

Финальная работа

СПЕЦИАЛИЗАЦИЯ DE

Славкин Илья

[Анализ сайта «СберАвтоподписка»]



[Немного о себе]

Чем сейчас занимаюсь?

Являюсь специалистом по промышленной гидравлике, работаю в сфере продаж.



Зачем мне курс?

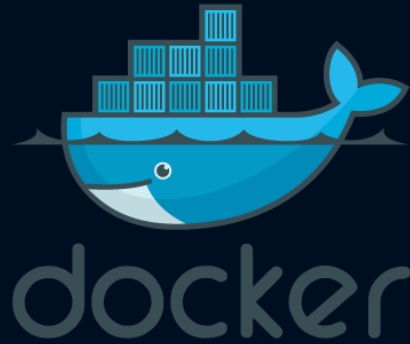
По зову сердца совершаю потрясающее путешествие-переход в профессию "дата-инженер".



Дальнейшие планы

Собирать и применять знания в сфере работы с данными и внести свой вклад в развитие страны.

[Чему научился на курсе]



[Описание проекта]



«СберАвтоподписка» — это сервис долгосрочной аренды автомобилей для физлиц, предлагает новый для российского рынка способ владения автомобилем и выступает в качестве альтернативы автокредиту.

Клиент платит фиксированный ежемесячный платёж и получает в пользование машину на срок от шести месяцев до трёх лет.



[Описание проекта]



На сайте сервиса пользователь совершает **целевые** (нажимает кнопки типа «Оставить заявку», «Заказать звонок») и нецелевые действия (просмотр карточек авто или «блуждания» по основной странице и страницам с помощью).

Перед командой аналитики стоит задача **сбора и обработки данных**, поступающих из внутренних систем компании, а также с сайта сервиса.



[Цели и задачи проекта]

Обработать
предоставленный
датасет

Создать локальную
базу данных и
заполнить её
обработанными
данными

Настроить конвейер
сбора, обработки и
записи новых
данных



Получим работающую заполненную
базу данных с механизмом добавления
новых данных по расписанию.

[Реализация проекта]

1

Анализ датасета и первичная обработка

Производим базовую обработку и разведочный анализ данных из предоставленного датасета .csv.



Что используем?

- Jupyter Notebook
- библиотека Pandas

Чтобы познакомиться с данными, найти правильный подход в их обработке и увидеть особенности.

Зачем?

[Реализация проекта]

2

Написание промышленного кода для обработки датасета

Преобразуем код из Jupyter Notebook в промышленный формат.
Сохраняем обработанные данные во временные файлы .csv.



Что используем?

- PyCharm
- библиотеки Pandas, logging*

Для удобства использования
в производстве и приведения
кода к стандартам языка.

Зачем?

Реализуем задумку в отдельном модуле.

*Используем библиотеку logging для корректной записи сообщений выполнения кода

[Реализация проекта]

3

Создание и заполнение базы данных

Создаем нового пользователя с паролем, базу данных и таблицы, размеченные под наш датасет в полуавтоматическом режиме.

В конце удаляем временные файлы, созданные на предыдущем этапе.



Что используем?

- PyCharm
- SQL
- PostgreSQL
- библиотеки `psycopg2`, `logging`, `os`, `configparser`

Реализуем задумку в отдельном модуле.

[Реализация проекта]

3

Создание и заполнение базы данных Особенности



Создаем конфигурационный файл .ini с информацией о местонахождении проекта и параметрами для подключения к базе данных.

Информация заносится в файл перед началом работы и используется в процессе.

Это удобный формат для идентификационных данных, одно из возможных решений для масштабирования проекта.

Зачем?

Используем библиотеки `os` и `configparser` для реализации задумки.

[Реализация проекта]

3

Создание и заполнение базы данных Особенности



Создаем первичные и внешние ключи для связи таблиц в базе данных.



Предусмотрено удаление строк в таблице с внешними ключами, если значения ключей отсутствуют в главной таблице.

Для сохранения логической связи между таблицами и корректности данных в базе.

Зачем?

[Реализация проекта]

4

Добавление новых данных вручную

Загружаем новые файлы .json в датафреймы pandas, обрабатываем их по аналогии с основным датасетом на созданном конвейере обработки данных, добавляем в базу данных.



Что используем?

- PyCharm
- библиотеки Pandas, logging, os, glob, json, sklearn, sqlalchemy
- функции модулей других этапов

Для ручного выполнения операций сбора, обработки и загрузки данных в БД.

Зачем?

Реализуем задумку в отдельном модуле.

[Реализация проекта]

4

Добавление новых данных вручную

Особенности



Предусмотрен механизм разрешения конфликтов при записи



Предусмотрен отказ от записи строк в таблицу с внешними ключами, если значения ключей отсутствуют в главной таблице.

Механизмы разрешения конфликтов введены во избежание ошибок записи, дублирования данных.

Зачем?

[Реализация проекта]

5

Добавление новых данных по расписанию

Реализуем DAG Airflow сборки, обработки и загрузки новых данных с периодичностью в один день.

Все особенности идентичны этапу 4.



Что используем?

- PyCharm
- Apache Airflow
- библиотеки Pandas, logging, os, sys, glob, json, sklearn, sqlalchemy

Реализуем задумку в отдельном модуле.

Для добавления новых данных в БД по расписанию. Например, каждый вечер мы совершаем загрузку данных за день.




Зачем?

[Итоги]

- ✓ С помощью Jupyter Notebook и библиотеки Pandas мы познакомились с данными, нашли правильный подход в их обработке и увидели особенности датасета.
- ✓ Используя среду PyCharm, применили полученные знания для написания промышленного кода.
- ✓ Настроили PostgreSQL, создали базу данных с объектами и заполнили их, используя язык SQL.
- ✓ Добавили механизмы разрешения конфликтов при записи в базу данных.
- ✓ Создали конвейер обработки данных.
- ✓ Реализовали Airflow-пайплайн сборки, обработки и записи новых данных.

[Итоги]

В результате мы имеем **полностью работающее решение** поставленной задачи:

-  базу данных PostgreSQL с обработанными данными
-  программный модуль, позволяющий в полуавтоматическом режиме осуществить обработку основного датасета, создать базу данных и заполнить её, а также выполнить сбор, обработку и добавление новых данных
-  DAG в Airflow, выполняющий запись в базу данных новых файлов по расписанию

Подробнее с проектом можно ознакомиться
https://github.com/slawilja/final_work



FINISH