Prva domaća zadaća iz Podatkovnog inženjerstva

ETL & Event processing

Zadatak je napisati ETL proces koji detektira datoteke u zadanom direktoriju, provjerava datoteke, obrađuje datoteke te ih sprema. Potrebno je napisati svoju detekciju koja kao ulaz prima putanju na računalu, te vraća sve datoteke rekurzivno. Tu je potrebno proširiti IDetectionAPI interface, te implementirati svoju detekciju. Potom slijedi provjera kvalitete datoteka, te je za svaku datoteku potrebno provjeriti je li ispravno formatirana i ima li smisla koristiti je u svrhe računanja. Processing dio sastoji se od detekcije evenata nad podacima, te njihovog spremanja u pickle file. Čitav flow se treba implementirati u Prefectu, te se treba koristiti Prefect server za pokretanje flowova.

Početno stanje repozitorija

Kao ulazno stanje imamo 3 direktorija:

- data
  - input
  - output
- prefect_implementation
  - ime komponente
    - sučelje za komponentu
    - impl
      - implementacija koju se treba nadopuniti
  - ...
- processing
  - logika za procesiranje podataka, detekciju evenata, kalkulaciju kpi-eva

Zadatak

1 - detekcija i provjera kvalitete

Potrebno je nadopuniti sve implementacije sučelja koja su definirana u prefect_implementation folderu.

1.1 Detekcija datoteka

Datoteke se nalaze u data/input folderu te su sve tipa mf4. Potrebno je napraviti funkciju koja će rekurzivno proći kroz **data/input** direktorij i vratiti sve putanje do tih file-ova.

Za to je potrebno nadopuniti metodu detect_files(self) unutar klase prefect_implementation.detection.FileDetector. Ta klasa se pokreće tako da joj se pri inicijalizaciji preda lista ulaznih direktorija, te ona treba **rekurzivno proći** kroz sve direktorije i vratiti sve putanje do mf4 datoteka.

```
from prefect_implementation.detection.impl.FileDetector import FileDetector
detector = FileDetector(["data/input"])
all_files_detected = detector.detect_files()
```

1.2 Provjera kvalitete datoteka

Nakon što smo detektirali sve datoteke, potrebno je provjeriti jesu li one ispravno formatirane i imaju li smisla za daljnju obradu. Slično kao i u detekciji, potrebno je nadopuniti već postojeće sučelje u klasi prefect_implementation.quality_check.impl.QualityCheck.

```
from prefect_implementation.quality_check.impl.QualityCheck import QualityCheck
from typing import List
check = QualityCheck()
all_files_detected: List[str] = ["neki_file.mf4", "neki_drugi_file.mf4"]
successfull_result, error_result = check.check_quality(all_files_detected)
```

2 - Processing

Nakon što smo detektirali sve datoteke i provjerili njihovu kvalitetu, potrebno je napraviti procesiranje podataka. Procesiranje podataka je u ovom slučaju kalkulacija evenata (događaja). Event je odsječak vremena za koji vrijedi neki uvjet (u našem slučaju, taj je uvjet brzina u intervalu [40, 60]).

Očekivani use-case:

- detekcija evenata unutar mf4 datoteka, specifično nas interesiraju eventi gdje je iznos kanala brzine između 40 i 60 (interval je uključen sa obje strane [40, 60]).
- za svaki od evenata trebaju se izračunati standardne agregacije (min, max, avg, std) za svaki kanal.
- rezultat se mora spremiti u pickle datoteke koji će se nalaziti u data/output folderu.
- pri demonstraciji će se studente **tražiti da navedenu pickle datoteku i učitaju** tako da bi to bilo korisno za implementirati uz logiku za spremanje.

Potrebno je nadopuniti sljedeće funkcije / klase:

- processing.flow.get_events
- processing.flow.get_dataframe_from_mf4
- processing.calculators.YourCalculator
    - specificno metoda calculate_on_dataframe_chunk

3 - Prefect

Nakon što smo implementirali sve potrebne funkcije, potrebno je napraviti Prefect flow koji će pokrenuti sve potrebne funkcije. Main flows već ima čitavu logiku za detekciju, i procesiranje, međutim **potrebno je između detekcije i procesiranja ubaciti i kontrolu kvalitete koda**. Na kraju čitav flow se treba deployati kao prefect Deployment i treba se pokrenuti sa lokalnim Prefect serverom i lokalno pokrenutim prefect agentom.