개별과제: 빅데이터 분석

컴퓨터AI공학부
천세진

과제 목표

- 다양한 데이터소스(비디오, 소셜미디어 스트림 등)로부터 실시간 데이터 처리하는 기법에 대한 단계별 과제 학습
- 수집 → 스트리밍 ETL → 피처/라벨 → 모델선정 → 실시간 서비스에 대한 이해
  - 개별 코드 스니펫을 제공함

1과제. 멀티 소스 수집 & 배치 ETL (Bronze)

- 핵심 목표: API → 원본 보존(스키마 고정, 증분 수집), 희소/폭증 구간 대응을 위해 reservoir sampling 적용
  - 삽입 포인트 A — Reservoir Sampling (스트리밍 전 단계의 대표 샘플 확보)
    - 목적: API 폭증 시 전수 저장 전 점검/데이터 프로파일링용 대표 샘플을 저비용으로 유지
    - 방법: 각 쿼리/키워드별로 크기 k의 저장소 유지 (Vitter's Algorithm R)
  - 삽입 포인트 B — Bloom Filter(원본 중복 방지의 1차 게이트)
    - 목적: API가 중복 video_id/post_id를 자주 반환하는 경우, 저장/조인 비용 절감
    - 방법: 최근 7일의 ID를 집계해 Bloom filter 생성 → 인입 시 might_contain으로 빠른 배제

2과제. 구조화 & 정제 스트리밍 (Silver)

- 핵심 목표: Sliding Window + Watermark로 지연 이벤트 처리, 중복 제거
  - 삽입 포인트 C — Sliding Window & Watermark
    - 목적: 실시간 지표(1h 윈도우/5m 슬라이드), 지연 이벤트 허용(예: 10분) + 정확한 중복 제
    - 거방법: withWatermark로 이벤트 시간 기준 허용 지연 설정 → window('1 hour','5 minutes')

3과제. 피처 엔지니어링 & 준실시간 라벨링 (Gold/Features)

- 핵심 목표: 영상+소셜 조인, Flajolet–Martin 계열 근사, CDF/PDF로 임계치 도출
  - 삽입 포인트 D — Flajolet–Martin(근사 유니크 카운트) / HLL++
    - 목적: 거대 스트림에서 고유 사용자 수를 가볍게 추정
    - Spark 내장: approx_count_distinct(col)(HLL++, FM 계열 아이디어)로 대체 가능
  - 삽입 포인트 E — 경험적 PDF/CDF 기반 라벨링(상위 p% 구분)
    - 목적: "고성능(상위 10%)" 라벨을 **데이터 분포(CDF)** 로 동적으로 결정
    - 방법: (배치/마이크로배치) 히스토그램→PDF 추정, 누적합→CDF, 상위 p% 컷오프 산출

4과제. 다수 알고리즘 벤치마킹 & **Pareto-Optimal** 모델 선정

- 핵심 목표: 다목적 최적화(성능, 지연, 해석성 등)에서 파레토 전선으로 베스트 후보 자동 선택
  - 삽입 포인트 F — Pareto Front (성능-지연-비용 다목적)
    - 목적: AUC(↑), 추론지연(ms, ↓), 피처수(↓) 등 상충 목표를 동시에 고려
    - 방법: 각 후보 모델의 지표 벡터에 대해 지배관계 비교로 Pareto Front 추출

5과제. 실시간 예측/조회 Streamlit

- 핵심 목표: 사용자가 입력한 키워드/영상ID에 대해, 최신 예측 + 분포(CDF/PDF) 맥락과 함께 보여주기
  - 삽입 포인트 G — CDF/PDF 시각화, Bloom Filter로 쿼리 사전 점검
  - CDF/PDF: 현재 engagement_24h의 분위(예: 상위 10%) 기준선을 함께 표시
  - Bloom Filter: 입력한 video_id가 "최근 7일 데이터에 존재할 가능성"을 즉시 피드백
  - (선택) "Top-K 상승률" 카드에 Sliding Window(1h/5m) 집계 결과 반영

폴더 안내

- 설정 파일: .env.example, app.yaml.example, logging.yaml
- 라이브러리: libs/
  - session.py(Spark 세션/설정 로더), sampling.py(reservoir),
  - metrics.py(CDF/PDF 컷/라벨),
  - pareto.py(Pareto 전선), io.py(NDJSON/샘플 저장)
- 잡 스크립트:
  - 00_fetch_to_landing.py — 모의 API 페치 + reservoir 샘플 저장
  - 10_bronze_batch.py — Bloom 필터 기반 중복 차단 후 Delta 적재
  - 20_silver_stream.py — 파일 소스 스트리밍 + sliding window & watermark 집계
  - 30_gold_features.py — HLL(approx_count_distinct) + CDF 라벨
  - 40_train_pareto.py — 다모델 학습 + Pareto-front 산출
  - 50_predict_stream.py — (옵션) 간단 예측 append
- Streamlit 앱: app/app.py — CDF/PDF 시각화, Bloom 존재 가능성 체크, 테이블 뷰
- 스크립트: 초기화/데이터 생성/앱 실행 (reset_all.sh, make_stream_data.sh, run_streamlit.sh)
- 의존성: requirements.txt (pyspark, delta-spark, mlflow, streamlit, 등)
- README: 퀵스타트와 전체 실행 순서 포함