ДИПЛОМ

Рекомендательный сервис для выбора активностей

СОДЕРЖАНИЕ

Введение

Основная часть:

Архитектура

Парсинг

Загрузка данных в базу

Или SDLC: анализ, планирование, проектирование, разработка

Список литературы

Заключение

Приложения:

- БПМН

- названия столбцов в начальных таблицах

ВВЕДЕНИЕ

ОСНОВНАЯ ЧАСТЬ

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

**Описание проекта**

В Москве работает проект «Еще не бабушка» для граждан старшего возраста (55+ для женщин и 60+ для мужчин), который предоставляет им возможность принимать участие в широком диапазоне активностей.

Активности проходят офлайн и онлайн формате. Для удобства работы руководство проекта планирует внедрение рекомендательной механики.

**Постановка задачи**

Создание рекомендательной системы на основании имеющихся данных о пользователях и их активности в проекте.

Предоставлены данные о зарегистрированных пользователях, на их основании должна проводится идентификация пользователя в системе. Такому пользователю система должна предлагать активности на основании истории его посещений. Для нового пользователя система должна предлагать активности на основании поведения зарегистрированных пользователей.

Предусмотрены следующие пути пользователя:

а) поиск группы через поисковую строку;

б) поиск через каталог групп;

в) поиск через прохождение тестирования, которое определяет предпочтение пользователя.

Приоритет при подборе активности - офлайн занятия в районе проживания пользователя.

Основная задача сервиса - максимальное сокращение пути пользователя до кнопки “Записаться” (в группу).

АРХИТЕКТУРА ПРОЕКТА

**Цель работы**

Разработка технического проекта для возможного решения рекомендательного сервиса для проекта «Еще не бабушка» на основе предоставленных данных (далее по тексту ИД).

В практической части необходимо реализовать сервис на Python.

**Описание данных**

ИД содержат информацию о пользователях проекта «Еще не бабушка» и их активностях в 2023 году. ИД представлены 3 файлами в формате csv и одним файлом в формате excel. В каждом файле находится одна таблица.

В ИД содержится информация:

groups.csv - “Группы”

users.csv - “Пользователи”

attend.csv - “Посещаемость”

dict.xlsx - “Справочник направлений”

Наименования столбцов перечислены в Приложении 1.

**ERD**

Выделим следующие сущности:

пользователь

вопрос

ответ

тип активности

уровень 1 активности

уровень 2 активности

уровень 3 активности

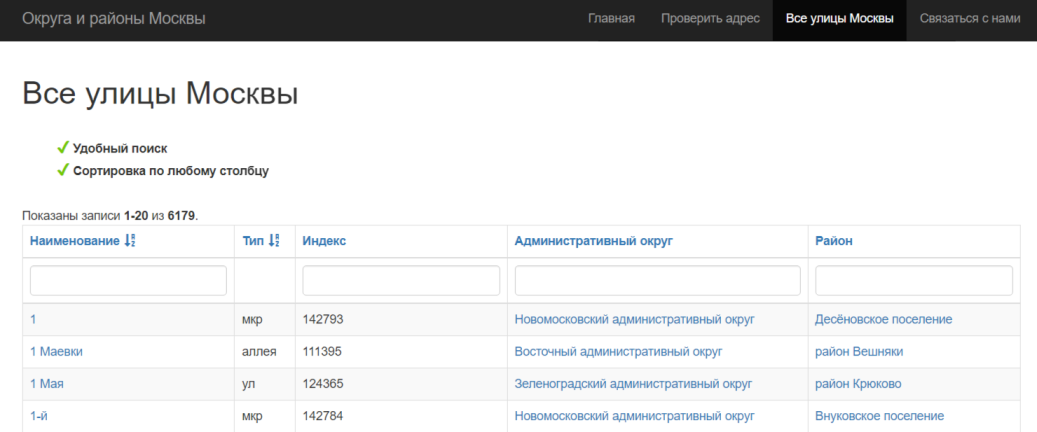
Для пользовтелей понадобится парсинг!

таблице юзерс представлены только адреса пользовтаелей.

В таблице групс есть привязка адресов к районам и округам.

Для удобства работы по местоположению пользователя добавлены район и округ проживания.

Для парсинга взят сайт <http://округа-районы.москва/все_улицы_москвы>.



Использовались библиотеки: BeautifulSoup и pandas. С помощью инструментов разрабочика на сайте получено название таблицы <table class="table table-striped table-bordered”>, и из нее в цикле выгружена информация из каждой строки.

Обработка данных не использовалась. Заголовки созданы вручную, так как таблица одна.

Результат загружен в таблицу "moscow\_streets".

Итого получена сущность "москоу стритс".

КАРТИНКА

СДЕЛАТЬ ТАК ЧТОБ парсер останавлився не от нолмера страницы

КАРТИНКИ

=

4. Описание архитектуры

База данных, построенная с использованием MySQL

Компоненты:

- Python, включая библиотеки:

-os,

-shutil,

-logging,

-pandas, numpy

-sklearn,

-Beautifulsoup

-

Архитектура проекта заключается в:

- последовательной обработке поступающего файла в формате excel;

- парсинге данных с интернет-ресурса;

- последующей записи полученных данных в базу данных (SQLite) с префиксом

dwh;

- подготовке данных для витрины;

- визуализации данных из витрины;

- создании модели классификации на основе ИД.

- последующей записи полученных результатов в базу данных (SQLite);

Весь код обработки данных написан на Python и SQL. В ИД не указано, как часто

поступают данные для обработки. Для целей данной работы предположим, что файл

поступает раз в сутки, также раз в сутки запускается скрипт. Задача скрипта -

добавление в базу новых данных из полученного нового файла и его последующая

архивация.

База данных состоит из двух уровней:

- стейджинг, на который подгружаются приходящие данные

- основной, где хранятся данные

5. Выбор СУБД

## Преимущества MySQL

### Производительность

Для веб-приложений важна скорость работы. Пользователь не будет ждать загрузки страницы — он просто уйдет с неё, если приложение не будет работать достаточно быстро. В плане производительности приложения «болезненным» местом является база данных, и MySQL не подведет вас.

### Кроссплатформенность

MySQL предоставляет кроссплатформенную базу данных: она работает на Linux, FreeBSD и конечно на Windows. Этот критерий нужно учитывать при выборе СУБД для проектов, нацеленных на несколько платформ, в частности веб-приложений. К слову, MySQL входит в состав популярных стеков для разработки веб-приложений — LAMP (Linux, Apache, MySQL, PHP) и WAMP (Windows, Apache, MySQL, PHP).

### Бесплатная СУБД

Oracle развивает MySQL как проект с открытым исходным кодом, что позволяет сообществу вносить в него свой вклад и, при необходимости, адаптировать его под свои задачи. СУБД предоставляется для конечных пользователей бесплатно. Многие компании выбирают в качестве СУБД MySQL из-за того, что она бесплатна.

Oracle предлагает клиентам некоторые проприетарные модули с закрытым кодом, для использования которых потребуется денежное вложение. Но эти модули нужны не всем и не всегда.

### Выбор движков

Движок базы данных (database engine) — это программный модуль, с помощью которого СУБД создает, читает и обновляет данные в базе. В MySQL есть два типа движков: транзакционные и нетранзакционные. Транзакционные движки хранения данных подразумевают, что операции в этих БД могут быть отменены, если они не завершены.

Всего MySQL поддерживает 8 движков:

**InnoDB**— это распространенный движок, который используется в MySQL по умолчанию.  Он поддерживает ACID, блокировку на уровне строк, восстановление после сбоев и многопоточность. Также это единственный движок, который обеспечивает ограничение ссылочной целостности внешнего ключа. Oracle советует использовать именно InnoDB, за исключением ряда случаев.

**MyISAM**— этот движок использовался по умолчанию до появления InnoDB. И в отличие от него, MyISAM не поддерживает ACID. В этом движке реализована блокировка только на уровне таблиц, поэтому он небезопасен для транзакций. В тоже время MyISAM оптимизирован для чтения. Сфера применения MyISAM — это системы хранения с минимальными транзакционными данными.

**Memory**— этот движок реализуется хранение данных в памяти в виде кучи. Такая система обеспечивает очень быстрый доступ к данным. Memory поддерживает блокировку на уровне таблиц и хранит данные в оперативной памяти — при перезагрузке сервера они теряются. Этот движок подходит для временных таблиц.

**CSV**— этот движок хранит данные в файлах формата .csv. Не поддерживает индексирование и разделение. CSV используется для обмена данными между различными программами или приложениями.

**Merge**— движок работает с таблицами MyISAM. Он логически группирует ряд идентичных таблиц MyISAM и ссылается на них, как на один объект. Merge упрощает управление большими объемами данных и используется в системах хранения данных.

**Archive**— этот движок оптимизирован для высокоскоростной вставки. При вставке он сжимает данные. Archive не поддерживает транзакции, операции DELELE и UPDATE, но зато поддерживает блокировку на уровне строк. Отлично подходит для хранения больших объемов архивных данных, на которые редко ссылаются.

**Federated**используется создания одной БД на нескольких физических серверах.  Подходит для распределенных систем.

Движок **Blackhole**принимает данные, но не сохраняет их. На первый взгляд этот движок бесполезен, но на самом деле он находит применение при репликации данных и тестировании. Blackhole можно использовать в качестве фильтра между серверами.

### Безопасность

MySQL предлагает пользователям высокий уровень безопасности. В MySQL есть встроенные инструменты безопасности, которые поддерживают управление пользователями и их привилегиями. При недостатке стандартных инструментов пользователь всегда может установить дополнительные плагины.

### Сообщество

Большое и развитое сообщество MySQL не оставит пользователя одного с проблемой. Кроме того, благодаря сообществу на базе MySQL появились новые системы управления базами данных — Drizzle, OurDelta, Percona Server и, самый популярный, MariaDB.

## Минусы MySQL

### Работа с большими базами

При работе с малыми и средними приложениями MySQL радует пользователей, но при увеличении объемов информации у MySQL возникают проблемы. Речь идет о базах данных с миллионами строк. Такие БД тяжело масштабировать, а сложные запросы к ним начинают буксовать и иногда не успевают выполняться в допустимое время ожидания. Для решения проблемы с запросами придется прилагать усилия для их оптимизации.

Большие объемы информации— это сфера применения для Oracle Database или SQL Server, а не MySQL.

### SQL

MySQL не соответствует стандарту SQL: СУБД не поддерживает некоторые функции и имеет расширения, не относящиеся к стандартному SQL. Для большинства проектов это не станет серьезной проблемой, но при миграции на другие СУБД этот момент может всплыть.

### Развитие

MySQL — это проект с открытым исходным кодом, что негативно сказывается на скорости его развития. Сообщество разработчиков сосредоточено в первую очередь на поддержку уже существующих функций, а Oracle, с учетом наличия проприетарных модулей для MySQL с закрытым кодом, не сильно заинтересованы в развитии проекта.

## Сфера применения

Для хранения табличной структуры выбрана реляционная база данных SQLite.

SQLite обладает рядом параметров, которые являются достаточными для нашего

проекта. Работает без сервера, при необходимости легко встраивается в приложение,

есть дополнительные библиотеки с открытым кодом, довольно высокая

производительность до 200 запросов на страницу.

6. Решение по используемой СХД

СХД - устройство для хранения и управления данными и их резервного

копирования. ИД приходят в файлах. За год всего 20000 транзакций, следовательно

промежуточные файлы удобно хранить в файловой СХД. Преобразованные данные

хранятся в базе данных, размещенной на блочной СХД.7. Перечисление алгоритмов и методов анализа

предлагаемых к использованию в решении кейса.

и

обработки,

Парсинг, обработка поступающих данных, запись и обновление данных в БД

происходит с помощью DAG в Airflow. Модель классификации и метрики

сохраняются в Model Registry в MLFlow. Результат записывается в базу данных.

Airflow и MLFlow разворачиваются из контейнера Docker.

8. Описание модели угроз

Перечень возможных угроз:

1. Неограниченные привилегии доступа у сотрудников

2. Внедрение SQL – кода

3. Непрофессионализм сотрудников при работе с информацией

4. Открытие копии БД

Список параметров фильтрации групп в сервисе записи:

●направления занятий (1,2,3 уровней)

●формат занятий (очно / онлайн)

●район занятий (для очных занятий)

●расписание (дни недели / время)

Список данных, содержащихся в датасете

В

СПИСОК ЛИТЕРАТУРЫ

<https://timeweb.cloud/blog/mysql-preimushchestva-i-nedostatki>

ЗАКЛЮЧЕНИЕ

СПИСОК ЛИТЕРАТУРЫ