PROJEKT KEDRO

Poniżej zilustrowano standardową strukturę projektu Kedro z pokazanymi kluczowymi folderami i plikami.

```
example_project/
├── conf/
    ├── base/ #zawiera podstawowe konfiguracje
        ├── catalog.yml  #Rejestr zbiorów danych, które mają zostać wykorzystane w projekcie. Ten plik ma kluczowe znaczenie, ponieważ zniekształcony lub niekompletny katalog danych może uniemożliwić uruchomienie potoku.
        ├── logging.yml
        └── parameters.yml  #Plik konfiguracyjny do przechowywania niestandardowych parametrów (np. hiperparametrów, określonych ścieżek plików).
    ├── local/
        └── credentials.yml
    └── README.md
├── data/  #Lokalne dane projektu
    ├── 01_raw/
    ├── 02_intermediate/
    ├── 03_primary/
    ├── 04_feature/
    ├── 05_model_input/
    ├── 06_models/
    ├── 07_model_output/
    └── 08_reporting/
├── docs/ #Dokumentacja projektu
├── logs/ #Dane wyjściowe projektu
├── notebooks/
├── src/ #Kod źródłowy projektu
    ├── example_project/
        ├── pipelines/
        ├── hooks.py
        ├── pipeline_registry.py
        ├── settings.py
        ├── __init__.py
        └── __main__.py
    ├── tests/
    ├── requirements.txt
    └── setup.py
├── pyproject.toml
├── README.md
└── setup.cfg
```

SZCZEGÓŁY PROJEKTU

W tym projekcie zbudowany zostanie potok danych w celu wgrania, oczyszczenia, przekształcenia i innego przygotowania zestawu danych do modelowania. Zbudowany zostanie potok data_preparation (uwaga: nazwa nie ma znaczenia) i hipotetycznie inny zespół mógłby jednocześnie zbudować potok prediction_pipeline, który byłby w stanie prognozować odsłony strony Wikipedii w oparciu o dane wyjściowe naszego potoku jako dane wejściowe.

ZESTAW DANYCH

Plik CSV z wierszem zawierającym liczbę odsłon strony Wikipedii Elvisa Presleya i Michaela Jacksona dla każdej daty w okresie od 01.07.2015 do 02.05.2021 (data1) oraz 01.05.2021 do 25.02.2023 (data2).

INSTALACJA KEDRO

Instalacja Kedro w środowisku Pythona jest prosta i prosta. Bibliotekę można zainstalować na wiele sposobów, ale dwa typowe sposoby to PyPI: `pip install kedro` i Conda-Forge: `conda install -c conda-forge kedro`

BUDOWA POTOKU KEDRO

PODSTAWY

Przepływ prac programistycznych Kedro jest prosty, prosty i zorientowany na rozwiązania. Proces składa się z czterech kroków:

1. Konfiguracja projektu
2. Konfiguracja danych
3. Konfiguracja potoku
4. Konfiguracja pakietu

1. KONFIGURACJA PROJEKTU

Aby skonfigurować projekt, należy najpierw utworzyć projekt. Aby to osiągnąć, użyj następującego polecenia:

```
Kedro new
```

W oknie terminala należy podać nazwę nowego projektu.

Ustawianie wszystkich wymagań projektu: możesz modyfikować wszystkie zależności swojego projektu w pliku src/requirements.txt.

black~=22.0
flake8>=3.7.9, <5.0
ipython>=7.31.1, <8.0; python_version < '3.8'
ipython~=8.10; python_version >= '3.8'

isort~=5.0
jupyter~=1.0
jupyterlab_server>=2.11.1, <2.16.0
jupyterlab~=3.0, <3.6.0
kedro~=0.18.14
kedro-telemetry~=0.2.0
nbstripout~=0.4
pytest-cov~=3.0
pytest-mock>=1.7.1, <2.0
pytest~=7.2
Pin problematic traitlets release - https://github.com/jupyter/notebook/issues/7048
traitlets<5.10.0

Instalacja wszystkich wymagań: `pip install -r src/requirements.txt`

2. KONFIGURACJA DANYCH

Po skonfigurowaniu projektu nadszedł czas na skonfigurowanie danych dla projektu. Kedro można skonfigurować do pracy z plikami SQLTableDataSet i SQLQueryDataSet. Najpierw należy zdefiniować zbiory danych w pliku katalogu danych (conf/base/catalog.yml).

Katalog danych - conf/base/catalog.yml

```
data1:
    type: pandas.CSVDataSet
    filepath: data/01_raw/data1.csv
data2:
    type: pandas.CSVDataSet
    filepath: data/01_raw/data2.csv
```

W powyższym przykładzie zdefiniowano dwa zestawy danych, podając nazwę, typ i ścieżkę pliku.
Ponieważ dane są w dwóch plikach, należy je połączyć w potoku danych. Należy pamiętać, że daty nakładają się na siebie i będziemy musieli również usunąć lub zignorować duplikaty.

3. KONFIGURACJA POTOKU

Konfigurowanie potoku jest krokiem krytycznym i najbardziej bezpośrednio związanym z budowaniem potoków danych w kedro. Biorąc pod uwagę złożoność tego kroku, można podzielić procedurę na wiele podetapów: utworzenie obszaru roboczego potoku, dodanie węzłów, montaż węzłów w potoku i zarejestrowanie potoku.

Po odpowiednim zdefiniowaniu naszego katalogu danych można budować nasze potoki. Po pierwsze, należy zrozumieć dwa kluczowe pojęcia: **węzły** i **potoki** .

- **Węzły** są elementami składowymi rurociągów. Są to zasadniczo funkcje Pythona, które reprezentują transformacje danych, które należy wykonać, np. wstępne przetwarzanie danych, modelowanie.
- **Potoki** to sekwencje węzłów połączonych w celu dostarczenia przepływu pracy. Organizuje zależności węzłów i kolejność wykonywania oraz łączy wejścia i wyjścia, zachowując modułowość kodu.

Najpierw należy utworzyć obszar roboczy potoku, uruchamiając następujące polecenie:

```
kedro pipeline create data_preparation
```

Ten krok tworzy pliki `nodes.py` i `pipeline.py` w katalogu `src/example_project/pipelines/data_preparation/`

Kolejnym krokiem będzie dodanie węzłów i złożenie ich w potok. Wprowadzony zostanie węzeł umożliwiający łączenie zestawów danych, czyszczenie danych i dodawanie nowych funkcji. Po zdefiniowaniu funkcji zdefiniowane zostaną węzły odwołujące się do funkcji oraz określamy zbiory danych wejściowych i wyjściowych.

Węzły - src/example_project/pipelines/data_preparation/nodes.py

```
import pandas as pd

def merge_datasets(df_initial: pd.DataFrame, df_new:
pd.DataFrame) -> pd.DataFrame:
    return pd.concat([df_initial, df_new])

def clean_data(df: pd.DataFrame) -> pd.DataFrame:
    # Remove duplicates
    return df.drop_duplicates(subset=["Date"])

def generate_datetime_features(df: pd.DataFrame) ->
pd.DataFrame:

    ### Extract date time features
    df['Date'] = pd.to_datetime(df['Date'])

    # Add a column for year
    df['year_num'] = df['Date'].dt.year

    # Add a column for month
    df['month_num'] = df['Date'].dt.month

    # Add a column for day of week
    df['dayofweek_num'] = df['Date'].dt.dayofweek
```

```
    # Add a column for day of month
    df['dayofmonth'] = df['Date'].dt.day

    # Add a column for day of year
    df['dayofyear_num'] = df['Date'].dt.day_of_year

    return df
```

Potok — src/example_project/pipelines/data_preparation/pipeline.py

```
from kedro.pipeline import Pipeline, pipeline, node
from .nodes import merge_datasets, clean_data,
generate_datetime_features


def create_pipeline(**kwargs) -> Pipeline:
    return pipeline([
    node(
                func=merge_datasets,
                inputs=["data1", "data2"],
                outputs="merged_data",
                name="merge_datasets_node"
            ),
            node(
                func=clean_data,
                inputs="merged_data",
                outputs="cleaned_data",
                name="clean_datasets_node"
            ),
            node(
                func=generate_datetime_features,
                inputs="cleaned_data",
                outputs="feateng_data",
                name="generate_features_node"
            )
    ])
```

Na koniec należy zarejestrować potok i wszelkie zestawy danych, które musimy utrwalić (tj. te używane jako dane wejściowe do węzła) lub które chcemy utrwalić (tj. wszelkie wyniki na końcu potoku).
Dodatkowo wprowadzono 3 nowe zestawy danych bazujące na wynikach węzłów – należy pamiętać, że ścieżki plików wykorzystują zdefiniowane powyżej warstwy danych.

Katalog danych - conf/base/catalog.yml

```
merged_data:
  type: pandas.ParquetDataSet
  filepath: data/02_intermediate/merged_data.parquet

cleaned_data:
```

```
  type: pandas.ParquetDataSet
  filepath: data/03_primary/cleaned_data.parquet

feateng_data:
  type: pandas.ParquetDataSet
  filepath: data/04_feature/feateng_data.parquet
```

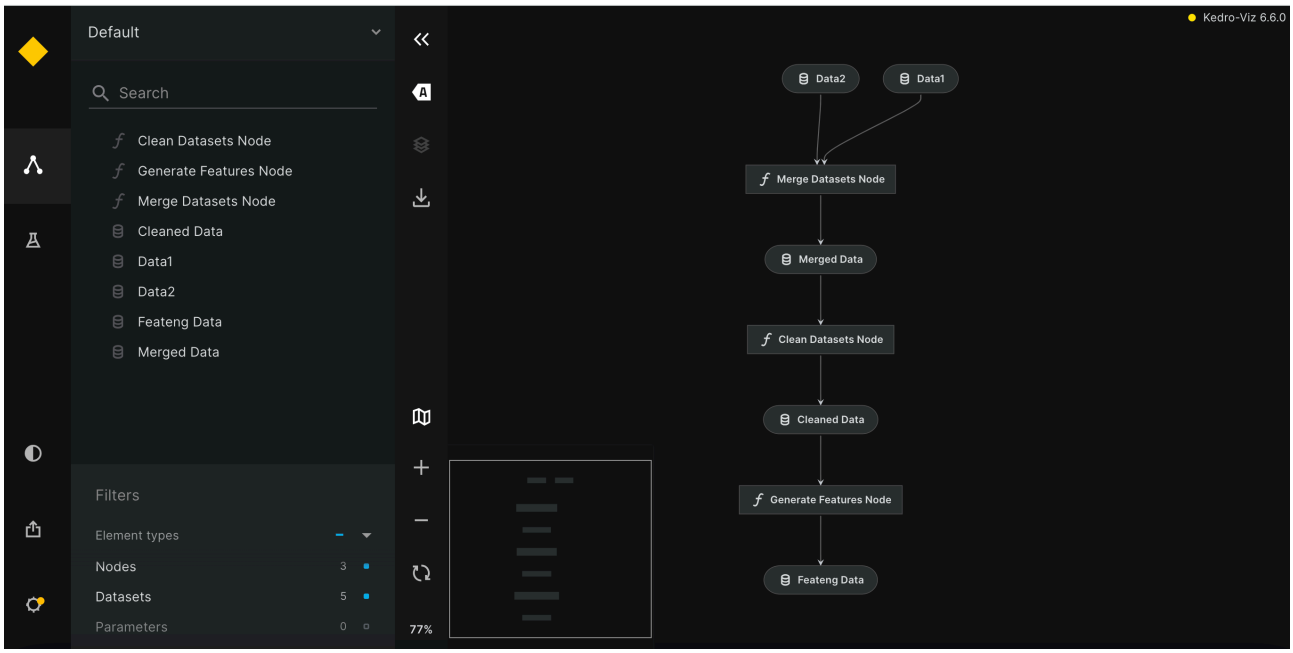
Za pomocą kedro run uruchomiony zostaje cały potok.

WIZUALIZACJA POTOKU KEDRO

Jeśli chcesz przejrzeć logikę potoku, pomocna może być wizualizacja potoku w postaci wykresu. Kedro może w tym pomóc - może być konieczna instalacja, kedro-viz jeśli jeszcze tego nie zrobiłeś.

Uruchom następujące polecenie, aby uruchomić serwer Kedro viz i wyświetlić wykres potoku:

```
kedro viz
```



Uruchamiając polecenie kedro viz, można wygenerować wizualną reprezentację swojego potoku. Otworzy się Twoja domyślna przeglądarka internetowa, a powita Cię wizualizacja Twojego potoku. Na grafie można wchodzić w interakcję z potokiem, wybierając węzeł, przeglądając kod węzła i przesuwając się po minimapie.