

Геоаналитика: сбор, анализ и отображение географических данных для решения задач размещения коммерческих организаций

Гринева Наталья Владимировна,

к.э.н., доцент Департамента анализа данных и машинного обучения, Финансовый университет при Правительстве Российской Федерации, NGrineva@fa.ru

Топыркин Алексей Дмитриевич,

студент магистратуры Института ЭМИТ, Российская академия народного хозяйства и государственной службы при Президенте РФ, topyrkinalexei@yandex.ru

Актуальность статьи заключается в формализации процесса сбора географических данных для решения задачи размещения коммерческих организаций и их подразделений. Целью исследования является решение задачи размещения и оценки степени влияния географических характеристик локаций на показатели экономической эффективности коммерческой организации. В рамках статьи определены понятия экономической эффективности и геоаналитики, а также то, как они связаны между собой. Описан ряд задач для решения которых используется геоаналитика. Детально освещены вопросы относительно используемых географических данных, а именно даны ответы на вопросы. Что? Как? Где? Что - какие факторы можно использовать для выявления потенциала локаций, как эти данные можно агрегировать и хранить, где их непосредственно собирать. Авторами отмечается, что главными инструментами для решения подобных задач математического моделирования, анализа данных - являются язык программирования Python и его библиотеки pandas, requests, selenium, h3, osmnx, networkx. Предложен процесс построения маршрутов между двумя точками в рамках задачи поиска наикратчайшего расстояния. В заключении продемонстрирован один из возможных способов отображения пространственных данных с помощью модуля kepler.gl. Результатом работы являются собранные географические факторы, которые в дальнейшем будут использованы в процессе построения модели для определения экономического потенциала локаций коммерческих организаций и их подразделений.

Ключевые слова: геоаналитика, математическое моделирование, анализ данных, экономическая эффективность, гексагоны, язык программирования Python.

Введение

Сейчас в эру цифровых технологий, высокой конкуренции и открытости информации для коммерческих организаций особенно актуально и остро стоит вопрос экономической эффективности бизнеса. Уже нельзя просто экстенсивно расширять свое производство и увеличивать обороты за счет открытия новых точек, так как в таких местах возрастает количество потенциальных конкурентов или же целевая аудитория конкретной фирмы. Именно поэтому в современном мире на первое место и выходят исследования, за счет которых в дальнейшем лицо принимающее решение и осуществляет свой выбор по открытию нового магазина, отделения, офиса и т.п.

Исследования такого рода и носят название «геоаналитика». Геоаналитика – это лишь один из многих терминов, используемых для описания анализа данных на основе местоположения. Другие названия для этого включают в себя: разведку местоположения, геолокацию, геопрограммный анализ, географический анализ и многие другие. На самом деле все эти термины описывают одну и ту же основную идею: применение географии к конкретным бизнес-процессам для получения аналитических сведений.

С помощью геоаналитики можно решить целый ряд задач, например, таких как: визуализация данных с географической точки зрения; интерактивное картографирование; географический анализ и поиск; кластеризация и построение тепловое карт; тематическое картографирование; проектирование и управление территориями; оптимизация маршрутов; прогнозирование.

Важность геоаналитики заключается в том, что она предоставляет для бизнеса новую точку зрения, основанную на их данных. В частности, когда речь идет о данных CRM, отчетов и таблиц, то этого недостаточно для того, чтобы действительно выявить нужные практические идеи. Геоаналитика берет данные, которые есть у организации и отображает их на карте, упрощая понимание и анализ полезной информации.

Отдельно необходимо остановиться на термине «экономическая эффективность». Пол Энтони Самуэльсон и Уильям Нордхаус в своей книге «Экономика» дают следующее определение: «это получение максимума возможных благ от имеющихся ресурсов, постоянно соотнося выгоды (блага) и затраты, при этом необходимо вести себя рационально. Производитель и потребитель благ стремятся к наивысшей эффективности, максимизируя при этом свои выгоды и минимизируя затраты» [2, с.55].

Джон Кеннет Гэлбрейт писал, что экономическую эффективность организационных форм корпораций определяет способностью достигать поставленные цели: защитные и положительные. Следовательно, критерием экономической эффективности организационных форм корпораций будет достижение каждой заинтересованной группой определенного уровня доходов и рост самой корпорации [1, с.144]. В то же время Луис Кабрал в книге «Introduction to industrial organization» понимал под экономической эффективностью, в первую очередь, «эф-

эффективность распределения ресурсов (аллокативная эффективность). Уже само это определение говорит о том, что распределение ресурсов должно быть эффективным» [7].

В нашей работе за основу примем трактовку, которую сформировали Пол Энтони Самуэльсон и Уильям Нордхаус, так как наиболее точно и полно обобщает все остальные определения. Измерить экономическую эффективность можно с помощью целого ряда различных показателей: рентабельности, фондоотдачи, производительности труда, максимального объема выпускаемой продукции и, конечно, прибыльности. Однако, из определений представленных выше, можно было заметить, что большинство ученых сходятся к тому, что основным показателем отражающим экономическую эффективность является прибыль.

Таким образом, экономическая эффективность и геоаналитика на сегодняшний момент очень тесно связаны между собой, так как любой офлайн бизнес стремится максимизировать отдачу от вложенных ресурсов, а геоаналитика, в свою очередь, дает организации тот инструментарий, на основе которого она может принять максимально эффективное решение, основанное не просто на догадках или соображениях, но и подтвержденное реальными результатами.

Постановка задачи

Высокая значимость количественной оценки взаимосвязи между изменениями географического расположения объекта и изменениями экономических показателей эффективности обусловлена тем, что выявленные закономерности лежат в основе инвестиционного анализа организации.

По мнению ряда ученых использование географических данных для решения задач размещения является, в достаточной степени эффективным методом (Вебер и Чепмен, 2011 г. [17]; Гош, 2008 [8]; Харрис, Слейт и Уэббер, 2005 [9]; Янковский, 1995 [12]). В иностранной литературе даже выработался отдельный термин «Location Intelligence», который можно интерпретировать, как «аналитика, основанная на местоположении». Появился этот тип аналитики в результате развития такого молодого направления в науке, как геоинформатика (geoinformatics).

Патрик Вебер и Дэйв Чепмен [17] из университетского колледжа Лондона в своей работе о инновационном подходе к расположению бизнеса приводят пример столицы Великобритании, города Лондона, и показывают, как разбиение города на районы может помочь в классификации и выявлении оптимальных мест под размещение различных коммерческих организаций.

Нашей основной задачей в рамках этой статьи будет описание исходных данных, а именно того, какие факторы можно использовать для определения потенциала локации, как эти факторы агрегировать и хранить, и где их непосредственно получать. Все эти моменты необходимо решить, так как именно на основе географических данных основывается решение задачи размещения. Полученные в рамках этого исследования результаты потребуются в дальнейшем для построения модели по оценке экономического потенциала локации.

Описание собираемых данных

В рамках исследования будут собираться 5 основных групп факторов:

- мобильные данные (предоставляются сотовыми операторами);
- GPS данные (предоставляются агрегаторами информации по мобильным приложениям);
- данные о недвижимости (собираются с сайта «ЦИАН»);
- данные о бизнесе (собираются через API сервис Яндекс.Карт);

- данные о постройках (собираются с сайта «OpenStreetMap»).

Каждая из категорий представляет собой совокупность различных признаков, составляемых и агрегируемых по определенным наборам правил. Структура данных внутри каждой группы в достаточной мере схожа и ее можно представить в следующем виде:

- фактор (конкретная информация, которая есть по этой группе);
- метод (каким образом агрегируются факторы в рамках определенного радиуса);
- время (период, за который была собрана информация);
- радиус (область, в рамках которой собирается информация по конкретному фактору);
- класс (уточняющий фильтр для фактора).

Детальное описание группы под номером 2 по структуре, описанной выше, представлено в таблице 1.

Таблица 1
Описание GPS данных

Тип	Описание
Фактор	<ul style="list-style-type: none"> - Количество сигналов; - Количество пользователей; - Количество сигналов жителей; - Количество жителей; - Количество сигналов работающих; - Количество работающих.
Метод	<ul style="list-style-type: none"> - Сумма; - Максимум.
Время	<ul style="list-style-type: none"> - весь день; - 0-6 часов; - 6-12 часов; - 12-18 часов; - 18-24 часа.
Радиус	<ul style="list-style-type: none"> - 75 метров; - 150 метров; - 300 метров; - 500 метров.
Класс	<ul style="list-style-type: none"> - Пешеходы; - Автомобили; - Все.

Таким образом, структура объясняющих переменных для моделирования и оценки экономической эффективности локаций будет выглядеть так. На примере таблицы 1 можно посчитать, какое количество GPS факторов будет нам доступно. Система расчета выглядит следующим образом:

Количество переменных = Фактор * Метод * Время * Радиус * Класс,

где: Фактор, Метод, Время, Радиус, Класс – это количество принимаемых значений для соответствующей строки из таблицы 1. Если строка пуста, то ее значение стоит принять равным единице.

В случае с данными GPS будет получено $720 (6 * 2 * 5 * 4 * 3)$ факторов. С остальными группами ситуация аналогичная.

Система агрегирования данных

Для прогнозирования используется географические факторы и получение этой информации не настолько тривиально. Для упрощения работы с огромными объемами данных используется следующая схема.

По сути, количество точек, которое потенциально можно разместить на нашей планете бесконечно. Как следствие, даже если у нас была возможность собрать данные о социальной-экономической характеристике всех точек, то встал бы целый ряд других вопросов. Где хранить такой объем информации? Логично, что таблица с данными была просто колоссальных размеров. И второе, что приходит в голову – это то, как

можно за измеримое количество времени эти данные достать и обработать, ведь база данных бесконечных масштабов?

Именно поэтому данные необходимо агрегировать в какие-то более крупные материи нежели, чем простая точка, обладающая долготой и широтой. Простейшим решением в таком контексте является совокупность точек или геометрия. Например, данные о количестве людей в конкретных точках можно обобщить до среднего или суммарного количества человек в определенном квадрате, прямоугольнике, круге и т.д.

Одним из решений, которое предложила компания Uber, является разбиение всего земного шара на отдельные элементы – гексагоны (шестиугольники). Сеточная система имеет решающее значение для анализа больших наборов пространственных данных, разбивая области Земли на идентифицируемые ячейки сетки. Имея это в виду, Uber разработал H3, свою систему сетки для эффективной оптимизации ценообразования и диспетчеризации поездок, для визуализации и изучения пространственных данных. H3 позволяет анализировать географическую информацию, устанавливать динамические цены и принимать другие решения на общегородском уровне. Фактически алгоритм представляет из себя систему иерархических индексов [10]. Визуально это можно посмотреть на рисунках 1, 2.

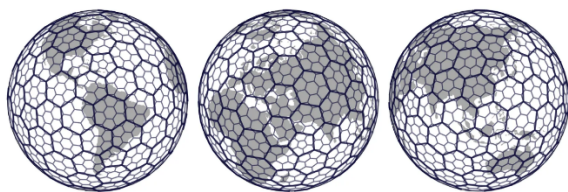


Рисунок 1 – Сеточная структура в рамках подхода H3

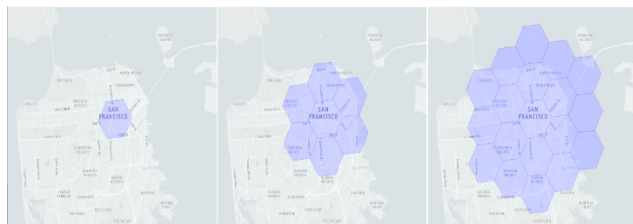


Рисунок 2 – Визуализация шестиугольной структуры

Каждый шестиугольник H3 можно рассматривать как отдельный объект, и к каждому объекту можно получить доступ за очень короткий промежуток времени, учитывая его идентификатор. Основная сила H3 в том, что он покрывает весь мир шестиугольниками разного размера. Это означает, что разрешение слоя можно регулировать в зависимости от решаемой задачи, например, масштабировать всю сетку вверх и вниз. H3 содержит в общей сложности 16 разрешений, как описано в таблице 2 [15], и каждое разрешение имеет определенное количество шестиугольников, которые охватывают всю землю как слой, начиная от 122 шестиугольников на самом высоком уровне и около 500 триллионов шестиугольников на самом низком уровне. Каждый слой состоит из более детализированного уровня шестиугольников, и каждый шестиугольник каждого слоя имеет свой уникальный идентификатор [16].

H3 определяет свои индексы в шестнадцатеричном формате (16 бит и, следовательно, 16 разрешений), поэтому разрешение ячейки можно сразу определить, взглянув на идентификатор ячейки. Каждый шестиугольник с низким разрешением содержит набор дочерних шестиугольников с более высоким разрешением. У каждого шестиугольника могут быть соседние шестиугольники, которые имеют один и тот же набор

родительских шестиугольников. Слои, по сути, определяют дерево шестиугольников, последний слой (разрешение) содержит 500 триллионов братьев и сестер [16].

Таблица 2

Характеристики сеточной структуры H3 в зависимости от разрешения

Разрешение	Количество гексагонов	Средняя площадь гексагона (км ²)	Средняя длина ребра (км)
0	110	4357449,416078380	1107,712591000
1	830	609788,441794133	418,676005500
2	5870	86801,780398997	158,244655800
3	41150	12393,434655088	59,810857940
4	288110	1770,347654491	22,606379400
5	2016830	252,903858182	8,544408276
6	14117870	36,129062164	3,229482772
7	98825150	5,161293360	1,220629759
8	691776110	0,737327598	0,461354684
9	4842432830	0,105332513	0,174375668
10	33897029870	0,015047502	0,065907807
11	237279209150	0,002149643	0,024910561
12	1660954464110	0,000307092	0,009415526
13	11626681248830	0,000043870	0,003559893
14	81386768741870	0,000006267	0,001348575
15	569707381193150	0,000000895	0,000509713

Алгоритмы сбора данных

Именно в рамках этого подхода и реализуется сбор, хранение и обработка данных для решения задачи размещения, то есть вся точечная информация агрегируется до каких-то более крупных геометрий. Возвращаясь к началу исследования, можно сказать следующее о 5 основных группах геофакторов.

Мобильные данные поступают в формате полигональной сетки и перевод их к формату шестиугольной разметки мог бы сопровождаться потерей части информации, именно поэтому они хранятся в исходном виде.

GPS данные идут приходят от поставщиков в формате гексагонов с разрешением 10.

Недвижимость во время парсинга с сайта ЦИАН представляет из себя точечные данные, которые характеризуются определенными координатами и сопутствующей информацией в виде количества квартир в доме и стоимости 1 квадратного метра жилья [4]. Сам процесс индексации использует уникальную систему идентификаторов. Если взять какие-то определенные координаты, затем перекодировать их в индекс шестиугольника, то исходной долготы и широты будет соответствовать лишь один единственный гексагон в определенном разрешении (соответственно 16, если рассматривать все разрешения) [11]. И это позволяет упростить задачу агрегации до чего-то максимально тривиального. Достаточно просто для всех координат объявлений получить уникальные идентификаторы гексагонов с разрешением 10, в которых они находятся. После чего можно спокойно осуществить группировку результатов о квартирах и стоимости по id шестиугольников.

Данные о бизнесе также, как и данные о недвижимости в сыром виде представляют из себя точечную информацию. Собирается она аналогично с помощью парсинга и запросов к Яндекс.API. Так как в возможности API не входит предоставление данных об организации (отзывов, рейтингов), то процесс получения этой информации усложняется. Для начала в целом становится необходимым получения списка организаций, по которым уже в дальнейшем можно будет реализовать получение метаданных. В рамках Яндекс.API есть возможность узнать,

присутствует ли какая-то определенная организация в заданной области. Причем необязательно указывать конкретное название точки, можно ограничиться классом, к которому эта точка относится. В таком случае структура запроса будет выглядеть следующим образом. Нужно передать: API ключ; название организации или класс; язык, на котором будет выдан ответ; в типе возвращаемых результатов указать «организации»; в границах области поиска задать координаты долготы и широты нижнего левого и правого верхнего углов, а также необходимо явно передать, что за пределами данной области поиск осуществлять не нужно [3]. После отправки такого запроса в ответ должен прийти JSON, который будет содержать в себе все организации с указанным классом или названием в определенной области. Для каждой точки в этом JSON будет содержаться целый спектр информации (название, фактический адрес, координаты на Яндекс.Карта, сайт, номер телефона, время работы и т.д.). Здесь в основном интересны 3 элемента: класс, к которому принадлежит организация (продуктовый магазин, кафе, ТЦ и т.д.), координаты (долгота и широта), а также самое важное id (у каждой организации на карте есть свой определенный идентификационный номер, причем он разный даже в рамках точек, принадлежащих одной компании). ID в дальнейшем и будет выступать основным инструментом для сбора информации об отзывах, среднем рейтинге, а также количестве организаций. Таким образом, для каждой заданной области мы можем получить информации о точках, принадлежащих к определенным классам. Далее проделать эту же операцию для другой области и т.д. В таком случае мы получим большую базу данных из организаций, в которой сейчас условно есть 4 столбца (id, класс, широта и долгота). Затем необходимо спарсить информацию об отзывах и рейтингах с Яндекс.Карт [5]. Здесь и будут играть основную роль id, полученные ранее. После того, как парсинг выполнен исходная база данных расширилась на 2 столбца. Теперь можно произвести агрегацию данных о количестве организаций, количестве отзывов и рейтинге на основе id гексагонов и классов, к которым принадлежат объекты.

Постройки или объекты можно получить с помощью API OpenStreetMap. Однако, для упрощения взаимодействия с ним для Python существует библиотека OSMnx [6]. OSMnx – это пакет Python, который позволяет загружать геопространственные данные из OpenStreetMap, а также моделировать, проектировать, визуализировать и анализировать реальные уличные сети и любую другую геопространственную геометрию. Также есть возможность легко загружать и работать с различными типами инфраструктуры: достопримечательностями, коммерческими объектами, детскими садами и школами, улицами, контурами зданий и т.д.

Задача по сбору точечных данных о постройках в таком случае сводится к получению координат центров геометрий и расчету площади геометрий, информация о количестве квартир может формироваться напрямую из списков.

Когда все сырые географические данные собраны и обработаны, можно приступить к расчету значений финальных показателей для конкретных координат. Теперь достаточно лишь взять исходные координаты, задать для них определенный радиус области, в рамках которой будут агрегироваться значения факторов, получить эту геометрию. После чего осуществить ее заполнение гексагонами для формирования конечного списка уникальных идентификаторов, подпадающих под заданную область. Далее сформировать соответствующие запросы к базам данных. Для всех типов факторов, где данные проиндексированы, просто осуществить выборку информации по конкретным id. Для факторов с иной сеточной структурой проверить вхождение круговой геометрии в полигон, точку или

мультиполигон и, соответственно, получить данные, удовлетворяющие этому условию.

Пространственное отображение данных

Так как географические факторы – это большие данные, то не всегда удобно отсматривать результаты моделирования или же просто сырые переменные в формате таблиц, ибо отсутствует пространственное понимание их принадлежности. Так как любая точка или же область на карте имеет свои координаты или список координат, то достаточно, чтобы у исходных данных была лишь какая-то привязка к геометрии и это, в свою очередь, позволит отобразить значения признаков с их принадлежностью к конкретным местам.

Одним из инструментов для отображения пространственных данных на языке Python является библиотека kepler.gl [13]. Эта библиотека использует слои в качестве основных строительных блоков, позволяющих создавать интерактивные карты. В ней можно добавлять, удалять, отображать и скрывать слои, а также изменять их внешний вид.

Основными видами поддерживаемых геометрий являются линии, гексагоны, точки, полигоны, сетки, маршруты, иконки, кластеры. Визуализация GPS данных на примере Москвы представлена на рисунке 3.

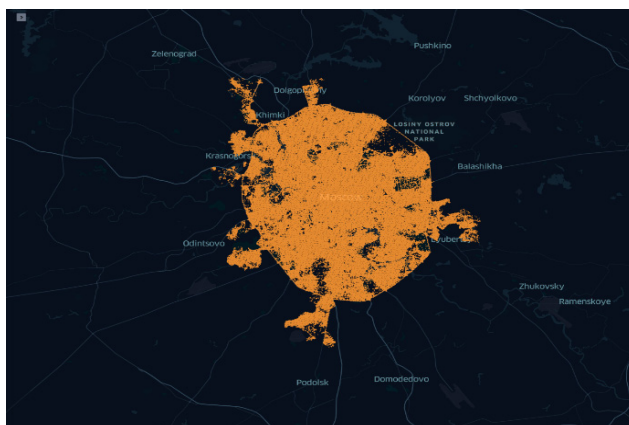


Рисунок 3 – Визуализация покрытия гексагонами Москвы на примере GPS

Если теперь взять какой-то конкретный вид данных по GPS. Например, жителей и сделать отображение по этому столбцу, то Kepler.gl автоматически сформирует слой в виде тепловой карты, где меньшим значения будут соответствовать более светлые тона, а большим более яркие. Тепловую карту по GPS жителям на темной подложке можно видеть на рисунке 4.

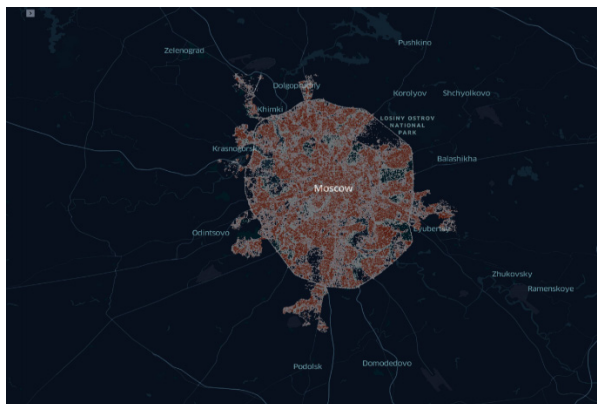


Рисунок 4 – Тепловая карта Москвы на примере GPS данных по жителям (темная подложка)

На рисунке 5 изображена аналогичная рисунку 4 карта за исключением лишь того, что здесь в качестве подложки выступает спутниковая схема.



Рисунок 5 – Тепловая карта Москвы на примере GPS данных по жителям (спутниковая подложка)

Если приблизить карту с рисунка 5, то можно более детально рассмотреть распределение жителей или любых других данных в масштабах небольших шестиугольников. Также при нажатии на любую такую геометрию будет всплывать список с информацией, которую также можно отдельно настроить (см. рисунок 6).

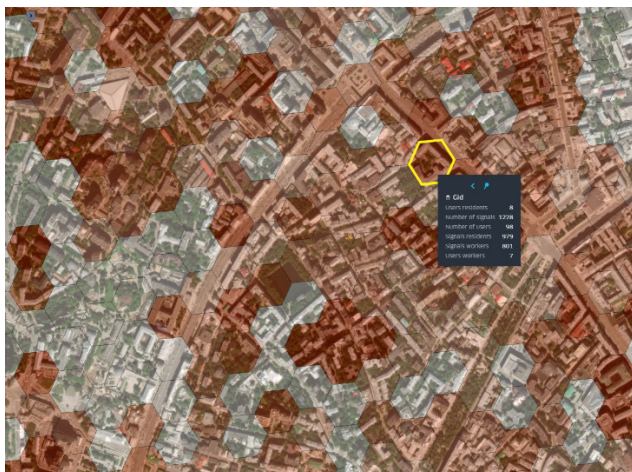


Рисунок 6 – Тепловая карта Москвы на примере GPS данных по жителям (близко)

Генерация дополнительных данных

Так одной из проверяемых гипотез в ходе создания моделей для оценки прибыльности локаций для магазинов конкретной организации может быть влияние доступности метро. Среди имеющихся данных такого фактора нет, поэтому можно создать его вручную. Для начала необходимо получить информацию о координатах всех станций метрополитена и выходах из них, в процессе ее поиска становится ясным, что такие данные присутствуют в открытом доступе, по крайней мере, по самым крупным городам России. После их сбора нужно оценить степень близости конкретной торговой точки ко всем имеющимся выходам из станций метро, а затем выбрать расстояние до того выхода, который окажется ближайшим. Эта задача лишь на первый взгляд может показаться простой, ибо не ясно какой инструмент для оценки расстояния окажется лучше. Например, одним из решений может являться просто нахождение векторного (евклидова) расстояния между точками. Однако, здесь возникает целый ряд недостатков. Во-первых, наше наикратчайшее расстояние, рассчитанное этим методом, будет просто прямой линией. Но в реальном мире не всегда ближайшая точка, найденная таким способом, будет в действительности ближайшей. Например, выход из станции

метро и магазин могут отделяться друг от другой рекой, дорогой, различными зданиями, постройками и т.п. Векторное расстояние – это наикратчайший путь только в условиях плоскости, но форма поверхности нашей планеты имеет изгибы. Для учета этого недостатка можно, например, воспользоваться, формулой Гаверсинуса, которая позволит учесть сферичность Земли. Однако, все равно никак не будет учитываться окружение вокруг. Именно поэтому для расчета расстояния между точками был выбран третий путь, основанный на использовании двух библиотек для Python. Одна из них – это OSMnx (подробнее об этом модуле говорилось в пункте «Алгоритмы сбора данных»), а другая – NetworkX [14]. Модуль NetworkX служит для генерации, взаимодействия и исследования структуры, динамики и функционирования сложных сетевых структур. Основными возможностями библиотеки являются:

- классы для работы с простыми, ориентированными и взвешенными графами, где узлом может быть практически что угодно: time-series, текст, изображение, XML;
- сохранение / загрузка графов в/из наиболее распространенных форматов файлов хранения графов;
- генерация графов базовых типов;
- методы для обнаружения подграфов, клик и K-дольных графов;
- получение таких характеристик графа как степени вершин, высота графа, диаметр, радиус, длинны путей, центр, промежуточности, и т. д.;
- визуализация сети в виде 2D и 3D графиков.

Например, для того, чтобы узнать наикратчайший маршрут между двумя точками необходимо выполнить следующий список действий:

- 1) инициализировать координаты широты (latitude) и долготы (longitude) начальной точки;
- 2) инициализировать координаты широты (latitude) и долготы (longitude) конечной точки;
- 3) получить граф из OSM в границах некоторой геокодированной местности. Для этого можно воспользоваться функцией `graph_from_place` библиотеки OSMnx. После чего на вход ей передать значения двух параметров `query` (строка с названием места, города, района и т.д.) и `network_type` (тип уличной сети, для которой будет рассчитываться маршрут (в нашем случае «walk»)).

4) найти ближайший узел к начальной точке. Реализуется через функцию `nearest_nodes` библиотеки OSMnx. В качестве аргументов нужно передать значения трех параметров: `G` – граф, в котором нужно найти ближайшие узлы (пункт 3), `X` – longitude из пункта 1, `Y` – latitude из пункта 1.

5) найти ближайший узел к конечной точке. Реализуется через функцию `nearest_nodes` библиотеки OSMnx. В качестве аргументов нужно передать значения трех параметров: `G` – граф, в котором нужно найти ближайшие узлы (пункт 3), `X` – longitude из пункта 2, `Y` – latitude из пункта 2.

6) рассчитать наикратчайшее расстояние можно с помощью функции `shortest_path_length` библиотеки NetworkX. На вход данной функции необходимо подать значения 4-х параметров: `G` – граф из пункта 3, `source` – узел из пункта 4, `target` – узел из пункта 5, `weight` – тип рассчитываемого расстояния (в нашем случае «length»).

Так для двух точек, находящихся в Москве, со следующими наборами адресов и координат: Красная площадь, Москва – 55.753544, 37.621202; Театральная площадь, 1, Москва – 55.759716, 37.618847. Наикратчайшим расстоянием будет являться 964,097 метра. Также одним из несомненных преимуществ библиотеки OSMnx является возможность визуализации маршрутов в статичном графовом представлении или же с помощью динамической карты `folium`. Подробнее на рисунках 7, 8.

Для того, чтобы иметь возможность отрисовать наикратчайший маршрут необходимо получить список узлов, через которые он будет проходить. Это можно сделать с помощью функции `shortest_path` библиотеки `NetworkX`. В качестве аргументов ей необходимо передать аналогичные значения из пункта 6.

Далее для получения динамической карты с рисунка 7 нужно в функцию `plot_route_folium` библиотеки `OSMnx` передать три аргумента: `G` – граф из пункта 3, `route` – список узлов наикратчайшего маршрута, получение которого описано ранее, `tiles` – наименование подложки (в нашем случае «`openstreetmap`»). Граф с рисунка 8 строится аналогичным образом через библиотеку `OSMnx`, однако в качестве функции используется `plot_graph_route` с параметрами `G` и `route`.

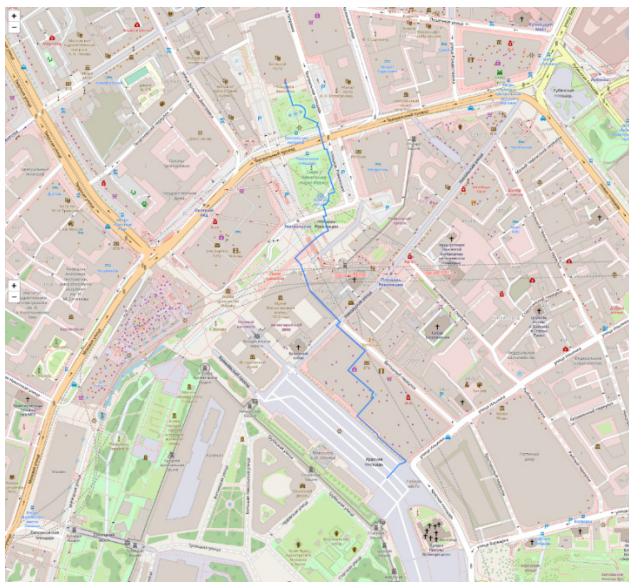


Рисунок 7 – Визуализация маршрута с помощью карты `folium`

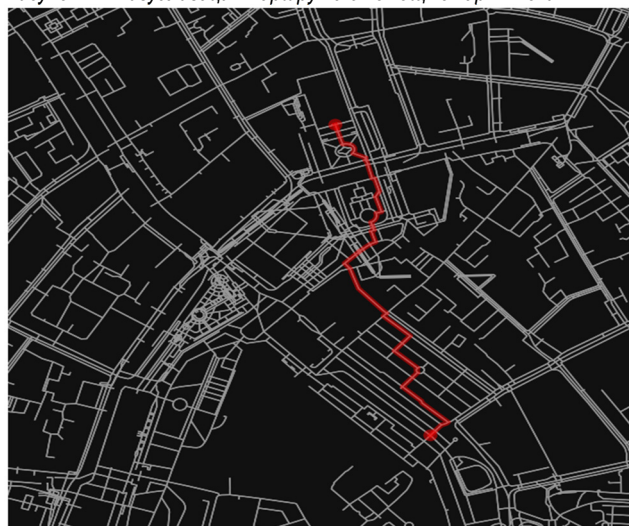


Рисунок 8 – Визуализация маршрута в статичном графовом представлении

Обсуждение результатов и выводы

Следует выделить, что, в рамках проведенного исследования:

1. Авторами был рассмотрен целый ряд понятий. А также даны определения для экономической эффективности и геоаналитики. Для такой относительно новой сферы науки, как геоаналитика, были представлены задачи, в решении которых

она используется и указаны авторы и работы, в которых освещался вопрос решения задач размещения.

2. Далее в работе были классифицированы и описаны 5 групп географических факторов, которые можно использовать для оценки экономического потенциала локации коммерческих организаций и их подразделений. Сами группы представляют из себя совокупности признаков, объединенные по принципу источников, из которых они были получены. А именно: мобильные данные (предоставляются сотовыми операторами); GPS данные (предоставляются агрегаторами информации по мобильным приложениям); данные о недвижимости (собираются с сайта «ЦИАН»); данные о бизнесе (собираются через API сервис Яндекс.Карт); данные о постройках (собираются с сайта «OpenStreetMap»).

3. Авторами была предложена методологическая база по работе с географическими данными и разобраны теоретические аспекты для решения задач размещения. Главными инструментами на этом этапе стали сеточная структура, представленная в виде системы иерархических индексов и реализованная компанией Uber в рамках библиотеки `H3` для языка программирования `Python` и гексагоны, геометрии шестиугольной формы, до которых можно агрегировать точечные пространственные данные. После чего в исследовании описаны основные алгоритмы сбора географических данных, которые применялись в этом исследовании для каждой из пяти групп, о которых говорилось ранее. В качестве основных инструментов для получения информации использовались: сайт недвижимости «ЦИАН»; Яндекс.API; Яндекс.Карты; API `OpenStreetMap`, а также модули `pandas`, `requests`, `selenium`, `h3`, `osmnx`, `networkx`.

4. К результатам представленного исследования следует отнести (программную) реализацию алгоритмов обработки географических данных, которые, по мнению авторов, в дальнейшем могут быть использованы в процессе построения модели для оценки потенциала локаций коммерческих организаций и их подразделений. В заключительной части также был показан процесс визуализации части полученных данных.

Литература

1. Еланцев С.В. Проблемы роста эффективности корпоративного сектора российской экономики // Вестник Шадринского государственного педагогического института. – 2013. – № 4 (20). – 143–146 с.
2. Самуэльсон П., Нордхаус У. Экономика. – М.: Вильямс, 2014. – 1360 с.
3. API Поиска по организациям // Яндекс / Электронный ресурс – Режим доступа: <https://yandex.ru/dev/maps/geosearch/doc/concepts/about.html?from=geosearch> (Дата обращения: 20.09.2022)
4. ЦИАН // CIAN / Электронный ресурс – Режим доступа: <https://www.cian.ru/> (Дата обращения: 20.09.2022)
5. Яндекс Карты // Яндекс / Электронный ресурс – Режим доступа: <https://yandex.ru/maps> (Дата обращения: 20.09.2022)
6. Boeing, G. (2017). OSMnx: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks. *Computers, Environment and Urban Systems*, 65, p. 126–139.
7. Cabral, Luis M. B. (2000). Introduction to industrial organization. Cambridge, UK: MIT Press. – p. 354.
8. Ghosh, D. (2008). A Loose Coupling Technique for Integrating GIS and Multi-Criteria Decision Making. *Transactions in GIS*, 12 (3), p. 365–375.
9. Harris, R.J., Sleight, P., & Webber, R. (2005). *Geodemographics, GIS and Neighbourhood Targeting*. Chichester, UK: John Wiley & Sons. – p. 288.
10. H3: Uber's Hexagonal Hierarchical Spatial Index // Uber / Электронный ресурс – Режим доступа:

<https://www.uber.com/en-IN/blog/h3/> (Дата обращения: 20.09.2022)

11. h3-py: Uber's H3 Hexagonal Hierarchical Geospatial Indexing System in Python // Uber.Github / Электронный ресурс – Режим доступа: <https://uber.github.io/h3-py/intro.html> (Дата обращения: 20.09.2022)

12. Jankowski, P. (1995). Integrating geographical information systems and multiple criteria decision-making methods. *International Journal of Geographical Information Systems*, 9 (3), p. 251–273.

13. Kepler.gl // GitHub / Электронный ресурс – Режим доступа: <https://github.com/kepler.gl/kepler.gl> (Дата обращения: 20.09.2022)

14. NetworkX: Network Analysis in Python // NetworkX / Электронный ресурс – Режим доступа: <https://networkx.org/> (Дата обращения: 20.09.2022)

15. Tables of Cell Statistics Across Resolutions // H3Geo / Электронный ресурс – Режим доступа: <https://h3geo.org/docs/core-library/restable/> (Дата обращения: 20.09.2022)

16. Uber H3 for Data Analysis with Python // Towards Data Science / Электронный ресурс – Режим доступа: <https://towardsdatascience.com/uber-h3-for-data-analysis-with-python-1e54acdccc908> (Дата обращения: 20.09.2022)

17. Weber, P., & Chapman, D. (2011). Location Intelligence: An Innovative Approach to Business Location Decision-making. *Transactions in GIS*, 15 (3), p. 309–328.

Geoanalytics: collection, analysis and display of geographical data for solving problems of location of commercial organizations

Grineva N.V., Topyrkin A.D.

Financial University under the Government of the Russian Federation

JEL classification: C01, C02, C1, C4, C5, C6, C8

The relevance of the article lies in the formalization of the process of collecting geographic data to solve the problem of locating commercial organizations and their divisions. The aim of the study is to solve the problem of location and assess the degree of influence of geographical characteristics of locations on the economic efficiency of a commercial organization. The article defines the concepts of economic efficiency and geoanalytics, as well as how they are related. A number of tasks for the solution of which geoanalytics is used are described. Questions regarding the geographic data used are covered in detail, namely, answers to questions are given. What? How? Where? What - what factors can be used to identify the potential of a location, how this data can be aggregated and stored, where it can be directly collected. The authors note that the main tools for solving such problems of mathematical modeling and data analysis are the Python programming language and its libraries pandas, requests, selenium, h3, osmnx, networkx. The process of constructing routes between two points within the framework of the problem of finding the shortest distance is proposed. In conclusion, one of the possible ways to display spatial data using the kepler.gl module is demonstrated. The result of the work is the collected geographical factors, which will be further used in the process of building a model to determine the economic potential of the location of commercial organizations and their divisions.

Keywords: geoanalytics, mathematical modeling, data analysis, economic efficiency, hexagons, Python programming language.

References

1. Elantsev S.V. Problems of increasing the efficiency of the corporate sector of the Russian economy // *Bulletin of the Shadrinsk State Pedagogical Institute*. - 2013. - No. 4 (20). - 143–146 p.
2. Samuelson P., Nordhaus U. *Economics*. – M.: Williams, 2014. – 1360 p.
3. Search API for organizations // Yandex / Electronic resource - Access mode: <https://yandex.ru/dev/maps/geosearch/doc/concepts/about.html?from=geosearch> (Date of access: 20.09.2022)
4. CIAN // CIAN / Electronic resource - Access mode: <https://www.cian.ru/> (Date of access: 20.09.2022)
5. Yandex Maps // Yandex / Electronic resource - Access mode: <https://yandex.ru/maps> (Date of access: 20.09.2022)
6. Boeing, G. (2017). OSMnx: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks. *Computers, Environment and Urban Systems*, 65, p. 126–139.
7. Cabral, Luis M. B. (2000). *Introduction to industrial organization*. Cambridge, UK: MIT Press. – p. 354.
8. Ghosh, D. (2008). A Loose Coupling Technique for Integrating GIS and Multi-Criteria Decision Making. *Transactions in GIS*, 12(3), p. 365–375.
9. Harris, R.J., Sleight, P., & Webber, R. (2005). *Geodemographics, GIS and Neighborhood Targeting*. Chichester, UK: John Wiley & Sons. – p. 288.
10. H3: Uber's Hexagonal Hierarchical Spatial Index // Uber / Electronic resource – Access mode: <https://www.uber.com/en-IN/blog/h3/> (Date of access: 20.09.2022)
11. h3-py: Uber's H3 Hexagonal Hierarchical Geospatial Indexing System in Python // Uber.Github / Electronic resource - Access mode: <https://uber.github.io/h3-py/intro.html> (Date of access: 20.09.2022)
12. Jankowski, P. (1995). Integrating geographical information systems and multiple criteria decision-making methods. *International Journal of Geographical Information Systems*, 9(3), p. 251–273.
13. Kepler.gl // GitHub / Electronic resource - Access mode: <https://github.com/kepler.gl/kepler.gl> (Date of access: 09/20/2022)
14. NetworkX: Network Analysis in Python // NetworkX / Electronic resource - Access mode: <https://networkx.org/> (Date of access: 09/20/2022)
15. Tables of Cell Statistics Across Resolutions // H3Geo / Electronic resource - Access mode: <https://h3geo.org/docs/core-library/restable/> (Date of access: 09/20/2022)
16. Uber H3 for Data Analysis with Python // Towards Data Science / Electronic resource - Access mode: <https://towardsdatascience.com/uber-h3-for-data-analysis-with-python-1e54acdccc908> (Date of access: 20.09.2022)
17. Weber, P., & Chapman, D. (2011). Location Intelligence: An Innovative Approach to Business Location Decision-making. *Transactions in GIS*, 15(3), p. 309–328.