Санкт–Петербургский государственный университет

Хачатрян Григор Гагикович

Выпускная квалификационная работа

Разработка методов интеллектуального анализа данных о курортных предпочтениях пользователей социальной сети Instagram

Уровень образования: бакалавриат
Направление 02.03.02
«Фундаментальные информатика и информационные технологии»
Основная образовательная программа СВ.5003.2016
«Программирование и информационные технологи»

Научный руководитель:
профессор кафедры математического
моделирования энергетических систем,
д.ф. - м.н. Крылатов Александр Юрьевич

Рецензент:
доцент кафедры технологии программирования,
кандидат технических наук
Блеканов Иван Станиславович

Санкт-Петербург
2020 г.

Содержание

Введение

В настоящее время индустрия туризма является одной из наиболее динамично развивающихся сфер в мире. Популярность туризма с каждым годом растет, как и потребность людей в новых путешествиях и ощущениях.

Каждый человек, планируя отпуск, сталкивается с одной из основных проблем: куда именно отправиться в этот раз. Данная проблема имеет несколько причин.

Во-первых, красивых стран и мест - большое множество, поэтому не всегда удается быстро определиться с выбором, чтобы он соответствовал всем ожиданиям.

Во-вторых, опытным туристам, побывавшим во многих местах и повидавшим мир, сложнее выбрать новый пункт назначения.

На данный момент уже имеется несколько способов решения данной задачи, но привычные для нас подходы с каждым годом устаревают. В эпоху технологий и развития социальных сетей большая часть информации о человеке и его предпочтениях находится в интернете.

На 2019-2020 год социальная сеть Instagram входит в десятку популярнейших платформ по всему миру и в первую тройку по России. Instagram за время своего существования из приложения для обмена фотографиями и видеозаписями превратился в систему практически точно характеризующую любого ее пользователя, его пожелания и предпочтения. Это касается и туристических взглядов человека.

В данной работе поставлена цель найти метод решения этой задачи посредством анализа страниц пользователей социальной сети Instagram и разработать общедоступный программный продукт, который сможет рекомендовать страны отдыха для определенного человека, у которого имеется активная страница в Instagram.

При разработке данного метода были использованы данные с веб-приложения Instagram: собирался датасет из 50 000 пользователей с открытыми аккаунтами и минимум с 5 публикациями из разных стран. По-

лученные данные обрабатывались языком программирования Python и использовались в качестве обучающих данных.

С помощью полученной выборки, программа выполняет заложенные в нее алгоритмы анализа, а именно определяет эмоциональную окраску каждого поста и кластеризует пользователей по курортным предпочтениям.

Используя готовые модели кластеризации и определения тональности постов, программа способна обрабатывать новых пользователей, определять, к какому кластеру они относятся. Будем считать, что в каждом кластере находятся пользователи с одинаковыми курортными предпочтениями. После нахождения группы, к которой принадлежит пользователь, программа выдает список популярных в данном кластере стран, которые еще не посещал пользователь.

Постановка задачи

Основной задачей данной работы является создание программного обеспечения и инструментария для обработки и анализа данных пользователя социальной сети Instagram.

Цель анализа состоит в выявлении курортных предпочтений на основе публикаций с геолокациями посещенных мест отдыха.

Основная задача делится на следующие подзадачи:

1. Сбор данных о 50 000 пользователей Instagram без использования стандартного API.

2. Предобработка полученных данных, которая включает в себя удаление пустых аккаунтов и постов.

3. Сбор размеченных данных с сайта Tripadvisor, а именно получение 500 000 отзывов с оценками о различных местах отдыха. Данные понадобятся для обучения нейронной сети.

4. Создание и обучение нейронной сети, определение тональности текста публикаций, разделение на 5 классов: отлично (т.е. очень понравилось); хорошо (ничего плохого нет, но и не отлично); нейтрально (посты информационного характера); плохо (в целом, не понравилось, но есть положительные моменты); отвратительно (агрессивные посты с призывом не посещать данное место).

5. Кластеризация 50 000 пользователей по курортным предпочтениям.

После решения всех подзадач мы можем обрабатывать новых пользователей, выявлять их курортные предпочтения и рекомендовать страны отдыха, в которых им с большей вероятностью понравится.

Обзор литературы

1. Петин. В.А., “API Яндекс, Google и других популярных веб-сервисов.”

   В данной книге рассказывается о таком инструменте, как API. Описывается взаимодействие с популярными веб-сервисами. Особое внимание уделяется обработке полученных данных.

2. Satya Avasarala, “Selenium WebDriver Practical Guide.”

   В данной книге описываются возможности WebDriver Selenium, с помощью которого производится автоматизированное тестирование. Были изучены технологии имитации действий пользователя.

3. Harry J.W. Percival, “Test-Driven Web Development with Python.”

   Книга посвящена созданию веб-приложения на языке Python. В процессе изучены основы Django, Selenium, Git, а также современные методы веб-разработки.

4. Бен Хеник, "HTML и CSS путь к совершенству".

   С помощью данной книги были изучены основы front end разработки сайтов, инструменты для придания сайту адаптивности и методы визуального отображения данных.

5. Охеда Тони, Билбро Ребекка, Бенгфорт Бенджамин, “Прикладной анализ текстовых данных на Python”.

   Данная книга является прикладной и рассказывает о применении методов машинного обучения для анализа текста. Из данной книги взяты основные методы определения тональности текста.

6. Официальный сайт документации API Instagram.

   С помощью данной документации были изучены возможности сервера Instagram, а также способы взаимодействия с ним. После прочтения было принято решение использовать нестандартное API. С помощью инструментов Instagram собирается вся необходимая информация о пользователях.

Глава 1. Основные понятия

1.1 Взаимодействие с Instagram

API — интерфейс прикладного программирования (интерфейс программирования приложений), т. е. набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) для использования во внешних программных продуктах [1].

API Instagram Basic Display предоставляет доступ к основной информации страницы пользователя.

Стандартное использование:

- Получение маркера доступа пользователя Instagram и разрешений от пользователя Instagram.
- Получение профиля пользователя Instagram.
- Получение изображений, видеозаписей и альбомов пользователя Instagram [6].

Стандартное использование не подходит для решения поставленной задачи, так как требуется собрать информацию о 50 000 страниц без использования маркера доступа от пользователей.

В связи с данной проблемой возникает необходимость получить данные, не используя возможности стандартного API. Это можно осуществить с помощью специальных запросов.

Запрос, отправляемый на сервер Instagram, выглядит следующим образом:

https://www.instagram.com/graphql/query/?query_hash=QUERY_HASH&
variables={“id”: USER_ID, ”first”: COUNT , “after”: END_CURSOR},

где QUERY_HASH - захэшированный запрос;
USER_ID - идентификатор пользователя;
COUNT - количество публикаций, которые необходимо получить (максимальное количество - 50);

END_CURSOR - идентификатор следующей страницы с публикациями пользователя.

Данные по этому запросу доступны только реальным пользователям, которые заходят в Instagram через браузер или приложение, поэтому необходимо имитировать действия пользователя. Самым эффективным способом имитации является добавление в запрос специального заголовка.

Для наглядности приведем пример программной имитации пользователя:

header = {'User-Agent': USER, 'X-Requested-With': REQUESTED},

где USER - строка с названием, версией веб-приложения, операционной системой компьютера и языком;
REQUESTED - нестандартный заголовок, который сообщает средство запроса.

В ответ на запрос от сервера приходит следующая информация о странице пользователя: количество публикаций и их идентификаторы.

Для наглядности на рис. 1 приведен фрагмент ответа от сервера, где соответствующие параметры принимают следующие значения:
QUERY_HASH = 472f257a40c653c64c666ce877d59d2b;
USER_ID = 3170954593;
COUNT = 1;
END_CURSOR - отсутствует.

После получения всех идентификаторов публикаций пользователя необходимо собрать информацию о каждом из постов, в котором указаны геолокация и текст.

Запрос отправляемый на сервер для получения данных выглядит следующим образом:

https://www.instagram.com/p/SHORT_CODE/?__a=1,

где SHORT_CODE - идентификатор публикации.

Ответ от сервера приходит так же в JSON формате и содержит всю информацию о публикации (фотография, комментарии, геолокация, текст,

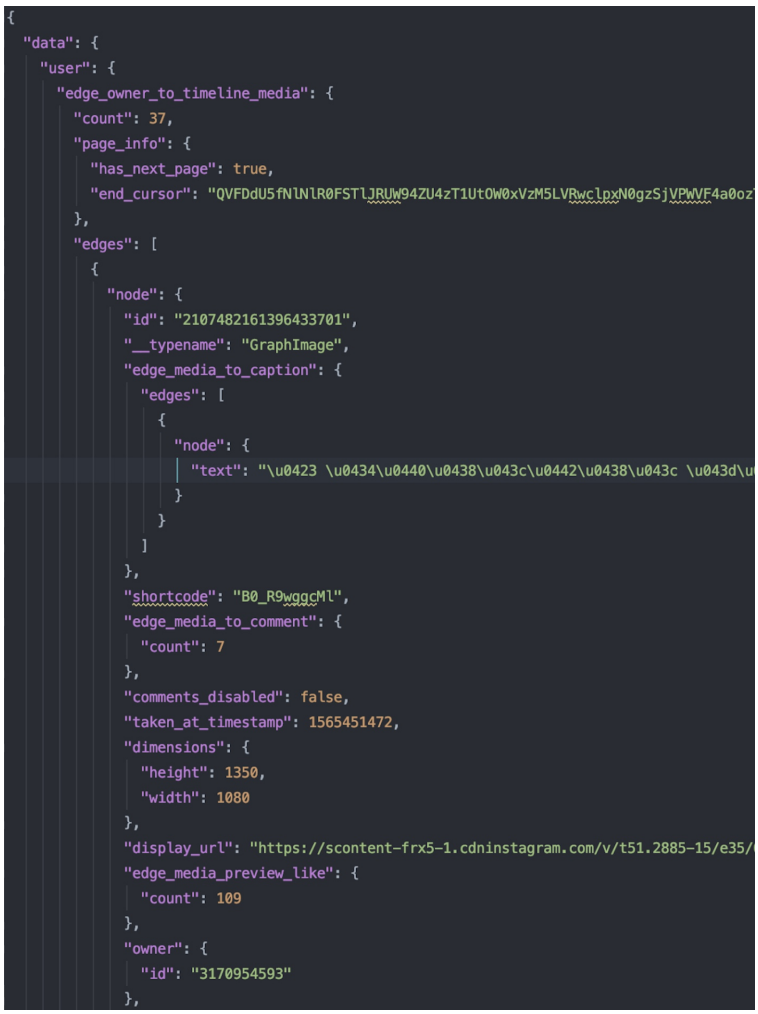

Рис. 1: JSON ответ от сервера на запрос о количестве публикаций и их идентификаторов

количество лайков), полученные данные необходимо обработать и сохранить только необходимое (геолокация и текст) для дальнейшего анализа.

1.2 Анализ тональности текста

Анализ тональности текста - класс методов, который предназначен для определения эмоционально окрашенной лексики и оценки автора текста.

В данной работе используется метод на основе машинного обучения с учителем.

Данный метод делится на несколько этапов.

1.2.1 Сбор данных, на которых обучается машинный классификатор

Задача заключается в определении эмоциональной окраски постов. Будем считать, что каждая публикация пользователя с геолокацией является его отзывом о данной стране. Чтобы обучить систему определять тональность постов, необходимо собрать размеченный датасет отзывов туристов об отдыхе и достопримечательностях с другой платформы.

Сбор требуемых данных осуществляется с помощью веб-сервиса Tripadvisor, на котором находятся уже размеченные отзывы (т.е. отзывы с оценками).

Для наглядности на рис. 2 изображен пример данных, расположенных на сайте.

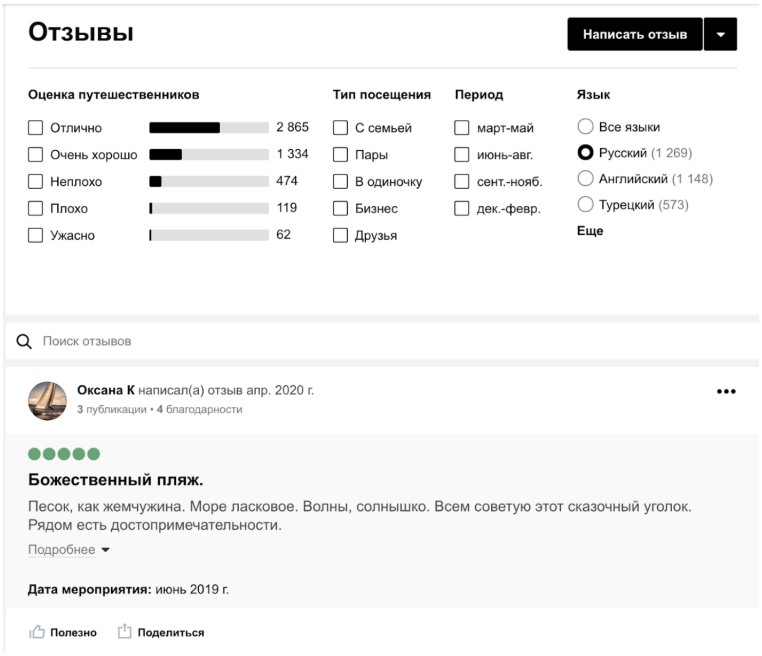

Рис. 2: Пример отзыва на сайте Tripadvisor

На данном этапе возникает следующая трудность: веб-приложение Tripadvisor не позволяет получать отзывы с помощью встроенного API. В связи с этим возникает потребность в использовании инструмента для тестирования Selenium, который имитирует действия пользователя, способен переходить по ссылкам и нажимать на кнопки. С его помощью собираются HTML-документы с необходимым набором отзывов. Далее документы об-

рабатываются посредством Python, а отзывы и оценки сохраняются в базу данных.

Для дальнейшей работы полученные данные необходимо обработать и сохранить в csv формате. Для этого нужно выполнить следующую последовательность действий:

1. Удалить лишние знаки в тексте и преобразовать все слова в начальную форму.

2. Из составленного набора слов убрать “стоп-слова” (т. е. слова, которые не несут сильной смысловой нагрузки).

3. Преобразованные отзывы с оценками сохранить в csv файл.

На основе этих данных создается и обучается токенизатор, который в свою очередь тоже сохраняется в отдельных файл.

Токенизатор - это инструмент для автоматического разделения текста на токены (слова). С его помощью находятся наиболее встречающиеся слова в тексте и сохраняются в отдельный файл. Количество слов зависит от входного параметра, который задается при создании токенизатора. Такой подход позволит представлять отзывы в виде массива с числами, где каждому числу соответствует определенное слово.

1.2.2 Выбор нейронной сети

В качестве нейронной сети для определения эмоциональной окраски постов пользователей была выбрана LSTM сеть.

LSTM (Long short-term memory) - один из видов рекуррентных нейронных сетей, хорошо приспособленных к обучению на задачах классификации.

Для наглядности на рис. 3 приведен пример структуры сети. На схеме видно, что рекуррентная нейронная сеть имеет форму цепи повторяющихся модулей.

Помимо слоя LSTM модель имеет:

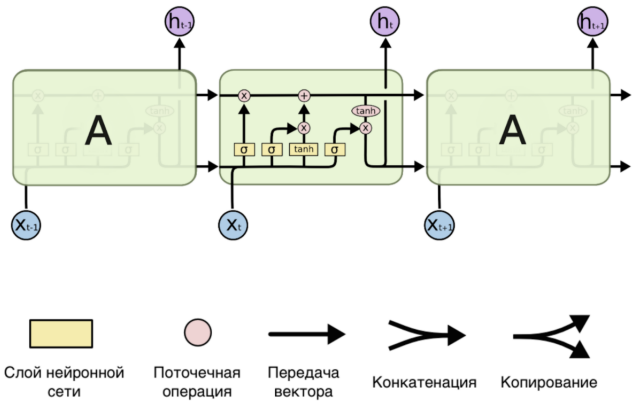

Рис. 3: Структура сети LSTM

1. Слой Embedded, который позволяет создать плотную кодировку слов. На вход данный слой получает количество слов, длину создаваемого вектора слова и количество слов в каждом посте. Обычно используется в качестве первого слоя в модели.

2. Слой Dense с 5 нейронами, которые отвечают за каждый класс тональности, и функцией активации “softmax”. Данная функция преобразует полученный вектор в вектор такой же размерности, где каждая координата представлена в виде вещественного числа на интервале [0,1] и с суммой координат равной 1, то есть, на выходе получаются вероятности принадлежности поста к каждому из классов тональности.

После выбора нейронной сети и ее создания необходима компиляция, чтобы настроить модель для обучения. Данный метод имеет несколько аргументов:

1. optimizer - используемый оптимизатор;

2. loss - функция потери;

3. metrics - список метрик, которые будут оцениваться моделью во время обучения и тестирования.

   Для решения данной задачи используются:

- Эффективный алгоритм оптимизации adam (Adaptive Moment Estimation). Adam - это алгоритм оптимизации, который можно использовать вместо классической процедуры стохастического градиентного спуска для итеративного обновления весов сети на основе обучающих данных.

- Функция потерь - категориальная кросс-энтропия, которая является мультиклассовой и берет сумму значений логарифмических функций потерь каждого прогноза наблюдаемых классов. Задача обучения заключается в минимизации данной функции.

- В качестве метрики - accuracy, доля правильных ответов алгоритма.

Полученную модель необходимо обучить на подготовленных данных, которые разделены на две группы: обучающая и проверочная. Обучение модели будет проходить в 10 эпох, то есть, весь датасет пройдет через нейронную сеть в прямом и обратном направлении 10 раз. После каждой эпохи проверяется качество работы на проверочном наборе данных. Если оно улучшилось, то данная сеть сохраняется в файл.

На рис. 4 показано качество работы на обучающих и проверочных данных.

После каждой эпохи система выводит 4 показателя: loss - показатель функции потерь на обучающей выборке, accuracy - точность на обучающей выборке, val_loss - показатель функции потерь на проверочной выборке, val_accuracy - точность на проверочной выборке. На рис. 4 видно, что качество работы было наилучшим на 6 эпохе, где val_accuracy равен 0.70957, следовательно, программа сохранила данную сеть.

Следующим этапом является оценка работы сети на тестовом наборе данных, который не участвовал ранее в обучении модели. На данном этапе сеть показала хороший результат, где параметр accuracy принял значение 0.7085.

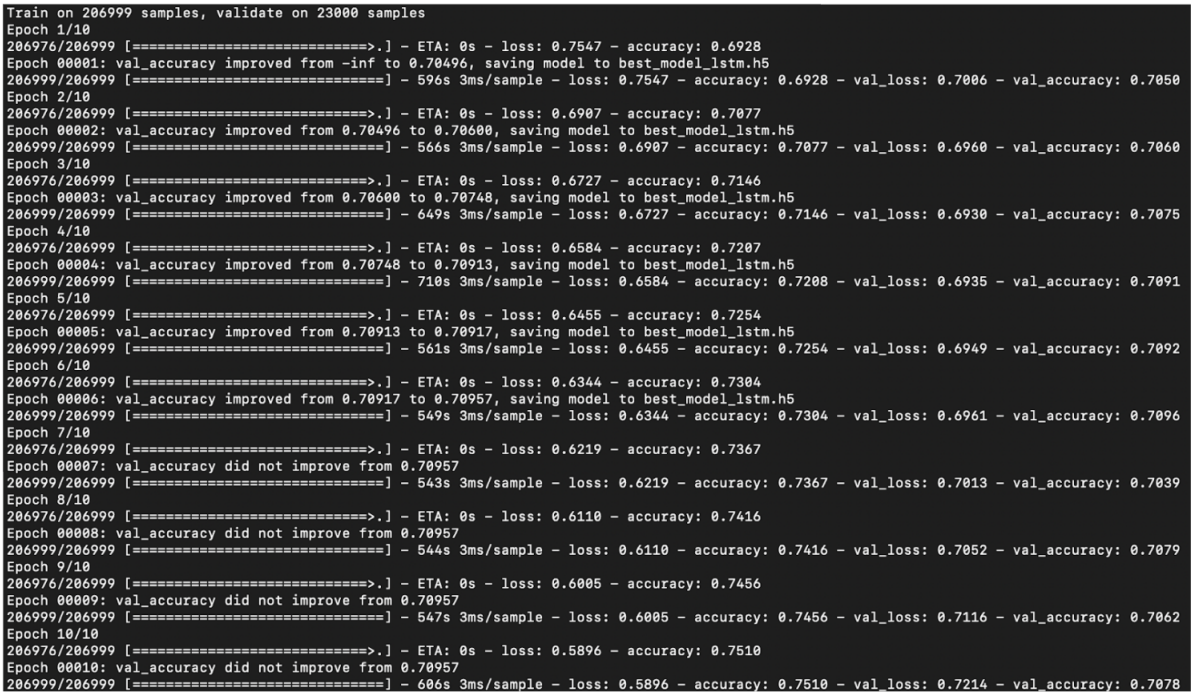

Рис. 4: Качество работы нейронной сети

1.2.3 Использование полученной модели для определения тональности постов пользователей

Для определения эмоциональной окраски текста необходимо вначале загрузить сохраненную сеть и обученный токенизатор. После получения всех необходимых инструментов нужно удалить из текста знаки препинания и стоп-слова, а также привести каждое из слов в начальную форму. Обработанный пост пользователя в виде вектора передается на вход нейронной сети, которая в свою очередь возвращает вектор, состоящий из 5 чисел на интервале [0,1], которые показывают вероятность принадлежности текста к каждому из наших эмоциональных классов.

Для наглядности далее продемонстрирована работа программы, входными данными которой являются 3 отзыва: отвратительный, нейтральный, отличный.

1. Отвратительный отзыв.

   “This journey was awful. I liked nothing. The hotel was disgusting, the villagers were really rude. I hate it. I ate some food in the local restaurant. So there was a fly in my soup! I have never been in such a bad trip!”

Отработанная программа выдает искомый вектор:

[0.877991 0.05526818 0.0492589 0.00945517 0.00802669]

Из полученного вектора можно сделать вывод, что модель корректно определяет плохие отзывы, так как вероятность принадлежности значительно больше у класса 1, который отвечает за отвратительные отзывы.

2. Нейтральный отзыв.

   “This journey was normal. It wasn’t very good, but it wasn’t bad too. I liked some things there, but I also hate my room in the hotel. If it is okay for you that the hotel has three stars than this is what you want.”

   Полученные результаты:

   [0.09393866 0.15703937 0.44787294 0.17399909 0.1271499]

   Можно заметить, что нейронная сеть правильно определила класс, к которому принадлежит текст, но с меньшей вероятностью принадлежности, чем в 1 случае.

3. Отличный отзыв.

   “This journey was amazing. I liked everything! The hotel was so great, the villagers were really nice. I love them. I ate some food in the local restaurant. So I have never eaten so delicious food before! I will come back soon!”

   Полученный результат:

   [1.8745090e-04 1.3963127e-04 1.9476451e-03 0.053849522 0.94387573]

   Нейронная сеть, с большой вероятностью принадлежности, верно определила класс, к которому принадлежит отзыв.

1.2.4 Итог

В ходе данного эксперимента тестируется работа обученной модели на настоящих данных из социальной сети Instagram. Полученные данные помогают сделать вывод, что нейронная сеть корректно определяет эмоциональную окраску постов пользователей и что с крайними классами (отвратительными и отличными отзывами) система работает лучше. Это связано с тем, что в обучающей выборке плохие отзывы в какой-то степени похожи на нейтральные, а нейтральные отзывы в свою очередь схожи с хорошими.

С помощью полученной нейронной сети обрабатывается каждой пост пользователей и определяется к какому классу тональности относится та или иная публикация. Полученные данные сохраняются для дальнейшей кластеризации.

1.3 Кластеризация пользователей Instagram

Самый популярный метод кластеризации k-means - это метод, задача которого заключается в разбиении множества объектов M на группы (кластеры), где каждый объект относится к тому кластеру, к центроиду (точке, являющейся центром кластера) которого он ближе всего.

В качестве меры близости используется Евклидово расстояние:

$$\mathrm{b}_{ij} = \sqrt{\sum_{k=1}^{m}(x_{ij} - x_{jk})^2}$$

Для кластеризации пользователей по курортным предпочтениям применяется данный метод, так как он прост в использовании и эффективен.

Вначале необходимо обработать имеющиеся данные о пользователях и для каждого из них подсчитать среднюю оценку отзывов в различных странах (если пользователь не побывал в какой-то стране, то выставляется оценка равная 0). Далее полученные данные выгружаются в матрицу, где столбцами являются названия стран, а строками - идентификаторы пользователей.

Метод кластеризации k-means требует указания количества кластеров, на которое необходимо поделить данные. Для определения числа групп можно воспользоваться методом “локтя” (elbow method), который основан

на отображении графика качества модели с изменением количества кластеров. Выбор оптимального числа центров определяется наличием “локтевого сгиба” на графике.

Для наглядности на рис. 5 приведен полученный график.

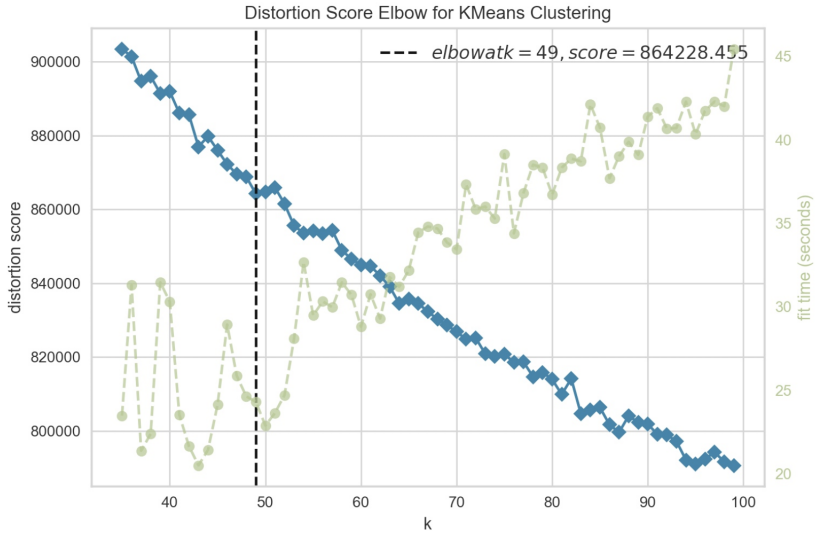

Рис. 5: График качества работы метода с изменением количества кластеров

Данный метод показывает оптимальное количество кластеров, равное 49.

Значит, запускаем кластеризацию на 49 групп. С помощью этого алгоритма искомые 50 000 пользователей Instagram разбиваются на различные группы. Информация о каждом кластере сохраняется в два файла: в одном хранятся данные о том, к какой группе относится тот или иной пользователь, в другом - центроиды самих кластеров, с помощью которых в дальнейшем можно посчитать евклидово расстояние и присвоить нового пользователя к какой-нибудь группе.

Глава 2. Метод определения курортных предпочтений пользователя Instagram на примере

Рассмотрим применение метода на конкретном примере пользователя. Будем анализировать страницу человека с логином “orientalnastya”. На рис. 6 представлена страница данного пользователя в социальной сети Instagram.

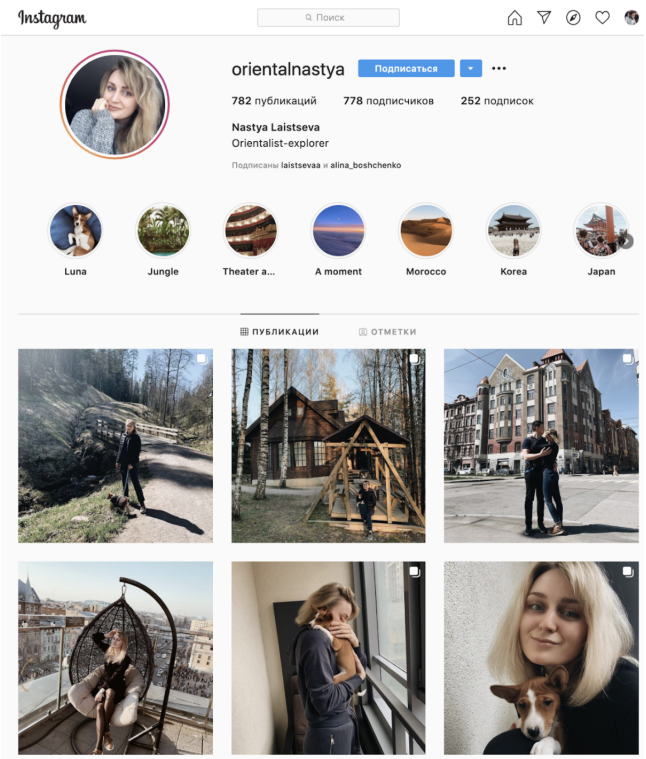

Рис. 6: Страница пользователя “orientalnastya”

Для начала получим публикации с геолокациями и определим тональность постов.

Сумма оценок по странам и их количество показаны на рис. 7.

Далее посчитаем среднюю оценку каждой страны и создадим матрицу, состоящую из одной строки и 255 столбцов (стран). Проставим полученные оценки в соответствии с посещенными странами (если пользователь не бывал в какой-то из стран, то оценка равна 0). С помощью полученной матрицы добавим пользователя в схожий по курортным предпочтениям кластер. Теперь можем проанализировать данные в искомом кластере, вы-

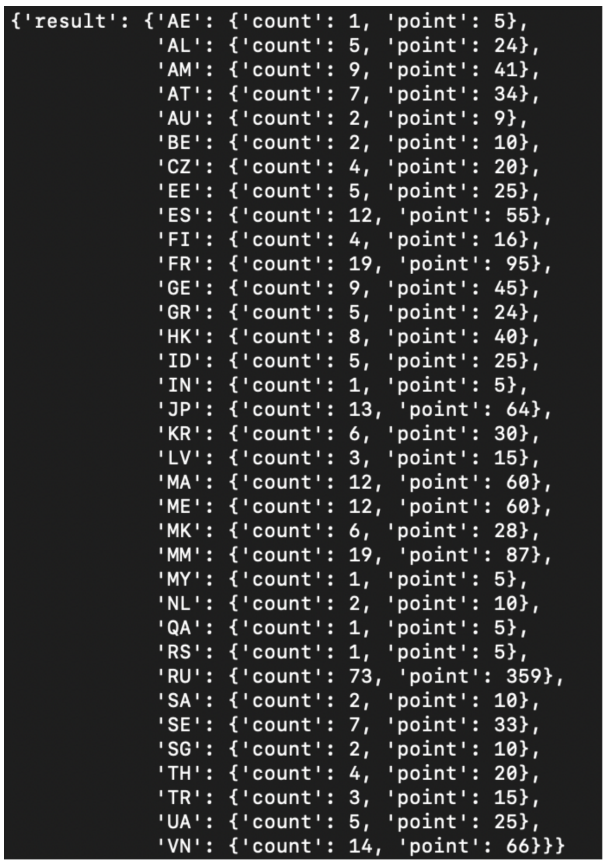

Рис. 7: Страны, в которых побывал пользователь, и выставленные оценки

явить и составить список стран, понравившихся большинству пользователей, которые попали в тот же кластер. Далее пользователю будем рекомендовать страны, ранее не посещенные им, из полученного списка. Полученный список стран для пользователя “orientalnastya” представлен в таблице 1.

Таблица 1: Рекомендованные страны.

Логин пользователя	Страны
orientalnastya	Италия, США, Германия, Великобритания, Камбоджа

Глава 3. Тестирование метода

3.1 Проведение опроса и сбор данных

Для тестирования качества работы метода был проведен опрос среди пользователей с открытой страницей в Instagram. В опросе приняло участие 48 человек. Им необходимо было указать 5-10 стран, в которых они еще не побывали, но в дальнейшем хотели бы побывать. В таблице 2 показана часть результатов опроса.

Таблица 2: Результаты проведенного опроса

Логин в Instagram	Страны
ms_cehanovich	США, Аргентина, Великобритания, Швейцария, Греция, Италия, Испания, Португалия, Япония, Армения
annushkaa.p	США, Канада, Япония, Португалия, Занзибар, Мексика
lexayat	Канада, Австралия, Великобритания, Чили, Украина
a.gogotov256	Швеция, Германия, Чехия, Эстония, Латвия, Армения, Марокко
nicks.ksks	Япония, Франция, США, Австралия, Великобритания, Швейцария, Чили, Италия
avdeev1van	США, Канада, Швейцария, Грузия, Армения, Дания, Норвегия, Исландия, Англия
andrew_grape	Беларусь, Украина, Казахстан, Швеция, Нидерланды, Австрия, Венгрия, Испания, Мьянма, США
ikono_stas	Армения, Испания, Япония, Австралия, США, Англия, Греция

savnastya	США, Канада, Испания, Япониия, Бразилия, Куба, Мексика, Таиланд
cornul11	Дания, США, Япония, Китай, Исландия

3.2 Результаты работы метода

Воспользуемся созданным методом, для определения курортных предпочтений пользователей, которые поучаствовали в опросе. Будем считать, что метод работает корректно, если в результате программа выдает для каждого пользователя хотя бы одну страну, которую он указал в опросе. В таблице 3 указана часть результатов, которые выдает программа.

Таблица 3: Результаты работы метода для пользователей из опроса

Логин в Instagram	Страны
ms_cehanovich	Швеция, США, Португалия, Филиппины, Малайзия
annushkaa.p	Италия, Португалия, Швейцария, Великобритания, США
lexayat	США, Канада, Филиппины, Греция, Тайвань
a.gogotov256	Швеция, США, Португалия, Филиппины, Малайзия
nicks.ksks	США, Швейцария, Швеция, Великобритания, Италия
avdeev1van	Швеция, США, Португалия, Филиппины, Малайзия
andrew_grape	США, Тайланд, Китай, Канада, Нидерланды
ikono_stas	Швеция, США, Португалия, Филиппины, Малайзия

savnastya	Канада, Мексика, Индонезия, Великобритания, Таиланд
cornul11	Великобритания, США, Португалия, Данниия, Швеция

Данное тестирование показало, что в 95% случаев метод выдает хотя бы одну страну, которую пользователь указал в опросе. Также можно сделать вывод о том, что количество посещенных пользователем стран сильно влияет на точность рекомендации. Например, у пользователей с логинами “savnastya” и “nicks.ksks” количество посещенных стран больше 5, поэтому точность рекомендации стран у этих пользователей лучше, то есть, программа выдала для каждого из них по 3 страны, которые были указаны в опросе.

Глава 4. Программная реализация

4.1 Инструменты

В качестве языка программирования был выбран Python, который является самым популярным и гибким языком и включает в себя множество библиотек, помогающих в создании проекта.

Далее представим список используемых библиотек и их назначение:

- requests - отправление запросов на сервер Instagram;
- selenium - имитация действий пользователя и сбор данных с веб-приложения Tripadvisor;
- json - работа с полученными json ответами от сервера Instagram;
- numpy, pandas - работа с датафреймами (таблицами);
- tenzorflow - работа с нейронной сетью, определяющей тональность постов;
- sklearn - кластеризация пользователей на группы;
- scipy - определение Евклидового расстояния;
- psycopg2 - работа с базой данных PostgreSQL;
- aiohttp, asyncio - асинхронное программирование для более быстрого сбора данных о публикациях пользователей;
- langdetect - определение языка, на котором написан пост;
- nltk - удаление “стоп-слов” и перевод слов в начальную форму;
- re - регулярные выражения для обработки html документов и строк;
- flask - создание веб-приложения;
- redis, rq - создание очереди задач.

4.2 Предварительный сбор, обработка данных, кластеризация

В пунктах 1.1-1.3 подробно описаны основные понятия, способы взаимодействия с веб-приложениями и способы обработки полученных данных.

Далее кратко опишем последовательность действий.

4.2.1 Сбор данных для кластеризациии

Для сбора данных с сервера инстаграм была использована библиотека requests, которая позволяет отправлять запросы и получать ответы в формате json. Полученную информацию необходимо сохранить в базу данных. Для наглядности на рис. 8 показана физическая модель базы.

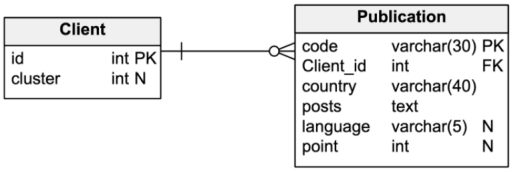

Рис. 8: Физическая модель базы данных

Более подробно про сбор данных написано в пункте 1.1.

4.2.2 Обработка данных, определение тональности постов

Первым делом необходимо определить на каком языке написан текст публикации. Это можно сделать, используя библиотеку langdetect. Сохраним полученные данные в базу, а в дальнейшем будем использовать только посты на английском языке.

Далее мы можем получить из базы все англоязычные публикации и обработать каждый пост с помощью библиотеки nltk и re.

Воспользуемся описанной в пункте 1.2 нейронной сетью и определим тональность каждого из постов. Полученные результаты сохраняем в базу данных.

4.2.3 Кластеризация

Выгрузим из базы обработанные данные пользователей, у которых в таблице Publication значение point не пустое (т.е. публикации, обработанные в пункте 4.2.2). Сгруппируем данные по идентификаторам пользователей, которые выложили публикации.

С помощью библиотек numpy и pandas создадим dataframe размером 50 000 x 255 и заполним его нулями. Столбцами данной таблицы являются страны, а строками - пользователи.

Далее необходимо рассчитать для всех пользователей средние оценки постов для каждой из стран. Оценками называем класс, к которому относится пост по тональности (5 - отличный, 4 - хороший, 3 - нейтральный и тд.). Полученные данные добавим в созданный ранее dataframe.

На следующем шаге с помощью библиотеки sklearn необходимо определить оптимальное количество кластеров и кластеризовать пользователей, с использованием полученного dataframe. Результаты кластеризации нужно сохранить в csv файл и в базу данных.

Более подробно про определение оптимального количества кластеров и кластеризацию написано в пункте 1.3

4.3 Определение курортных предпочтений определенного пользователя

4.3.1 Сбор данных о пользователе

Воспользуемся ранее созданным методом взаимодействия с Instagram, для которого не требуется авторизация и подтверждение пользователя (смотри пункт 1.1). Возникают трудности с тем, что ранее созданный метод собирал данные о пользователе по его идентификатору. Для решения данной проблемы необходимо с помощью библиотеки requests отправить следующий запрос:

https://www.instagram.com/LOGIN/?__a=1,

где LOGIN - логин пользователя.

В качестве ответа от сервера получаем json объект, где одним из параметров является идентификатор пользователя.

После получения специального идентификатора можно воспользоваться ранее созданным методом и собрать всю интересующую нас информацию.

Далее необходимо обработать данные следующим образом:

- перевести все публикации на английский язык;
- удалить “стоп-слова” и привести все слова к начальной форме;
- определить тональность текста.

Для того, чтобы перевести все публикации на английский язык, воспользуемся Yandex Translate API. Для наглядности на рис. 9 представлена часть программы, которая взаимодействует с API и получает переведенный текст.

```
def translate_text(self, text, dest_language="en"):
    translate_url = f'https://translate.yandex.net/api/v1.5/tr.json/translate?key={API_KEY}'
    param = {'text': text, 'lang': dest_language, 'format': 'plain'}
    header = {'Content-Type': 'application/x-www-form-urlencoded'}
    response = requests.post(translate_url, params=param, headers=header).json()
    return response['text'][0]
```

Рис. 9: Функция взаимодействия с Yandex API

Для определения тональности текста воспользуемся нейронной сетью из пункта 1.2.

Теперь нужно посчитать средние оценки постов пользователя, в каждой из стран.

После обработки постов необходимо создать dataframe размером 1 x 255, столбцами которого являются страны, и заполнить полученными оценками.

4.3.2 Выбор кластера

Для определения группы, к которой принадлежит пользователь, воспользуемся методом euclidean из библиотеки scipy. Данный метод считает

Евклидово расстояние между пользователем и каждым центроидом кластера. Будем считать, что пользователь принадлежит кластеру, если Евклидово расстояние до центроида данного кластера меньше, чем до других центроидов. Более подробно про Евклидово расстояние и выбор кластера описано в пункте 1.3

4.3.3 Поиск наиболее понравившихся стран в кластере

После определения кластера, к которому принадлежит пользователь, необходимо проанализировать имеющиеся оценки стран в данной группе. Будем считать, что люди, находящиеся в одном кластере, имеют идентичные курортные предпочтения, поэтому мы можем рекомендовать каждому пользователю наиболее популярные страны в данной группе, у которых средняя оценка больше 4,5.

Выводы

Таким образом, анализируя страницу пользователя из социальной сети Instagram, мы получаем всю необходимую информацию о курортных предпочтениях данного пользователя и можем предлагать ему список стран, которые с большой вероятностью ему понравятся.

Возвращаясь к статистике о том, что более 51% российских туристов обращаются в турагентства для поиска и бронирования подходящего отдыха, хотим отметить, что данная программа позволит людям, которые ведут активно свою Instagram страницу и не знают куда им поехать на отдых, определиться со страной, а в дальнейшем с курортом, не выходя из дома.

Реализованная программа дает неплохие результаты, но в дальнейшем их можно улучшить следующими способами:

1. увеличить начальный набор данных (взять за основу больше 50 000 страниц) для улучшения качества кластеризации;

2. каждый раз при запуске программы для нового пользователя, добавлять его в кластер и пересчитывать центроиды.

Заключение

В результате с помощью разработанного метода интеллектуального анализа курортных предпочтениий удалось создать общедоступный сервис [10], который на основе публикаций пользователя в социальной сети Instagram рекомендует ему новые страны для путешествий.

В работе используются размеченные отзывы с веб-платформы Tripadvisor для обучения нейронной сети, а также данные о 50 000 пользователях, их публикации с отмеченными геолокациями и тексты постов для кластеризации людей по курортным предпочтениям.

Для определения качества работы созданного метода интеллектуального анализа данных о курортных предпочтениях пользователей Instagram был проведен опрос среди пользователей с открытыми аккаунтами, в котором приняло участие 48 человек. Созданное программное обеспечение верно определило интересы 95% пользователей, участвовавших в опросе.

В дальнейшем планируется улучшение работы системы и создание адаптивного инструмента для ее использования.

Список литературы

[1] Петин. В.А., «API Яндекс, Google и других популярных веб-сервисов.», 2012. 480 с.

[2] Satya Avasaral, «Selenium WebDriver Practical Guide.», 2014. 264 с.

[3] Harry J.W. Percival, «Test-Driven Web Development with Python.», 2017. 613 с.

[4] Бен Хеник, «HTML и CSS путь к совершенству», 2011. 240 с.

[5] Охеда Тони, Билбро Ребекка, Бенгфорт Бенджамин, «Прикладной анализ текстовых данных на Python», 2019. 368 с.

[6] Официальный сайт документации API Instagram. URL: `https://www.instagram.com/developer/`.

[7] Официальный сайт документации API Yandex Translate. URL: `https://yandex.ru/dev/translate/doc/dg/concepts/about-docpage/`.

[8] Официальный сайт документации Tensorflow. URL: `https://www.tensorflow.org/api_docs`.

[9] Официальный сайт документации Sklearn. URL: `https://scikit-learn.org/stable/`.

[10] Сервис представленный в данной работе. URL: `https://insta-resorts.herokuapp.com/`.

[11] Программная реализация метода. GitHub URL: `https://github.com/GrigorKhachatryan/instagram_parser_without_api`.

Приложение

Продемонстрируем веб-интерфейс созданного сервиса. На рис. 10 стартовая страница, которую видит клиент, когда заходит на сайт.

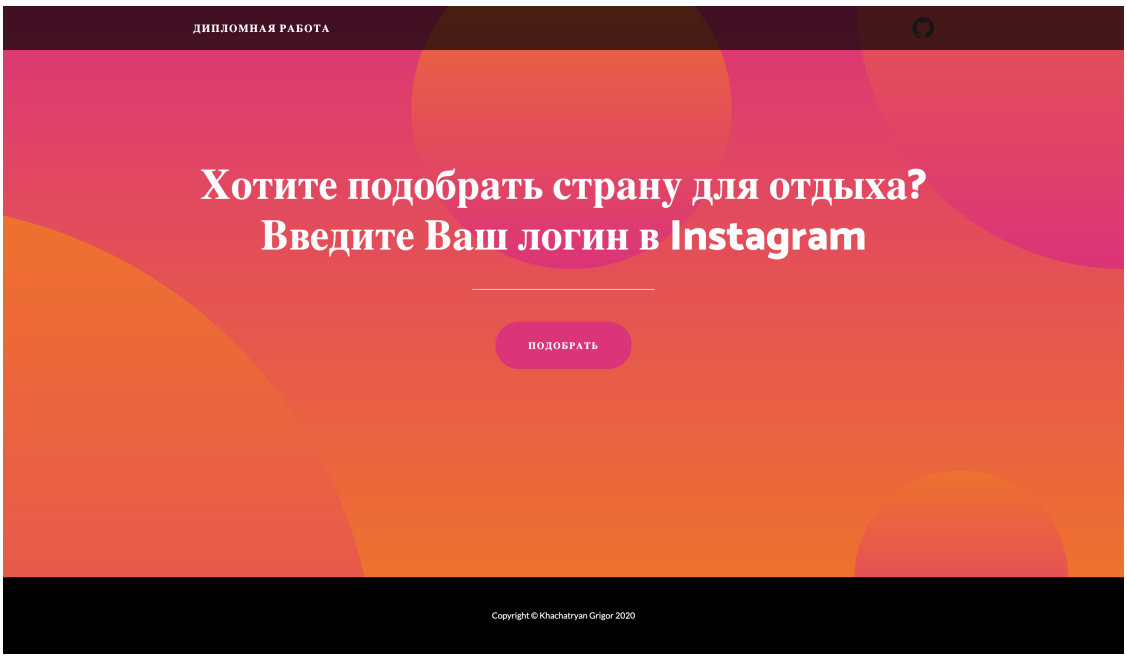

Рис. 10: Стартовая страница

Далее пользователь должен ввести свой логин из социальной сети инстаграм и нажать на кнопку «Подобрать». На рис. 11 можно наблюдать введенные и отправленные данные, а также ответ от сервиса «please wait...», который означает, что данные переданы на обработку.

Созданные методы интелектуального анализа о курортных предпочтениях проанализируют страницу пользователя. На рис. 12 и 13 показаны 3 рекомендованные страны.

С кодом данного сервиса можно ознакомиться на странице репозитория GitHub [11].

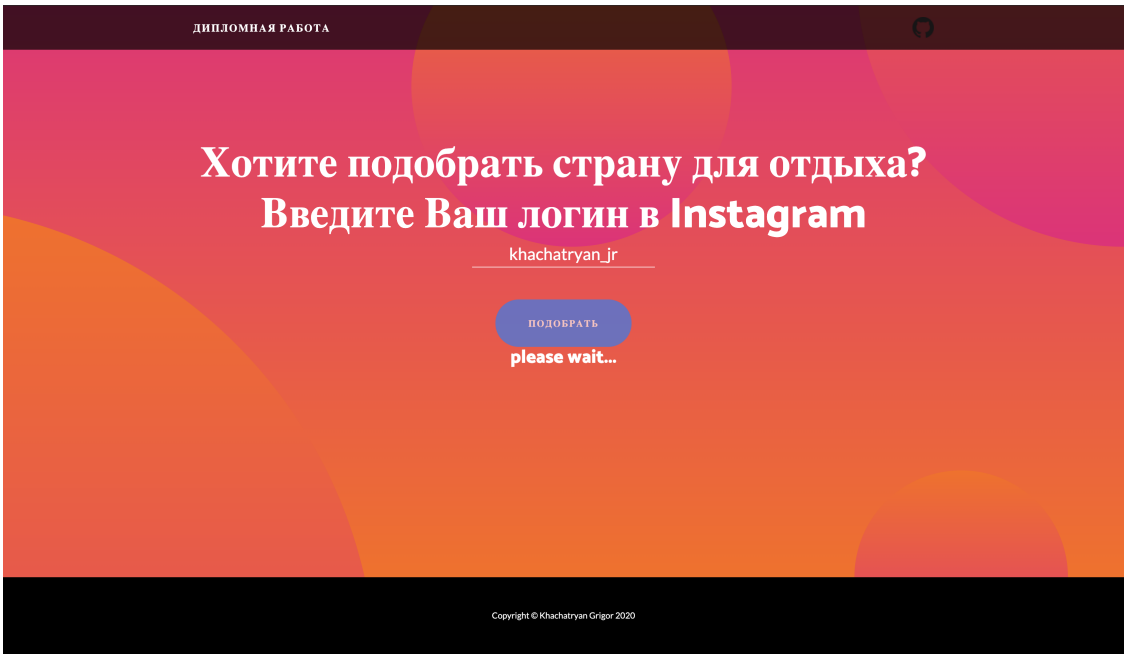

Рис. 11: Ввод данных и отправка на обработку

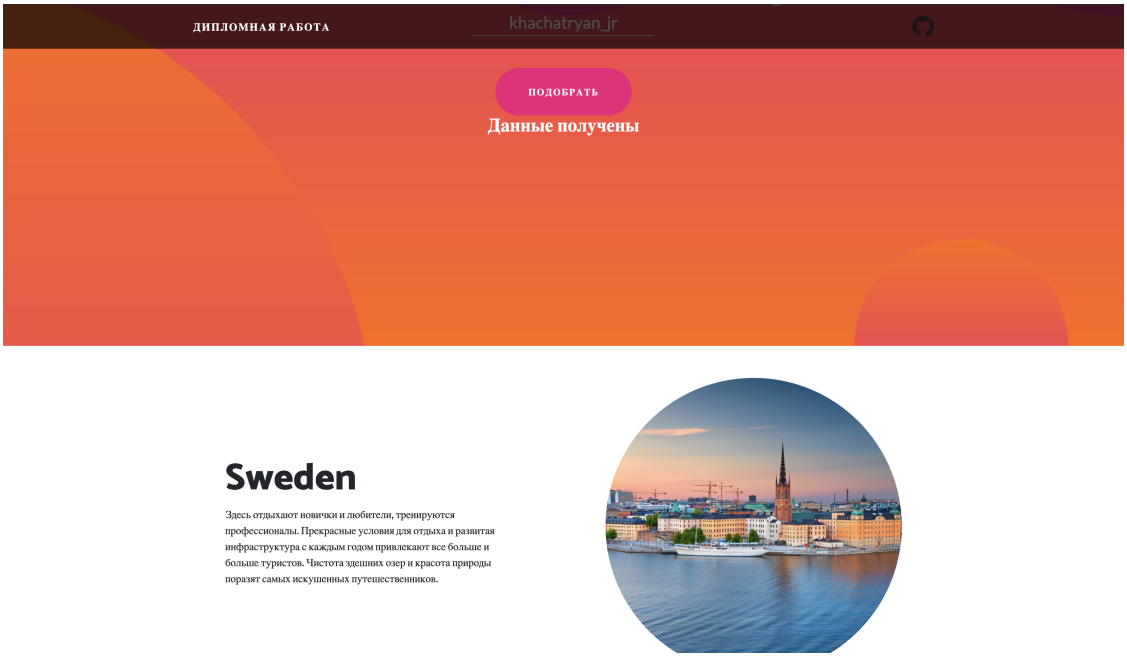

Рис. 12: Рекомендованные страны

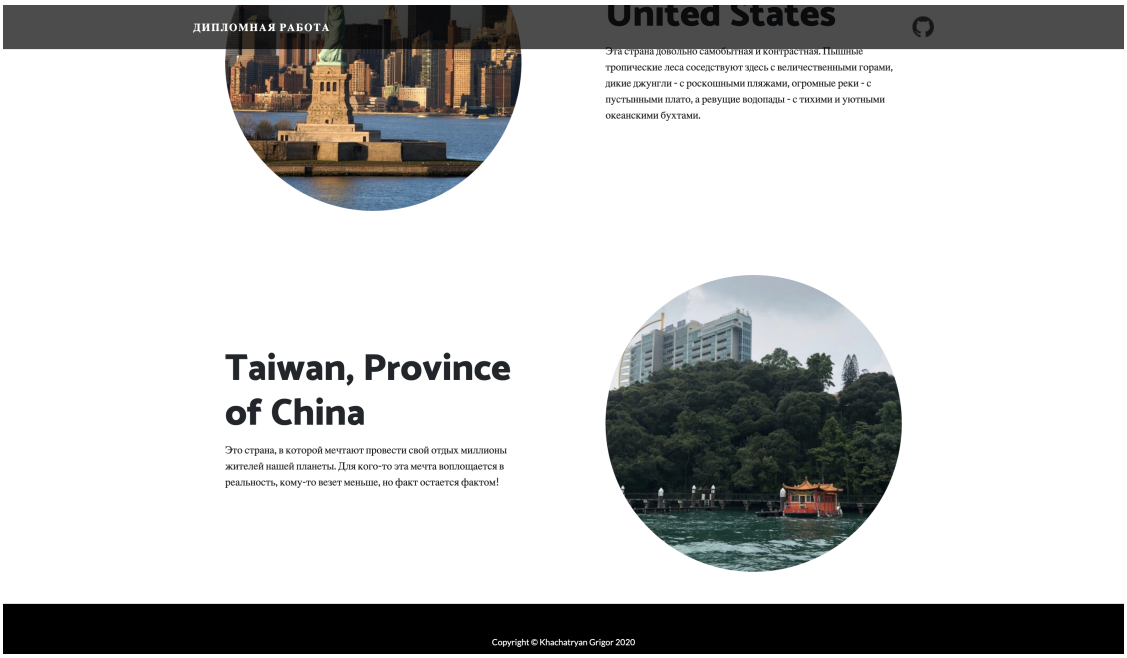

Рис. 13: Рекомендованные страны