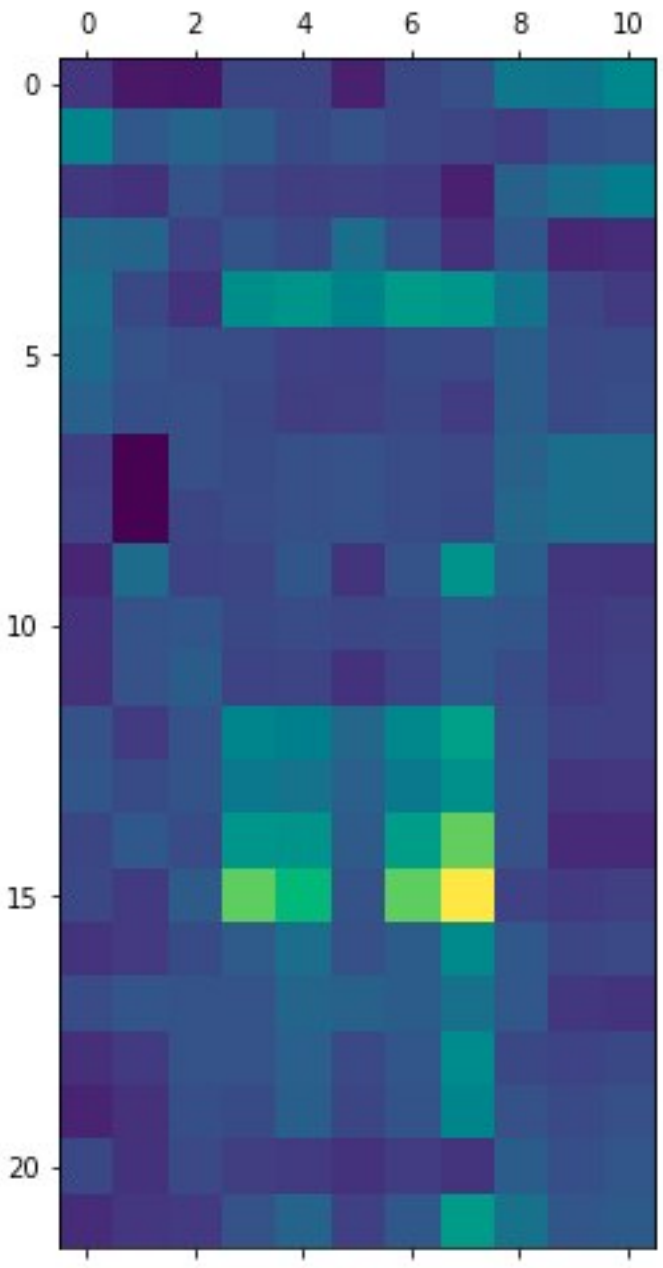

Рис. 3. Корреляция признаков клиента и пользователя

Цвет ячеек задается линейно в диапозоне от темно-синего до желтого цветов, которые соответствуют значениям корреляции -0.08 и 0.3. Белым цветом обозначены ячейки для которых невозможно вычислить корреляцию (NaN).

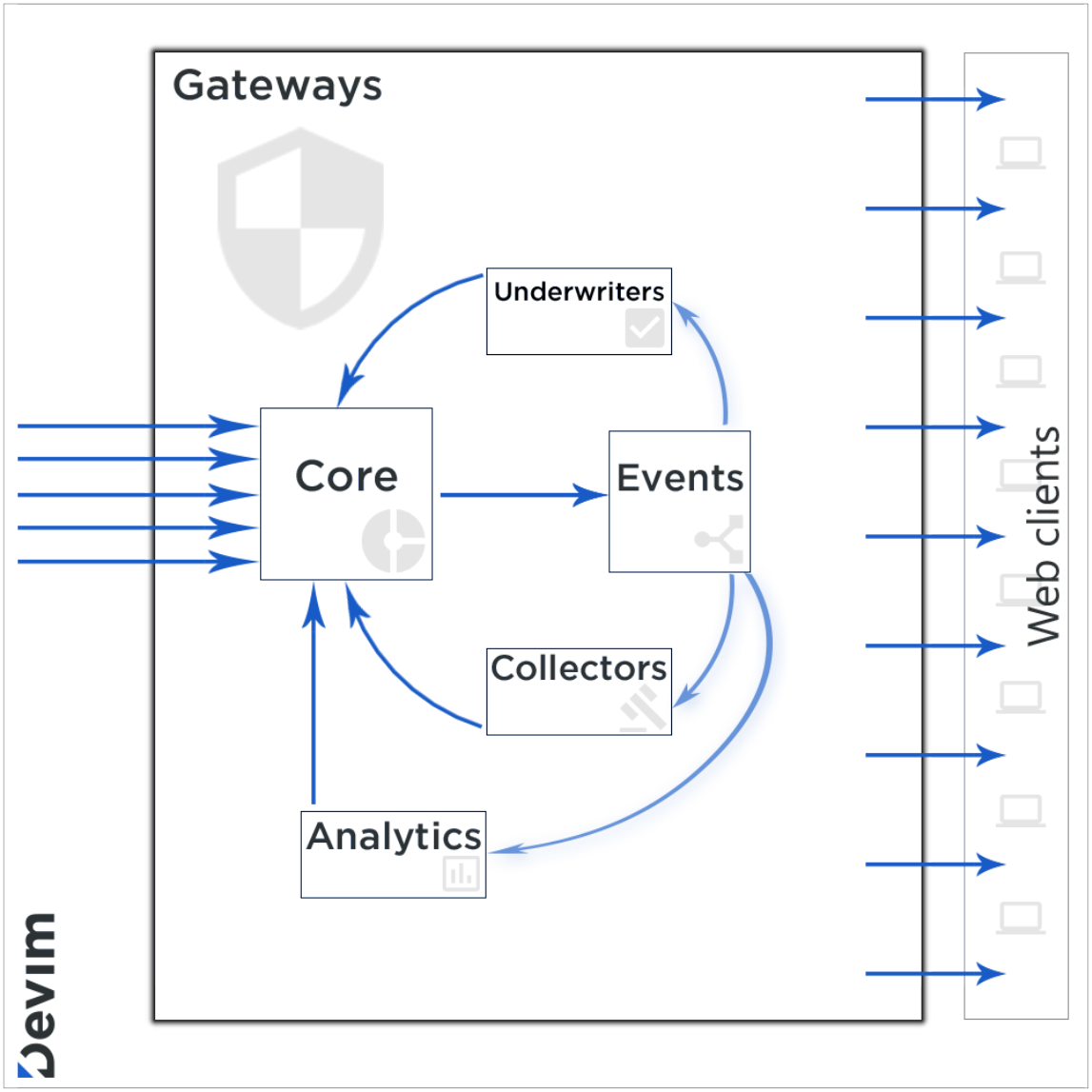

Рис. 4. Архитектура системы

Квадратами обозначены абстрактные границы компонентов или участников системы. Темно-синие стрелки обозначают направления поступления и изменения данных, а свето-синие указывают на взаимодействие компонентов по модели *Publisher/Subscriber*. Данное изображение не отражает полной системы, но помогает визуализировать ключевые в данном контексте подходы.

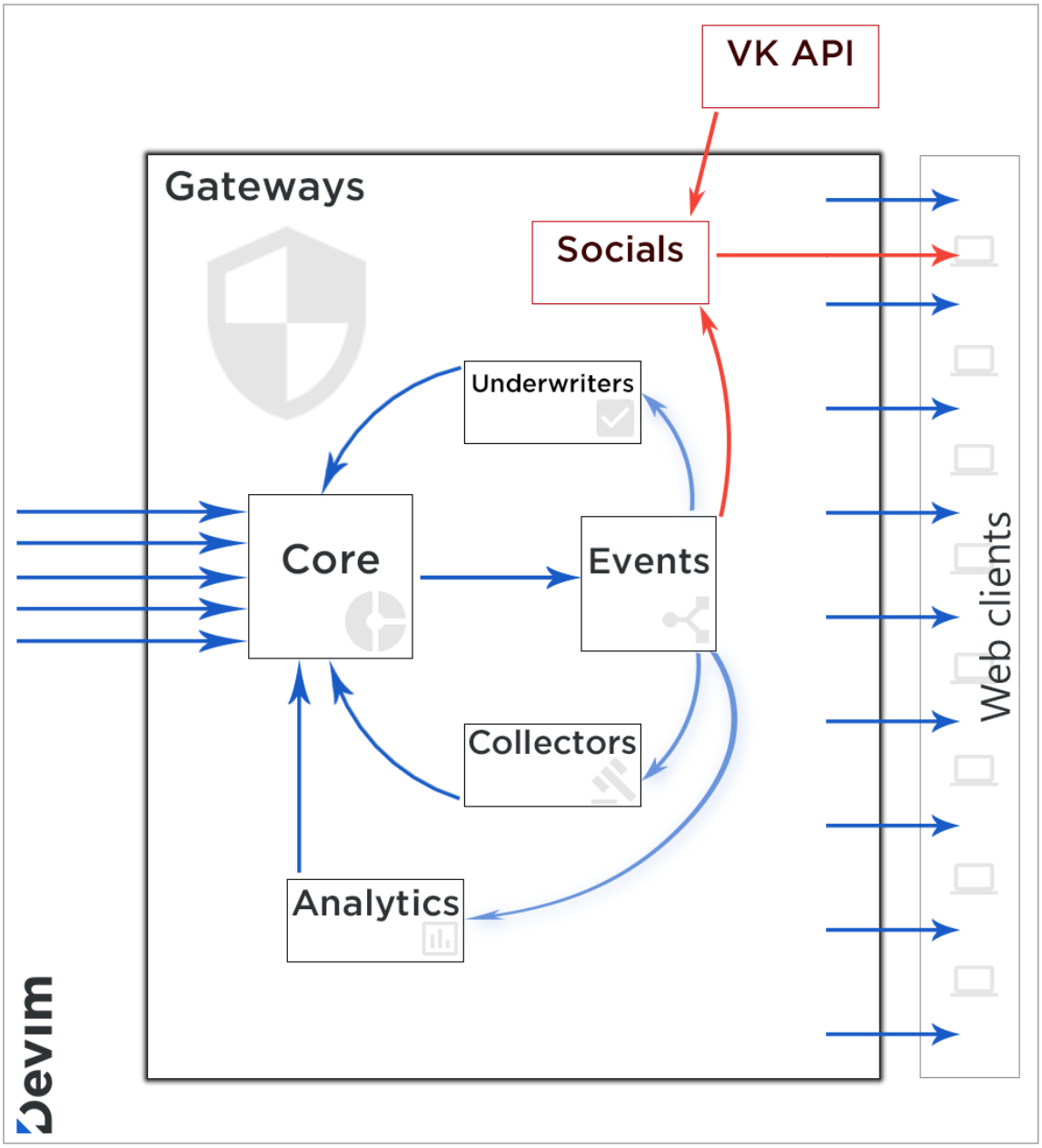

Рис. 5. Внедрение нового сериса в архитектуру системы

Красным обозначены новые компоненты и направления движения данных. Описание остальных частей можно узнать из рис. 4.

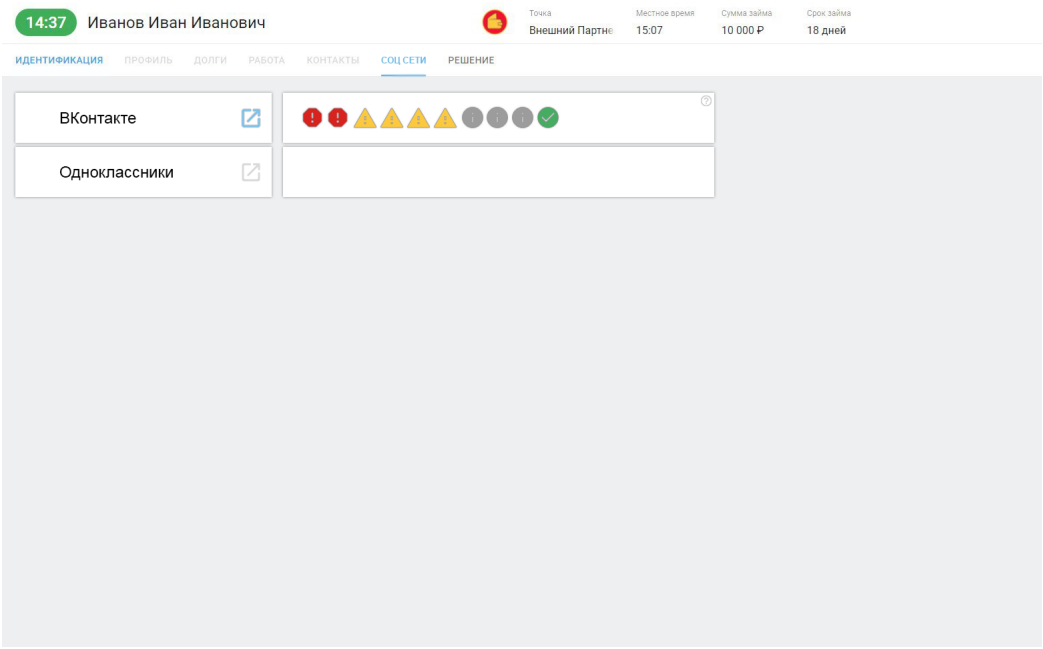

Рис. 6. Интерфейс веб клиента системы андеррайтинга