Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра математической
кибернетики и компьютерных наук

РАЗРАБОТКА СИСТЕМЫ СБОРА И АНАЛИЗА ДАННЫХ РЫНКА НЕДВИЖИМОСТИ В Г. САРАТОВЕ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Карелова Максима Дмитриевича

Научный руководитель
доцент, к. ф.-м. н. _____ В. Г. Самойлов

Заведующий кафедрой
к. ф.-м. н. _____ С. В. Миронов

Саратов 2017

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В открытых источниках на информационных сервисах имеется информация, которая может подлежать сбору и обработке, чтобы на основе получаемых данных отобразить общую картину состояния рынка для анализа.

На текущий момент нет доступных открытых многопользовательских решений для получения исторической информации по состоянию предложений рынка недвижимости. Их использование дало бы представление о его состоянии в определенных временных промежутках в прошлом, а также позволило бы совершать возможные предсказания на основе тех трендов рынка, которые могут быть обнаружены при более высокоуровней оценке состояний как рынка в целом, так и отдельных его областей, включающие различные наборы характеристик.

Рассмотрение подобной задачи может быть значимо в бизнесе, особенно в условиях кризиса. В такое время на рынке недвижимости снижается спрос на квартиры. Вместе с этим растут цены на отдельные предложения. Это связано с человеческой психологией — одни стараются продать квартиру на падающем рынке и понижают цену в надежде привлечь оставшихся покупателей, другие пытаются компенсировать нестабильность рубля и рост потребительских цен, завышая стоимость недвижимости. Рынок недвижимости крайне неоднороден.

Решение проблемы — разработка системы оценки стоимости объектов недвижимости, с помощью анализа объявлений в интернете которая сможет помочь оценивать их стоимость относительно стоимости подобных предложений во времени. Клиент — условное агентство недвижимости или обычный человек, которому необходимо иметь упрощенный доступ к процессу определения рыночной стоимости квартиры или комнаты с целью сравнения.

Задача сводится к мониторингу объявлений рынка недвижимости и пополнением данных этой информацией.

Цель работы — создание системы, которая будет в автоматическом режиме собирать информацию о предложениях из открытых источников, обрабатывать, хранить ее определенным образом и по собираемым данным, давать пользователям возможность увидеть настроенные им отчеты.

1 Описание проблемной области

1.1 Определение и декомпозиция задачи

Для выполнения поставленной цели, были сформулированы требования по созданию полноценной многопользовательской онлайн системы, в которой весь доступный функционал может быть использован в режиме реального времени на основе настроек, указанных пользователем.

Подобная масштабная задача требует детальной декомпозиции на более мелкие части. В качестве основного показателя в первую очередь учитывались максимальная направленность на пользователя и возможность удобного представления информации с возможностью выбирать параметры самим пользователем по необходимости.

При оценке различных способов визуализации информации, содержащей различные поля и имеющей некоторое количество метрик, подлежащие к использованию вместе с фильтрами, были выбраны следующие способы представления результирующих данных, применяющихся для отображения метрик:

1. круговая диаграмма;
2. столбцовая диаграмма;
3. график с использованием кривых;
4. точечное представление на карте;
5. представление на карте результатов выполнения алгоритма пространственной интерполяции по заданным данным.

1.2 Анализ потенциальных источников данных

Для подготовки качественной и достоверной аналитики используются только эксклюзивные и проверенные статистические данные, накопленные в процессе длительной работы на рынке недвижимости. Качество любой аналитики строится на свойствах этих статистических данных. Сложность аналитики в недвижимости заключается именно в наличии единых качественных источников информации. Это объясняется закрытостью сферы.

Среди групп источников информации фактически доступны для использования только специализированные публичные ресурсы в сети интернет, среди которых можно выделить следующие:

— интернет-ресурсы, предоставляющие возможности публиковать поль-

зователям различные объявления, например, можно выделить «Avito», «Из рук в руки»;

— государственный интернет-портал «РосРеестра», предоставляющий возможности для получения достоверной информации о фактах совершения сделок о недвижимости на рынке.

1.3 Определение необходимых метрик

Для визуального представления в виде графиков были составлены следующие группы:

1. получение общего количества предложений по парам конфигураций: квартира и продажа, квартира и аренда, комната и продажа, комната и аренда;
2. в дополнение к фильтрам из пункта 1 конфигурации использование информации о районе, в котором находится объект;
3. в дополнение к фильтрам из пунктов 1 и 2, использование фильтров по информации о количестве комнат;
4. дополнительная группа, которая представляет собой использование всех вышеописанных сценариев, с получением результата в виде серии временных меток, но не с количеством доступных предложений, а с информацией о стоимости одного квадратного метра.

1.4 Архитектура системы и структура компонентов

В соответствии с поставленными требованиями были определены основные подсистемы визуализации для работы с первой и второй группой представлений:

1. подсистема представления на основе графиков;
2. подсистема представления на основе географического расположения объекта на карте.

Для написания серверных частей использовалася питон библиотека Flask, которая решает задачу быстрого написания программного API слоя для обмена информацией между компонентами.

Для создания пользовательского интерфейса было решено использовать Javascript библиотеку Angular Material.

1.5 Выбор технологий и инструментов

При анализе способов выделения серверных ресурсов для работы частей системы, был выделен AWS EC2 для решения данной задачи.

Вычислительное облако **Amazon Elastic Compute Cloud** — это веб-сервис, предоставляющий масштабируемые вычислительные ресурсы в облаке. Он позволяет упростить процесс выполнения вычислений через Интернет для разработчиков.

Amazon S3 — хранилище для Интернета. Это сервис простого хранения данных, предлагающий разработчикам ПО надежную инфраструктуру хранилища данных с высокой масштабируемостью и низкой задержкой при очень незначительных затратах.

Amazon Athena — это интерактивный сервис запросов, позволяющий легко анализировать данные в хранилище Amazon S3 с помощью стандартных средств SQL. Athena — бессерверный сервис, здесь нет инфраструктуры, требующей настройки или управления, поэтому можно сразу же приступить к анализу данных.

Graphana предназначена для отображения всевозможных циклических метрик. Помимо предопределенных системных метрик, можно сконфигурировать любой набор произвольных метрик [1], а также создавать настраиваемые пользователем панели. Так как для решения описываемой в данной работе проблемы данный продукт не позволяет получить результат «из коробки», но предоставляет возможность его расширения путем написания собственных обработчиков данных, которые на основе задекларированного в документации интерфейса могут иметь общее взаимодействие, то было принято решение о его использовании совместно с плагином SimpleJsonDatasource и написании собственного API сервера.

В данном случае результатом решения этой задачи интерполяции является получение цифрового изображения с отображением определенного цвета для определенных заданных участков плоскости [2]. Цвета на карте можно соотнести с цветами на «легенде», которая будет заранее определена. Цвет на «легенде» соответствует средней стоимости квадратного метра общей площади в тысячах рублей.

Интерполяция по методу обратных взвешенных расстояний (IDW) использует предположение, что объекты, расположенные ближе к другу, в боль-

шей степени похожи, чем удаленные друг от друга. Чтобы найти значение в какой-либо точке, метод IDW использует опорные точки, находящиеся в окрестностях искомой.

$$Z(x_0, y_0) = \sum_{i=1}^{N} \lambda_{i,j} Z(x_i, y_j),$$

где

$Z(x_0, y_0)$ — значение свойства в искомой точке,

$\lambda_{i,j}$ — весовой коэффициент для значения свойства в каждой опорной точке $Z(x_i, y_i)$ из числа тех, что будут учитываться в вычислениях,

N — число опорных точек, находящихся в окрестности искомой точки и используемых в вычислениях.

Веса определяются по следующей формуле:

$$\lambda_i = d_{i0}^{-p} / \sum_{i=1}^{N} d_{i0}^{-p},$$

где

d_{i0} — расстояние между искомой точкой (x_0, y_0) и i-ой опорной точкой (x_i, y_i).

С увеличением расстояния вес уменьшается за счет коэффициента p.

$$\sum_{i=1}^{N} \lambda_i = 1$$

Jenkins CI — проект для непрерывной интеграции с открытым исходным кодом, написанный на Java. Позволяет автоматизировать часть процесса разработки программного обеспечения, в котором не обязательно участие человека, обеспечивая функции непрерывной интеграции.

2 Компоненты системы

Реализация компонентов и связей между ними, а также выбор технологий и инструментов были основаны на схеме проекта.

2.1 Компонент RealtyGeoCacheService

В ходе работы над компонентом сбора данных было обнаружено, что на веб-страницах сайтов с объявлениями о продаже и аренде недвижимости отсутствует информация о географических координатах объектов, но в то же время присутствует информация о районе города и адрес. Таким образом была поставлена подзадача по созданию промежуточного микро-сервиса для RealtyCollector, который бы реализовывал данную логику конвертации текстового описание адреса объекта в его географические координаты, получая запросы с информацией с HTML страниц сайтов.

Для реализации согласно определенным требованиям была разработана структура микросервиса. На схеме было описано наличие объекта Redis. В ходе анализа работы сервиса была найдена проблема с сокращением количества запросов к стороннему API от сервиса «Яндекс Карт» с целью уменьшения времени отклика самого компонента. Для этого было выбрано решение основанное на использовании кеширования, для чего и был выбран Redis [3, 4] как подходящий для этой задачи продукт.

Спецификация запросов к данному сервиса была определена в виде шаблона https://URL/cache?address=ADDRESS

Реализация сервиса программно задается функцией маршрутизации запросов с использованием библиотеки Flask.

2.2 Компонент RealtyCollector

Для сбора данных было рассмотрено решение с парсингом данных на web-страницах сайтов и добавлением их в общую систему хранения [5].

Результатом прохода программы по страницам сайта является файл в формате CSV.

С целью эффективного хранения и быстрой возможности получения файлов было решено использовать bucket. Данная сущность была создана с именем realty-information для дальнейшего размещения в ней файлов.

С использованием AWS SDK для языка GoLang была реализована работа с файлами и их загрузка в установленный bucket.

2.3 Компонент RealtyGrafanaDatasource

Для получения возможности визуализации в Grafana было необходимо реализовать интерфейс взаимодействия между системами.

Для реализации данного API сервиса в соответствии с документацией Grafana UI необходима реализация URI с контроллерами для HTTP запросов [6].

В этом случае работа с данными представляет собой использование JDBC драйвера на языке программирования Java. С его помощью в системе реализован доступ SQL запросами к хранящимся данным.

Таким образом, мы можем использовать практически полноценный SQL синтаксис для запросов записей.

Для необходимых нам метрик была составлена таблица с шаблонами наименований соответствующих метрик для программного доступа.

Подключение доступа к данным в контроллерах API:

Для фильтрации и получения данных были составлены SQL запросы для каждого типа шаблона метрик.

2.4 Компонент RealtyMapService

Задача сервиса состоит в предоставлении пользователям архивной информации о предложениях, а также в выполнении интерполяции для получения цветовой схемы на основе выбранных данных на географической карте и построению предположений об областях на карте без существующей информации.

Для выполнения задачи интерполяции была реализована программа на языке Python, с применением метод обратных взвешенных состояний.

Для расчета расстояний и для трансляции географических координат в (X, Y) координаты на растровой плоскости применяется упрощенная линейная формула без применения используемой в Google MapsWeb Mercator проекции [7].

Также используется функция, которая производит обратное вычисление из координат точек на растровой плоскости в точки с географическими координатами [8].

В результате работы алгоритма на выходе получается изображение.

Для интеграции полученного изображения с RealtyMapService в HTML

код основной страницы была добавлена специальная функция.

После использования соотнесения данных с изображения созданного методом обратных взвешенных расстояний и картой в веб-странице пользователя становится доступным визуализация на карте.

В итоге был создан одностраничный интерфейс, состоящий из двух основных частей:

— область в браузере с картой города (левая часть экрана);
— область с фильтрами, необходимыми элементами управления и кнопками (правая часть экрана).

Для работы с картами использованы Google Maps с Javascript библиотекой SDK и API, позволяющие работать с динамическими данными на географической плоскости.

Элементы веб-страницы, необходимые для конфигурирования результатов оценки были разделены на условные визуальные строки:

— выбор даты;
— тип недвижимости;
— тип сделки;
— район города;
— этаж от/этаж до;
— комнат от/комнат до;
— цена от/цена до;
— площадь от/площадь до;
— выбор сгенерированного ранее отчета пространственной интерполяции.

В итоге описанные элементы были представлены в виде HTML кода страницы.

3 Внедрение системы

3.1 Настройка работы подсистемы с графиками

Реализованный DataSource сервис был подключен в Grafana UI для получения визуального результата работы, и на основе возвращаемых им данных были настроены панели с использованием источников-метрик.

3.2 Расширяемость компонентов

Созданное решение предоставляет возможности для неограниченного расширения текущей функциональности под различные задачи.

В целом данная система была нацелена на работу с данными по городу Саратов, но может быть расширена на большее количество городов добавлением дополнительных запусков RealtyCollector с настройками для необходимых населенных пунктов.

Компонент RealtyGrafanaDasource позволяет программно добавлять новые метрики и определять для них необходимые SQL запросы, которые могут возвращать данные для отображения на графиках с заданной обработкой.

RealtyMapService может быть расширен добавлением Javascript обработчиков для выполнения определенных задач.

3.3 Развертывание системы

Для развертывания системы был использован контейнеро-ориентированный подход на основе применения Docker технологии.

Python компоненты были обернуты в Docker образы с помощью общего для них Dockerfile с realty-grafana, realty-geocache, realty-map тегами соответственно.

Для каждого компонента были созданы Jenkins задачи, которые собирали артефакты из соответствующих репозиториев и запускали компоненты из новых версий на сервере.

Для автоматизации работы RealtyCollector были созданы две Jenkins задачи. Первая — для задачи оркестрации выполнения программы с конкретными параметрами по расписанию «H 12 * * *». Во второй были заданы переменные окружения для конфигурации запуска с использованием определенного прокси-сервера и доступа к AWS S3 bucket под конкретным IAM пользователем.

3.4 Оценка стоимости ресурсов для работы решения

Для развертывания компонентов системы, занимающихся вычислениями, был выбран AWS EC2 сервис, а для веб-компонентов — Heroku.

Экспериментальным путем был подобран on-demand t2.small сервер-инстанс EC2, обладающий конфигурацией вычислительной мощности: высокочастотный Intel Xeon процессор, 1 vCPU, 2 GiB оперативной памяти, 20 GB SSD памяти. Стоимость использования составляет $0.023 за час работы. Таким образом для обеспечения бесперерывной работы необходимо $0.552 за сутки или $16.56 в месяц за использование виртуализированного сервера.

Для работы файловой системы сервера используется Amazon EBS General Purpose SSD (gp2) volume объемом 20 гигабайт. Стоимость одного гигабайта в месяц составляет $0.10, таким образом, общая стоимость блочного хранения файлов составляет $2.

Для архивирования CSV файлов с данными был использован один bucket в AWS S3 сервисе. Тип хранения — «Standard Storage», месячная стоимость которого зависит от общего объема хранимых данных и составляет $0.023 за один гигабайт в месяц.

Для одного города Саратова количество данных, которое собирается ежедневно составляет в среднем 15032 записей и занимает около 15 мегабайт хранилища. На момент 16.05.2017 система содержит информацию за 3 месяца, что в объеме занимает 1350 мегабайт данных с ежемесячным приростом порядка половины гигабайта данных. Таким образом получается, что месячная стоимость хранения данных составляет около $0.069 с ежемесячным увеличением издержек на хранение в размере $0.019.

Цена использования AWS Athena составляет $5 за каждые 5 терабайт сканированной информации с погигабайтной тарификацией. Стоимость использования в месяц составляет порядка $2.

По предварительным расчетам, издержки на содержание системы для объема данных, ограниченных на основе одного города составляют около $20.629 в месяц или 1185.37 рублей (состояние курса валют на 18.05.2017) с ежемесячными приростами издержек в размере около $0.019 или 1.09 рублей и зависят от того, насколько много информации хранится и обрабатывается на сервере.

ЗАКЛЮЧЕНИЕ

В рамках данной работы была достигнута поставленная цель и выполнены соответствующие ей задачи. Были рассмотрены и изучены способы сбора и обработки информации, визуализации собираемых данных о недвижимости с использованием современных технологий, создан двухкомпонентный web-сервис, предоставляющий возможности получать статистику на основе данных за предыдущие промежутки времени, а также использован один из алгоритмов пространственной интерполяции, на основе которого наглядно демонстрируется цветовая схема распределения стоимости недвижимости на карте для участков с отсутствующими данными. Было описано развертывание системы с применением облачных технологий и проведена стоимостная оценка ежемесячных издержек за используемые вычислительные ресурсы. Также было проанализировано, какие возможные технические средства могут быть применимы для обработки BigData в описанной области, и какие метрики могут быть собраны. Рассмотрена значимость полученного результата и проведена оценка значимости подобной информации для ее использования в своей области.

Таким образом, созданная система имеет широкие возможности для применения в сфере анализа определенных тенденций на рынке недвижимости в городе Саратове, а также практическую пользу для пользователей, которые могут на основе имеющейся информации строить предположения о ситуации на рынке в целом на ближайшее будущее.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Julian, M.* Practical Monitoring: Effective Strategies for the Real World / M. Julian. — O'Reilly Media, 2017.

2 *Stein, M.* Interpolation of Spatial Data / M. Stein. — Springer, 2005.

3 *Kreibich, J.* Redis: The Definitive Guide: Data modeling, caching, and messaging / J. Kreibich. — O'Reilly Media, 2014.

4 *Nadareishvili, I.* Microservice Architecture: Aligning Principles, Practices, and Culture / I. Nadareishvili, R. Mitra, M. McLarty, M. Amundsen. — O'Reilly Media, 2016.

5 *Атовмян, И.* Сбор и обработка исторических данных в автоматизированных информационных системах / И. Атовмян. — Синергия, 2012.

6 Grafana Documentation [Электронный ресурс] / Grafana Labs. — URL: http://docs.grafana.org/ (Дата обращения 23.04.2017). Загл. с экр. Яз. англ.

7 *Steffensen, J. F.* Interpolation: Second Edition / J. F. Steffensen. — Dover Publications, 2012.

8 *Davis, P. J.* Interpolation and Approximation / P. J. Davis. — Dover Publications, 2014.

05.06.2017