Data Engineering Obuka

Projekat

Projekat je osmišljen da obuhvata ključne aspekte u oblasti Data Engineering-a:

- sakupljanje i skladištenje sirovih podataka,
- kreiranje i upravljanje tokovima podataka,
- obrada sakupljenih podataka u *batch* i *streaming* modu,
- skladištenje dobijenih rezultata obrade u odredišne baze podataka (SQL/noSQL),
- automatizacija i orkestracija procesa sakupljanja i obrade,
- tabelarna i grafička prezentacija rezultata krajnjem korisniku.

Stoga će se projekat odvijati po fazama gde će prethodna faza biti neophodna za implementaciju sledeće.

I faza - pristup izvorima podataka

Izvori podataka koji će biti korišćeni za projekat su:

- Twitter - https://developer.twitter.com/en/docs/twitter-api
- YouTube - https://developers.google.com/youtube/v3/getting-started
- Reddit - https://www.reddit.com/dev/api/

Zadaci:

- potrebno je registrovati se na zadatu plaformu i obezbediti neophodne pristupne ključeve
- uveriti se da je pristup API-ju dozvoljen slanjem HTTP zahteva putem cURL ili *Postman*-a
- ako *streaming* postoji, uveriti se da je pristup *streaming* API-ju dozvoljen konektovanjem na *WebSocket* i prijemom poruka putem *Postman*-a
- ukoliko postoje odgovarajuće *python* biblioteke, probati konekciju na API kroz njih
- napraviti pregled entiteta koji su dostupni preko API-ja zajedno se njihovim atributima

II faza - sakupljanje podataka

Nakon obezbeđivanja pristupa podacima iz zadatog izvora, podatke je potrebno preuzeti i smestiti/proslediti ih u odgovarajuće skladište.

Zadaci:

- Obezbediti bar minimalnu *docker* infrastrukturu (*docker-compose*) sa:
  - HDFS/MinIO za skladištenje preuzetih podataka
  - Kafkom za prosleđivanje poruka u realnom vremenu
  - NiFi za *data ingestion*
- Napraviti NiFi *flow* koji:
  - preuzima podatke iz izvora (putem HTTP, *WebSocket*-a, *python* skripte, …)
  - formatira ih po potrebi (u JSON, Avro, Parquet, …)
  - prosleđuje na dve strane:
    - u Kafka topic za dalju *streaming* obradu
    - na HDFS/MinIO za dalju *batch* obradu tako što formira fajlove (od nekoliko MB) od sakupljenih podataka

III faza - *batch* obrada podataka

Nakon sakupljanja i skladištenja sirovih podataka u Data Lake (HDFS ili MinIO) potrebno je izvršiti čišćenje i transformaciju sirovih podataka, kao i smeštanje u odredišno skladište/bazu podataka.

Zadaci:

- implementirati Spark aplikaciju za čišćenje nepotrebnih i nepotpunih podataka uklanjanjem odgovarajućih redova i kolona
- implementirati jednu ili više Spark aplikacija za transformaciju i agregaciju očišćenih podataka
    - po potrebi uključiti Hive, Delta Lake ili Hudi za uvođenje sloja između Spark-a i skladišta sirovih podataka
- transformisane i agregirane podatke upisati u bazu podataka (npr. Mongo ili PostgreSQL)
- obezbediti (polu)automatizovano pokretanje Spark job-ova kroz Airflow ili bash skripte sa crontab-om

IV faza - *streaming* obrada podataka

Sakupljene podatke koji su presumereni u tokove podataka potrebno je obraditi u streaming režimu. Rezultate streaming obrade neophodno je smestiti u nove tokove podataka ili odgovarajuće real-time skladište.

Zadaci:

- implementirati Kafka/Spark streaming aplikaciju za čišćenje nepotrebnih i nepotpunih podataka
- tokove podataka sa očišćenim podacima obraditi sa jednom ili više streaming aplikacija
- rezultujuće tokove podataka sliti u odgovaraću bazu podataka (npr. Druid, Mongo ili PostgreSQL)

V faza - prezentacija rezultata

Rezultate obrade neophodno je predstaviti korisniku na što prirodniji način, putem interaktivnih tabela i dijagrama.

Zadaci:

- iskoristiti neku postojeću web aplikaciju za vizuelizaciju podataka (Metabase/Superset) ili isprogramirati svoju
- uspostaviti konekciju sa bazama podataka gde su smešteni rezultati batch i streaming obrade
- omogućiti krajnjem korisniku pretraživanje podataka interaktovno putem SQL-a
- kreirati nekoliko dashboard-a sa različitim zanimljivim dijagramima koji na što bolji način daju uvid u rezultate obrade izvornih podataka