МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»
Кафедра Дискретной математики и информационных технологий

РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ ДЛЯ АНАЛИЗА ДАННЫХ РЕГИОНАЛЬНОЙ СТАТИСТИКИ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 421 группы
направления 09.03.01 — Информатика и вычислительная техника
факультета КНиИТ
Окунева Андрея Александровича

Научный руководитель
доцент, к. э. н. _____ Г. Ю. Чернышова

Заведующий кафедрой
к. ф.-м. н. _____ Л. Б. Тяпаев

Саратов 2022

ВВЕДЕНИЕ

В условиях большого объема неструктурированной информации важной задачей является получение выборок, пригодных для дальнейшего исследования с помощью различных методов. Большие объемы информации представлены в виде открытых источников на различных интернет страницах в виде отдельных файлов в текстовых форматах. Задача представления данных в виде пригодном для дальнейшей обработки является весьма актуальной.

Целью дипломной работы является разработка web-приложения для анализа данных социально-экономических показателей.

Задачами дипломной работы представляют собой:

— применение технологии web-скрапинга для сбора данных из открытых источников;

— проектирование приложения для сбора и обработки данных региональной статистики;

— применение разработанного приложения на примере данных с сайта федеральной службы государственной статистики.

Объектом дипломной работы является совершенствование процессов обработки данных.

Предметом дипломной работы является разработка технологий для анализа данных региональной статистики.

Структура работы состоит из введения, двух глав, заключения и списка используемых источников. В первой главе представлен теоретический обзор применения технологий web-скрапинга, обзор инструментальных средств web-скрапинга и обзор проблем, возникающих в процессе сбора данных из открытых источников. Во второй главе представлен обзор разработанного приложения и пример применения приложения. Общий объем работы – 74 страницы, из них 41 – основное содержание, включая 12 рисунков, список использованных источников – 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел "Анализ методов автоматизации сбора неструктурированных данных" посвящен теоретическому обзору применения технологий web-скрапинга, обзору инструментальных средств web-скрапинга и обзору проблем, возникающих в процессе сбора данных из открытых источников.

Рассмотрим понятие web-скрапинга, которое широко используется в информационных технологиях. Web-скрапинг – это автоматизированный процесс сбора больших объемов web-данных путем извлечения их со страниц web-ресурсов [1]. Эта технология позволяет избежать рутинного и утомительного процесса ручного извлечения данных, используя интеллектуальную автоматизацию. Для сбора больших объемов данных необходима автоматизация, и web-скраперы выполняют именно эту функцию.

Web-скрапинг автоматизирует целый ряд задач [2]:

— копирование данных из Интернет;
— поиск необходимой информации;
— мониторинг обновлений информации на сайтах.

Web-скрапинг используется следующими прикладными системами:

— поисковые системы;
— агрегаторы контента;
— научные и маркетинговые;
— наборы данных для машинного обучения.

Обычно процесс web-скрапинга состоит из следующих этапов:

— определение целевого web-сайта;
— извлечение кода web-страниц из интернета с помощью посылки HTTP-запросов по URL-адресам;
— извлечение необходимой информации из кода web-страниц с помощью специальных механизмов (регулярные выражения, HTML-парсеры, искусственный интеллект);
— выделение необходимой информации из HTML кода;
— структурирование и сохранение полученных данных в таблицах или базах данных.

Существует несколько технологий для извлечения данных: регулярные выражения, web-парсеры, автоматический скрапинг с использованием искусственного интеллекта.

Web-скраперы можно разделить на несколько видов:

— пользовательские web-скраперы;

— облачные web-скраперы;

— web-скрапер как браузерное расширение;

— программные web-скраперы.

Можно выделить несколько методов web-скрапинга: копипаст вручную, обращение к прокси-сервису, сопоставление текстовых шаблонов, синтаксический анализ HTML, DOM, распознавание семантических аннотаций, анализаторы страниц.

При разработке приложения применялся метод Синтаксического анализа HTML.

Среди инструментальных средств web-скрапинга можно выделить:

— библиотеки MechanicalSoup и BeautifulSoup;

— Selenium WebDriver;

— Scrapy;

— Any23;

— Import.IO;

— Scraper API;

— Octoparse;

— DataOx.

В нашем случае для скрапинга применялся Selenium WebDriver с целью получения всех документов со страницы.

Некоторые администраторы заинтересованы в том, чтобы сделать невозможной работу web-скраперов на их сайте, так как это может приводить к проблемам с производительностью. Также, полученная информация может использоваться конкурентами для получения преимущества на рынке.

Таким образом, важно исследовать процесс web-скрапинга как с точки зрения защиты информации администраторами web-сайтов, так и с точки зрения web-скрапера, которому необходимо иметь возможность обойти встроенную блокировку сайта.

Программы скрейпинга могут быть распознаны по следующим признакам:

— необычное поведение пользователя (например, сотни переходов на новую страницу сайта каждую секунду);

— повторяющиеся безрезультатные действия (пользователь не будет выполнять одни и те же задачи раз за разом);

— использование ссылок, которые содержатся только в коде web-сайта и не видны обычным пользователям.

К способам блокировки относятся:

— запрет доступа на сайт с определённого IP-адреса (например, когда ботом пройдено более 100 страниц за сессию);

— запрет идентификатора пользователя, являющийся с точки зрения администратора сайта злоумышленником, заходящим на сайт по аутентификации.

Чтобы обойти блокировку, программы web-скрапинга должны производить на сайте действия, максимально близкие к поведению пользователей.

Второй раздел "Разработка web-приложения для анализа социально-экономических показателей" посвящен обзору разработанного приложения и примеру применения приложения.

Web приложение было разработано на языке программирования Python версии 3.10.4 [3].

Также при разработке приложения применялись библиотеки Selenium, Docx и Flask. Библиотеки были установлены с помощью системы управления пакетов pip.

Файлы с данными региональной статистики находятся на странице сайта федеральной службы государственной статистики. Всего файлов 394.

На сайте gks.ru в разделе региональной статистики представлены 22 группы показателей [4]. В каждой группе имеется список отдельных показателей социально-экономического развития. Причем структура каталогов может быть многоуровневой. Значения отдельных показателей находятся в файле с форматом .doc в виде таблиц с данными, которые немного отличаются друг от друга. Причем приводятся данные за несколько периодов. Значения каждого показателя представлены для каждого региона.

Чтобы при обновлении сайта не скачивать их каждый раз вручную, на первым этапом разработки был разработан web-скрапер с целью автоматического получения всех документов со страницы сайта gks.ru.

Изначально разрабатывать приложение планировалось на языке про-

граммирования C# с использованием библиотеки для web-скрапинга ScrapySharp. Но выяснилось, что сайт подгружает ссылки на файлы динамически, при раскрытии выпадающих списков, и справиться с этой проблемой позволила библиотека Selenium, которая в таких случаях позволяет дождаться полной загрузки страницы.

При запуске web-скрапер открывает браузер, ждет загрузки сайта, ищет элементы, которые относятся к раскрываемым спискам и последовательно раскрывает все выпадающие списки, ожидая подгрузки ссылок на файлы. Далее программа ищет в коде страницы элементы списка, которые содержат ссылки на документы, и создает список этих ссылок. После этого web-скрапер скачивает все файлы, сохраняя структуру папок как на сайте [5].

Следующий этап разработки – написание функций для обработки скаченных документов.

В приложении были реализованы следующие функции: checkFiles, checkRegions, check, checkBoth, save, graph, graphBoth, minimum, maximum, average, mediana, dispersion, checkDates, checkStatistics, checkDatesBoth, checkStatisticsBoth.

Функция checkFiles последовательно проходит по всем папкам с файлами, и файлам, которые были получены с помощью скрапера, открывает каждый файл и проверяет, есть ли внутри него таблица с данными статистики. Если таблица присутствует, то далее проверяется наличие периодов в шапке таблицы. Если периоды присутствуют, файл записывается в список документов [6]. Далее список документов сохраняется как txt файл и при следующем запуске программы, если документы не были обновлены web-скрапером, повторное выполнение этой функции не требуется.

Функция checkRegions принимает на вход порядковый номер i документа, считывает список документов, созданный функцией CheckFiles, находит в списке документов название с порядковым номером i, находит файл с нужным названием в папке со всеми документами, считывает его и создает список регионов для этого файла.

Функция check принимает на вход порядковый номер файла i и порядковый номер региона j, по ним из списка документов и списка регионов функция получает названия файла и региона, находит нужный файл, регион в этом файле. Далее, получив данные статистики для этого региона, функция

check вызывает функцию по построению графика, и функции описательной статистики.

Функция graph получает на вход название региона, название файла, данные статистики для региона и строит диаграмму для этого региона, указывая в заголовке название файла и региона. В некоторых документах, помимо периодов в шапке документа, встречается запись вида "Место, занимаемое в Российской Федерации". Функция также способна обработать эту информацию и вывести ее в заголовок графика.

Функция graphBoth получает на вход название файла, названия двух регионов, данные статистики для этих регионов и рисует два графика на одной диаграмме, указывая в заголовке название файла и регионы.

Функция CheckDates получает на вход порядковый номер документа и возвращает список периодов в этом документе.

Функция CheckStatistics получает на вход порядковый номер документа и год, записывает в файл формата csv данные региональной статистики для каждого региона, строит диаграмму для всех регионов, сохраняет ее в формате PNG.

Функция CheckDatesBoth получает на вход порядковые номера двух документов, находит года в каждом из них и возвращает список периодов, которые присутствуют в обоих файлах.

Функция CheckStatisticsBoth получает на вход порядковые номера двух документов и год, находит совпадающие в двух файлах регионы, записывает в csv файл данные региональной статистики для этих регионов для выбранного года, считает коэффициент корреляции Пирсона.

Функция minimum получает на вход список параметров и находит среди них наименьший. В некоторых документах данные пропущены или представлены в виде вещественных чисел. Для обработки таких данных в Python в первом случае необходимо заменить пропущенное значение на ноль, а во втором случае десятичный разделитель на точку. Функция позволяет обработать такие случаи. Функция maximum получает на вход список параметров и находит среди них наибольший. Функция average получает на вход список параметров и находит среднее арифметическое. Функция mediana получает на вход список параметров и находит медиану [7]. Функция dispersion получает на вход список параметров и находит дисперсию [8].

Следующий этап разработки – создание web-интерфейса [9].

На главной странице приложения находятся кнопки "Визуализация", "Запись в csv", "Корреляция Пирсона", "Обновление файлов" и "Запуск скрапера".

При нажатии на кнопку "Запуск скрапера" происходит вызов функции scrapper, которая запускает работу скрапера.

При нажатии на кнопку "Обновление файлов" происходит запуск функции checkFiles.

При нажатии на кнопку "Визуализация" происходит переход на страницу выбора документа. На этой странице находятся кнопки "На главную страницу" и "Выбрать файл". При нажатии на кнопку "На главную страницу" происходит переход на главную страницу приложения. При нажатии на кнопку "Выбрать файл" появляется список документов. При выборе документа происходит переход на страницу выбора регионов. На этой странице находятся кнопки "На главную страницу" и "Выбрать регионы". При нажатии на кнопку "Выбрать регионы" появляются списки регионов. При выборе регионов происходит вызов функции check для каждого из регионов, вызов функции checkBoth и переход на страницу с графиками. На этой странице находится кнопка "На главную страницу" и графики, которые сохранены как файлы с расширением .png.

Функционал приложения включает в себя:

— выбор показателя;
— выбор регионов;
— выбор периода;
— сохранение данных в формате csv за выбранный период для одного или для двух файлов;
— сохранение данных в формате csv для выбранного региона;
— формирование описательной статистики;
— вычисление коэффициента корреляции Пирсона;
— построение диаграмм;
— сохранение диаграмм.

Рассмотрим работу приложения на конкретном примере. Пользователь на странице выбора документа выбирает файл "Внутренние затраты на научные исследования и разработки". Далее на странице выбора регионов вы-

бирает Саратовскую и Пензенскую область. Результатом работы приложения являются три графика и описательная статистика. На графиках представлена статистическая информация по годам для двух регионов и место занимаемое регионом в РФ.

Пользователь переходит на главную страницу и нажимает кнопку "Запись в csv", выбирает документ и год. Результатом работы приложения является csv файл с данными региональной статистики за выбранный год для всех регионов и диаграмма.

Пользователь переходит на главную страницу, нажимает кнопку "Корреляция Пирсона" и выбирает два документа. Результатом работы приложения является csv файл с данными региональной статистики за выбранный год для всех регионов по двум файлам и коэффициента корреляции Пирсона.

Для файлов "Внутренние затраты на научные исследования и разработки"и "Численность персонала, занятого научными исследованиями и разработками"вычисленное значение составляет 0.95, что свидетельствует о высокой корреляционной зависимости между показателями.

ЗАКЛЮЧЕНИЕ

В данной работе был применен web-скрапинг для сбора данных с сайта федеральной службы государственной статистики. Для реализации применялась библиотека Selenium, которая позволяет получить данные с сайта, не смотря на динамическую подгрузку html кода.

Было разработано приложение для сбора и обработки данных региональной статистики. Для разработки использовались такие инструменты, как Python, pip, Flask, Docx.

Приложение обеспечивает функционал web-скрапинга, обработки данных, визуализации данных, сохранения данных в файле формата csv, вычисление описательной статистики и коэффициента корреляции Пирсона.

С помощью разработанного приложения был рассмотрен процесс обработки данных с сайта Федеральной службы государственной статистики для файлов "Внутренние затраты на научные исследования и разработки" и "Численность персонала, занятого научными исследованиями и разработками".

Предоставленное приложение обеспечивает удобный интерфейс для получения структурированного представления данных региональной статистики. Реализованные средства анализа представляют интерес для широкого круга пользователей, которые применяют данные региональной статистики.

Дальнейшим направлением исследований является расширение списка алгоритмов для обработки неструктурированных текстовых файлов различного вида.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 What is Web Scraping and How to Use It? [Электронный ресурс]. – URL: https://www.geeksforgeeks.org/what-is-web-scraping-and-how-to-use-it/ (Дата обращения 24.05.2022). Загл. с экрана. Яз. Англ.

2 Митчел, Р. Современный скрапинг веб-сайтов с помощью Python. – Санкт-Петербург: Питер, 2020.

3 Python [Электронный ресурс]. – URL: https://www.python.org/downloads/ (Дата обращения 24.05.2022). Загл. с экрана. Яз. Англ.

4 Регионы России. Социально-экономические показатели - 2020 г. [Электронный ресурс]. – URL: https://gks.ru/bgd/regl/b20_14p/Main.htm (дата обращения 24.05.2022). Загл. с экрана. Яз. Рус.

5 Документация для PYTHON [Электронный ресурс]. – URL: https://digitology.tech/docs/python_3/index.html (дата обращения 24.05.2022). Загл. с экрана. Яз. Рус.

6 Харрисон, М. Как устроен Python. Гид для разработчиков, программистов и интересующихся. – Санкт-Петербург: Питер, 2019.

7 Методы описательной статистики [Электронный ресурс]. – URL: https://sibac.info/blog/metody-opisatelnoy-statistiki (Дата обращения 24.05.2022). Загл. с экрана. Яз. Рус.

8 Дятлов, А.В., Лукичёв., П.Н. Методы математической статистики в социальных науках (описательная статистика). – Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2018.

9 Дронов, В. Практика создания веб-сайтов на Python. – Санкт-Петербург: Санкт-Петербург, 2019.