



Команда “Вахта”

Сервис для оценивания стоимости
объекта недвижимости

Введение

Человеческий труд при оценки стоимости объекта недвижимости несет в себе финансовые и временные затраты. Низкая конкурентоспособность, рентабельность, увеличение запросов на оценку залога при масштабировании - это все болевые точки бизнеса, которые мы хотим устранить с помощью нашего решения.

Как проблема решается сейчас:	Как предлагаем решить ее мы:
Использование человеческого труда: оценщики или внутренние специалисты	Автоматизировать процесс оценивания с помощью нашего сервиса оценки, основанного на технологии машинного обучения.

Наш сервис поможет:

снизить стоимость
оценки квартир

облегчить масшибирование
бизнеса за счет снижения
издержек

увеличить скорость
принятия решения

повысить лояльность
клиента

Анализ

1. Конкурентный анализ существующих решений:

Существующие модели обучения основаны на устаревших данных, потому имеющиеся методы вкупе с датасетами вряд ли можно считать релевантными в 2021 году.

2. Парсинг данных:

Скрипт для парсинга пишется на языке Python, так как для него существует много библиотек.

3. Технологии машинного обучения:

Использовались для апробации методы машинного обучения, которые являются наиболее популярными в случае регрессионного анализа¹

4. Технологии на back-end:

Мы сравнили протоколы HTTP и WebSocket, поскольку нам необходимо иметь возможность отправить результат вычислений с сервера в любой момент, мы выбрали WebSocket. Поскольку обработка данных происходит долго, для работы в фоновом режиме мы выбрали Celery+Rabbitmq. Также мы сравнили несколько API для оценки экологической обстановки по координатам. Лучшим по стоимости, количеству, качеству данных оказывается openweathermap.

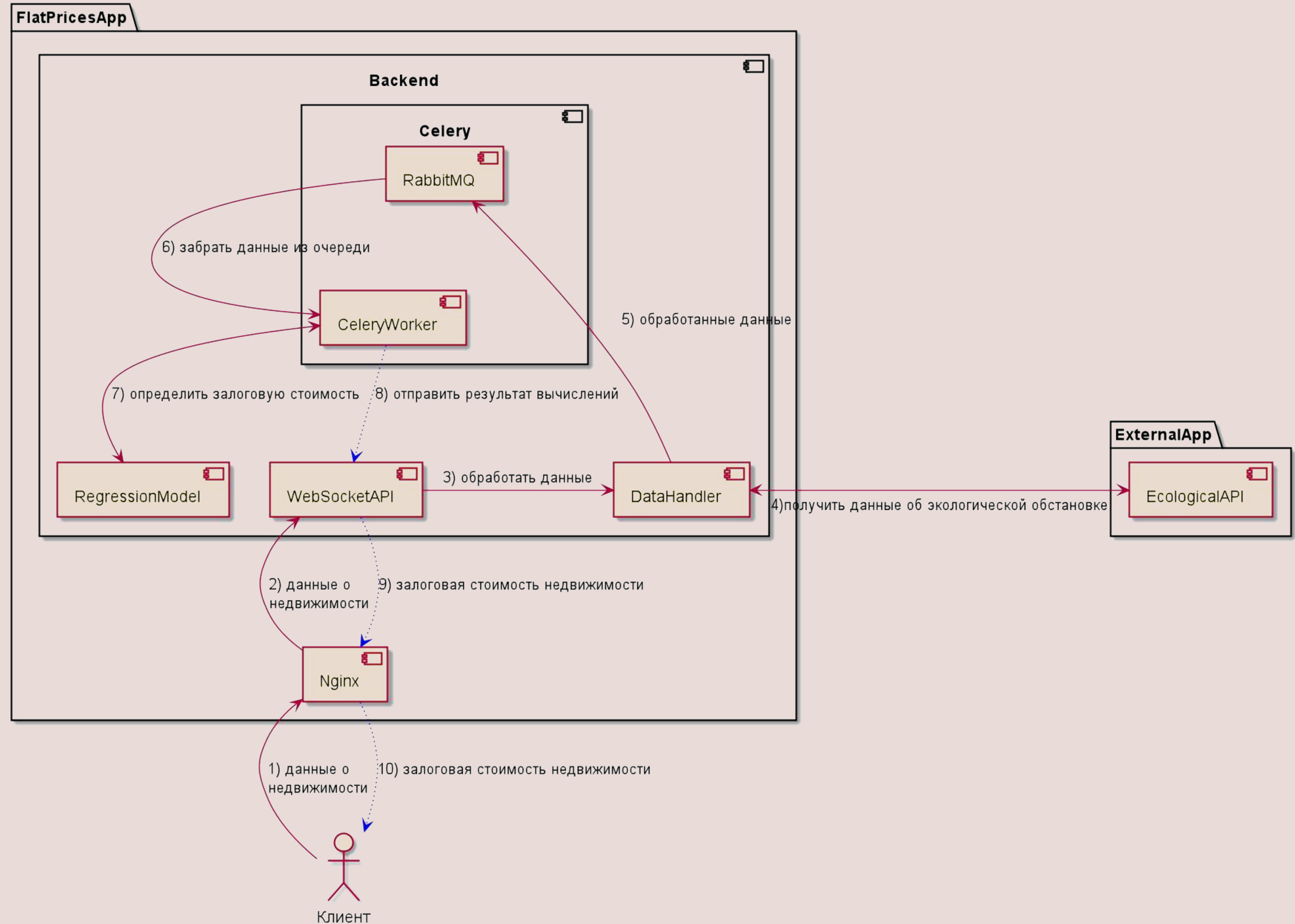
5. Технологии на front-end, дизайна и интерфейса:

Целевая аудитория: люди, старше 18 лет. Средний диапазон лет активных клиентов 30-50 лет. Поэтому интерфейс должен быть интуитивно понятным, достаточно простым, в спокойной цветовой гамме. Поэтому был выбран популярный Javascript фреймворк Vue.js, так как он довольно прост в освоении и на его основе можно создавать небольшие веб-приложения.

¹Подробнее в приложении №2

Наше решение

Клиент вводит данные, они приходят на сервер, клиенту приходит уведомление о принятии запроса. На сервере обогащаются данными о локации жилья и об экологии и отправляются обработке для оценки стоимости, клиенту отправляется рыночная стоимость, залоговая стоимость и данные об экологии. Более подробное описание решения представлено в приложении №1, №2, №3.



MVP и риски

Результаты MVP

Со стороны машинного обучения была использована модель BaggingRegressor, получена точность >82%¹

Результат вывода содержит в себе:

- предполагаемую рыночную цену жилья
- залоговую цену
- экологичность жилья

Можно посмотреть здесь:



Риск

Минимизация риска

Снижение точности системы на реальных данных ввиду колебаний рынка

Собственный парсер собирает данные об актуальных объектах, выставленных на продажу

Недостаточное количество параметров, по которым требуется определить стоимость

Усовершенствование сервиса по мере сбора статистики и анализа его работы после внедрения

Необходимость реорганизации работы кредитного сотрудника и сопряженные с этим ошибки в процессе обучения

Плавное введение сотрудника в новые обязанности, содействие, наставничество на каждом этапе

Увеличенная нагрузка на сотрудников на этапе внедрения, их сопротивление

Настроить систему мотивации сотрудников

Необходимость формирования квалифицированной группы внедрения

Привлечь к внедрению разработчиков сервиса на проектную занятость

Умалчивание клиентом фактов, влияющих на цену имущества; неточный ввод данных кредитным сотрудником – человеческий фактор

Запись разговоров, перепроверка финальной залоговой стоимости перед ее утверждением и подписанием договора

¹Результаты тестирования в приложении №2

Расчеты

Точка А

- На сайте Центр Инвест в каталоге залогового имущества выставлено на продажу 16 квартир и 28 домов
- Срок реализации должен быть в пределах 3х месяцев, то есть $(16+28)/3=14,6667$, примерно 15 объектов выкладываются в месяц
- По данным статистики, среди всех берущих кредит - 25% неплательщиков¹. Берем эту цифру за основу, хотя в реальности она может быть ниже, тк залог взимают в самых крайних случаях. $\Rightarrow 15*4=60$ одобренных кредитов под залог в месяц
- 36% всех кредитов одобряют росс банки (общ стат)² $\Rightarrow 167$ обращений в месяц получает банк Центр Инвест на оценку залога. Предположим, он обращается за оценкой в сторонние организации: 4.000 (стоимость оценки)* $167=$ **668.000 в месяц уходит на оценку стоимости залогов** (только дома и квартиры)

Точка Б

Расчеты на разработку и внедрение предлагаемого сервиса:

Средняя зарплата разработчика в инвест-банке:

- Разработка 5 мес: $70*5*4+35*5=1575$ тр
- Внедрение 2 мес: 140
- Поддержание: 35тр/мес
- Первоначальные вложения
Разработка+Внедрение=
1.715.000р

Окупаемость на 3ый месяц:
 $(1.715+(35*3))/668=2,7$, снижение
издержек на
 $35/668=0,0524$
94.76%

¹(https://lenta.ru/news/2021/04/22/ne_shmogli/)

²(<https://www.rbc.ru/finances/14/01/2020/5e1c91b89a7947..>)

Итог

Точность: **>82%**

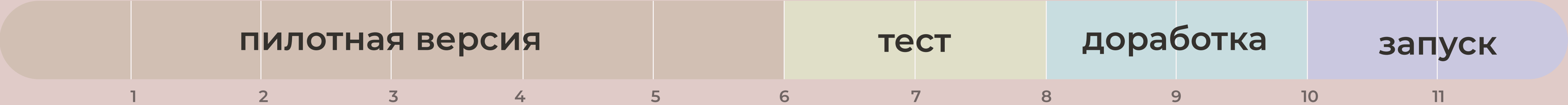
Сокращение
издержек: **94,76%**

Сокращение времени
на обслуживание: **98%**

Стоимость
разработки: **1 715 000р**

Вывод: повышение эффективности, рентабельности, легкости масштабирования.
невысокая стоимость разработки при высокой точности получения оценки, быстрота
сервиса

План выхода на рынок по месяцам:





1. Фёдор Кабаченко

- а. Специалист по машинному обучению и разработке на Python
- б. Опыт 3 года

2. Алёна Самарина

- а. Системны и бизнес-аналитик
- б. Опыт 4 года

3. Наталья Мальцева

- а. Бэкенд и фронтенд разработчик, специалист по базам данным
- б. Опыт 4 года

4. Любовь Ильина

- а. Финансист
- б. Опыт 2 года

5. Александр Зеленин

- а. Фронтенд разработчик
- б. Опыт 1 год

Приложение 1. Архитектурное решение

Со стороны back-end:

- 1) Т.к. обработка данных и предсказание стоимости не происходит мгновенно, нам нужно сразу ответить пользователю, что его запрос принят, а затем по инициативе сервера отправить результат. Поэтому мы выбрали вебсокет соединение, которое открыто постоянно и позволяет отправлять сообщения в обе стороны: от клиента серверу и обратно в отличие от HTTP запросов, которые основаны только на запросах от клиента.
- 2) Так как клиентов может быть много, нам нужно хранить их запросы и выполнять по очереди. Для этого мы используем сервис Celery вместе с брокером сообщений RabbitMQ.
- 3) Для того чтобы предлагать владельцам экологичных объектов более выгодные условия по кредитам, нам нужно дополнять полученные от клиента данные данными об экологической обстановке. Для этого бы выбрали открытый АПИ openweathermap, который позволяет оценить качество воздуха по шкале от 1 до 5, а также узнать концентрации вредных веществ по координатам.
- 4) Со стороны машинного обучения была использована модель BaggingRegressor.

Со стороны front-end:

- 5) Для создания интерфейса была осуществлена базовая верстка стандартными средствами CSS и HTML. Для взаимодействия с бэкендом была подключена библиотека socket.io. Данные между бэкендом и фронтендом передаются пакетами в формате json. Реализованы различные вариации поведения клиентской стороны, в том числе длительное ожидание от сервера, либо сообщение об ошибке.
- 6) Дизайн сайта осуществлялся в приложении Figma

Примечание:

- Использовались данные с сайта ЦИАН для регионов Санкт-Петербурга и Москвы;
- Мы предполагаем, что залоговая стоимость на 10% ниже рыночной
- Перспективы развития: высокая частота обновления данных в датасете (чтобы сервис был актуальным всегда), учитывание БОльших параметров экологичности (шумовое загрязнение, экологические выбросы, близость к загрязняющим источникам и т.д)

Приложение 2. Тестирование методов обучения

Для обучения модели использовался собственный датасет с данными парсинга сайта ЦИАН. Мы посчитали, что с учетом ежегодных корректировок цен на недвижимость, лучше воспользоваться свежими данными с актуальными ценами. Парсинг был проведен для городов Москва и Санкт-Петербург.

Использовались для апробации методы машинного обучения, которые являются наиболее популярными в случае регрессионного анализа:

- 1) SGDRegressor
- 2) Ridge
- 3) Lasso
- 4) ElasticNet
- 5) LarsCV
- 6) BayesianRidge
- 7) KNeighborsRegressor
- 8) DecisionTreeRegressor
- 9) LinearSVR
- 10) SVR
- 11) AdaBoostRegressor
- 12) BaggingRegressor
- 13) ExtraTreesRegressor
- 14) GradientBoostingRegressor
- 15) RandomForestRegressor

В ходе блиц-прогона обучающих моделей, наилучший результат показали Extra Trees, BaggingRegressor , Light Gradient Boosting и Random Forest.

Далее параметры регрессов были скомпенсированы и откорректированы под оптимальные значения, и самый лучший результат показал BaggingRegressor. О выборе параметров написано в приложении №3

