Dokumentacja Projektu 3

DAG 1: data_download_and_split_dag

Ten DAG zajmuje się pobraniem danych z zewnętrznego źródła (w tym przypadku z Kaggle), a następnie podzieleniem danych na dwa zbiory:

- Zbiór modelowy (70%)
- Zbiór douczeniowy (30%)

Po podziale danych, oba zbiory są zapisywane do osobnych arkuszy Google Sheets, co umożliwia ich dalsze wykorzystanie w kolejnych etapach analizy (np. w drugim DAG-u).

Kroki:

1. **fetch_data (PythonOperator)**
    - Pobiera dane z wybranego źródła (tu: Kaggle dataset).
    - Wczytuje dane do DataFrame Pandas.
    - Zapisuje dane do XCom.
2. **split_data (PythonOperator)**
    - Pobiera dane z XCom.
    - Dzieli dane za pomocą train_test_split na zbiór modelowy (70%) i douczeniowy (30%).
    - Używa random_state=42 aby zapewnić reprodukowalność.
    - Zapisuje podzielone dane do XCom.
3. **save_data (PythonOperator)**
    - Pobiera zbiory modelowy i douczeniowy z XCom.
    - Aktualizuje odpowiednie arkusze Google Sheets.
    - Każdy zbiór zapisuje w osobnym arkuszu, umożliwiając łatwy dostęp i dalsze wykorzystanie.

Wymagania techniczne:

- Plik **credentials.json** z danymi uwierzytelniającymi do Google Sheets.

- Biblioteka **google-api-python-client** do komunikacji z Google Sheets.
- Biblioteki **pandas** i **scikit-learn** do przetwarzania i podziału danych.

DAG 2: data_processing_dag

Ten DAG służy do wstępnego przetwarzania danych. Zakłada się, że dane wejściowe pochodzą z arkusza Google Sheets z poprzedniego etapu. Po pobraniu danych DAG wykonuje czyszczenie, usuwanie lub imputację braków danych, a następnie standaryzację i normalizację wartości numerycznych. Ostatecznie przetworzone dane są zapisywane ponownie do arkusza Google Sheets, gotowe do dalszej analizy lub modelowania.

Kroki:

1. **fetch_data (PythonOperator)**
    - Pobiera dane z zadanego arkusza Google Sheets (SOURCE_SPREADSHEET_ID).
    - Konwertuje dane do DataFrame Pandas.
    - Sprawdza, czy dane nie są puste oraz konwertuje wartości na typu numeryczne tam, gdzie to potrzebne.
    - Zapisuje pobrane dane do XCom.
2. **clean_data (PythonOperator)**
    - Pobiera dane z XCom.
    - Wyświetla informacje o danych, aby ułatwić debugging.
    - Usuwa duplikaty.
    - Uzupełnia brakujące wartości średnią dla zmiennych numerycznych.
    - Zapisuje wyczyszczone dane do XCom.
3. **scale_and_normalize_data (PythonOperator)**
    - Pobiera wyczyszczone dane z XCom.
    - Stosuje standaryzację (StandardScaler) i normalizację (MinMaxScaler) do wybranych kolumn numerycznych.
    - Zapisuje przetworzone dane do XCom.

4. **save_processed_data (PythonOperator)**
    - Pobiera przetworzone dane z XCom.
    - Wysyła je do docelowego arkusza Google Sheets (TARGET_SPREADSHEET_ID).
    - Dane są teraz gotowe do dalszej analizy lub jako input do modelowania.

Wymagania techniczne:

- Plik **credentials.json** z danymi uwierzytelniającymi do Google Sheets.
- Biblioteka **pandas** do manipulacji danymi.
- Biblioteki **scikit-learn** (StandardScaler, MinMaxScaler) do standaryzacji i normalizacji.
- Biblioteka **google-api-python-client** do komunikacji z Google Sheets.