I.빅데이터 분석 기획

03. 데이터 수집 및 저장 계획

3.1 데이터 수집 및 전환

-	**KeyWord**
데이터 수집	데이터 처리, 데이터 수집, ETL, FTP, Sqoop, Crawling, RSS, Scrapy, Apache Kafka, Flume, Scribe, Chukwa
데이터 유형 및 속성 파악	데이터 속성, 데이터 측정 척도, 명목척도, 순서척도, 등간척도, 비율척도
데이터 변환	평활화, 집계, 일반화, 정규화
데이터 비식별화	데이터 보안관리, 비식별화, 가명처리, 총계처리, 데이터값 삭제, 범주화, 데이터 마스킹, 적정성 평가
데이터 품질검증	데이터 품질검증

<데이터 처리 기술>

- 필터링
  - 목적에 맞지 않는 정보를 필터링하여 분석시간 및 저장공간을 효율적으로 사용
  - 정형 데이터: 오류 발견 / 보정 / 삭제 / 중복성 검사 등
  - 비정형 데이터: 자연어처리, 기계학습과 같은 추가 기술 적용 -> 오류 및 중복과 같은 저품질 데이터 필터링
- 변환
  - 분석하기 쉽도록 일관성 있는 형식으로 변환
  - 평활화 / 집계 / 일반화 / 정규화 / 속성 생성 기술을 사용
- 정제
  - 데이터의 불일치성을 교정하기 위함
  - 결측값 처리 / 잡음(Noise) 처리 기술을 활용
- 통합
  - 출처는 다르지만 상호연관성이 있는 데이터들을 하나로 결합
  - 연관 관계 분석 등을 통해 중복 데이터 검출 필요
- 축소
  - 분석에 불필요한 데이터 축소
  - 데이터의 고유한 특성은 손상되지 않도록해 분석 효율성 향상시킴

==========

1) 데이터 수집

(1) 데이터 수집 프로세스

- 수집 데이터 도출 -> 목록 작성 -> 데이터 소유기관 파악 및 협의 -> 데이터 유형 분류 및 확인(포맷 등) -> 수집 기술 선정(포맷에 맞게) -> 수집 계획서 작성 -> 수집 주기 결정 -> 데이터 수집 실행
- 수집 데이터 도출: 빅데이터 서비스 제공 시 서비스의 품질을 결정하는 중요한 핵심 업무
- 목록 작성: 수집 가능성 / 보안 / 정확성 / 수집 비용 등을 검토
- 수집 기술 선정: 다양한 유형의 데이터 수집을 위해 확장성 / 안정성 / 실시간성 / 유연성 확보 필요
- 수집 주기 결정: 데이터 유형에 따라 배치 or 실시간 방식을 적용

(2) 수집 데이터의 대상

- 데이터 위치에 따라 외부 or 내부 데이터로 구분
- 내부 데이터
    - 조직(인프라) 내부의 데이터, 주로 수집하기 쉬운 정형 데이터
    - 서비스 수명 주기 관리 용이
    - 서비스: SCM / ERP / CRM / 포털/ 인증 시스템 / 거래 시스템 등
        - SCM(Supply Chain Management, 공급사슬관리)
            - 부품 제공업자로부터 생산자, 배포자, 고객에 이르는 물류 흐름을 하나의 가치사슬 관점에서 파악하고, 필요한 정보가 원활히 흐르도록 지원하는 시스템
            - 기업이 외부 공급업체 또는 제휴업체와 통합된 정보시스템으로 연계하여 시간과 비용을 최적화시키기 위한 것
            - 자재구매, 생산/재고, 유통/판매, 고객데이터로 구성됨
        - ERP(Enterprise Resource Planning, 전사적 자원 관리)
            - 회사의 모든 정보, 공급사슬관리, 고객 주문정보까지 포함하여 통합적으로 관리하는 시스템
        - CRM(Customer Relationship Management, 고객 관계관리)
            - 소비자들을 자신의 고객으로 만들고 장기간 유지하고자 하는 경영방식
            - 고객에 대한 정보를 분석/저장하는 데에 사용하는 넓은 분야를 아우름
    - 네트워크: 백본 / 방화벽 / 스위치 / IPS / IDS 등
    - 마케팅: VOC 접수/ 고객 포털 시스템 등
- 외부 데이터
    - 조직(인프라) 외부의 데이터, 주로 수집하기 어려운 비정형 데이터
    - 소셜: SNS / 커뮤니티 / 게시판
    - 네트워크: 센서 데이터 / 장비 간 발생로그(M2M; Machine 2 Machine)
    - 공공: 정부에서 공개한 공공 데이터(LOD; Linked Open Data)

(3)데이터 수집 방식 및 기술

- 데이터 유형에 따라 적합한 방식이 다름(정형 / 비정형 / 반정형)
- 정형 데이터 수집 방식 및 기술
    - ETL / FTP / API / DBToDB / Rsync / Sqoop
    - ETL: 추출(Extract), 변환(Transform), 적재(Load)
        - 데이터를 추출, 변환하여 데이터 웨어하우스(DW) 및 데이터 마트(DM)에 저장하는 기술
        - 데이터 조회 or 분석을 목적으로 적절한 포맷 or 구조로 데이터를 변환
        - 데이터 웨어하우스(Data Warehouse): DB에 축적된 데이터를 공통 형식으로 변환해서 관리하는 저장소, 여기서 관리하는 데이터들은 시간의 흐름에 따라 변화하는 값을 유지
        - 데이터 마트(Data Mart): DW에서 데이터를 꺼내 사용자에게 제공하는 역할, 특정 사용자가 관심을 가지고 있는 데이터를 담은 비교적 작은 규모의 DW(재무 / 생산 / 운영 등과 같이 특정 조직의 특정 업무 분야에 초점을 맞추어 구축)

    - FTP: File Transfer Protocol
        - 파일 송수신 응용계층 통신 프로토콜
        - 원격지 시스템 간 파일 공유를 위한 서버-클라이언트 모델
        - TCP/IP 프로토콜을 기반으로 서버, 클라이언트 사이에서 파일 송수신
        - Active FTP: 클라이언트가 포트를 알려주면 데이터 전송해주는 방식
        - Passive FTP: 서버가 포트를 알려주면 데이터 가져가는 방식
    - API: Application Programming Interface
        - 실시간 데이터 수신 기능을 제공하는 인터페이스 기술
        - 솔루션 제조사 및 3rd party 소프트웨어로 제공되는 도구
    - DBToDB: 데이터베이스 시스템 간 데이터 동기화 및 전송 기능
    - Rsync: Remote Sync/ 서버-클라이언트 방식/ 수집 대상 시스템과 일대일로 파일 및 디렉터리를 동기화
    - 스쿱 (Sqoop)
        - 커넥터를 사용하여 RDBMS와 하둡 간 데이터 전송
        - 전송, 수집 등 모든 적재 과정이 자동화/ 병렬 처리 방식
        - 특징: 벌크 임포트 지원(한번에 전송) / 데이터전송 병렬화 / 직접입력 제공 / 프로그래밍방식의 데이터 인터랙션
        - 툴: Import(가져오기) / Export(내보내기) / Job(생성 및 실행) / Metastore / Merge(데이터셋 병합)

- 비정형 데이터 수집 방식 및 기술
    - 크롤링 / RSS / Open API / 스크래파이 / 아파치 카프카
    - 크롤링(Crawling): 웹 사이트로부터 콘텐츠를 수집
    - RSS(Rich Site Summary): XML 기반으로 정보를 배포하는 프로토콜을 활용한 데이터 수집 기술
    - Open API: 응용 프로그램을 통해 실시간 데이터 수신
    - 스크래파이(Scrapy)
        - Python 기반 비정형 데이터 수집 기술
        - 웹 사이트를 크롤링하여 구조화된 데이터 수집
        - 주요기능: Spider / Selector / Items / Pipelines / Settings
    - 아파치 카프카(Apache Kafka)
        - 대용량 실시간 로그 처리를 위한 분산 스트리밍 플랫폼
        - 레코드 스트림을 발행, 구독하는 방식(기존 메시징 시스템과 유사)
        - 특징: 신뢰성 / 확장성 제공(수평 확장 및 분산 처리 가능)
        - 주요기능: 소스(수집영역) / 채널(소스와 싱크간 버퍼구간) / 싱크(전달 및 저장) / 인터프리터(가공)

- 반정형 데이터 수집 방식 및 기술
    - 센싱 / 스트리밍 / 플럼 / 스크라이브 / 척와
    - 센싱(Sensing): 센서 데이터 → 네트워크로 수집 및 활용
    - 스트리밍(Streaming)
        - 네트워크로 미디어 데이터를 실시간 수집(센서, 오디오, 비디오 등)
        - 플럼/ 스크라이브/ 척와의 활용은 점차 증가하는 중
    - 플럼(Flume)
        - 분산형 대용량 로그 수집 기술 / 이벤트, 에이전트 활용
        - 발행/구독 모델: 풀(Pull) 방식으로 부하 감소, 고성능 기능 제공

        - 풀(Pull) 방식: 사용자가 자신이 원하는 정보를 요청할 때, 서버에서 정보를 전송하는 기법
        - 푸시(Push) 방식: 사용자가 요청하지 않아도 자동으로 원하는 정보를 제공하는 기법
      - 고가용성(High Availability) 제공: 클러스터 구성, 분산 처리를 통해 고가용성 서비스 제공 가능
      - 파일 기반 저장방식: 데이터를 디스크에 순차 저장
      - 주요 기능: 소스(이벤트를 전달하는 컨테이너) / 채널(이벤트 전달 통로) / 싱크(이벤트 저장 및 전달)
    - 스크라이브(Scribe)
      - 대용량 실시간 로그 수집 기술
      - 다수의 서버로부터 실시간 스트리밍 로그 데이터를 수집하여 분산 시스템에 저장
      - 데이터 수집 다양성: 클라이언트 서버 타입에 상관없이 로그 수집 가능
      - 고가용성: 단일 중앙 서버 장애 -> 다중 로컬 서버에서 저장
    - 척와(Chukwa)
      - 대규모 분산 시스템 모니터링을 위한 에이전트-컬렉터 구성의 데이터 수집 기술
      - 분산된 서버에서 에이전트 실행 -> 컬렉터가 데이터 수집 -> HDFS에 저장 및 실시간 분석 기능 제공
      - 특징: HDFS 연동 / 실시간 분석 / 청크 단위 처리
      - 구성: 에이전트(데이터 수집) / 컬렉터(데이터를 주기적으로 HDFS에 저장)
      - 데이터 처리: 아카이빙(Archiving, 시간 순서로 그룹핑) / 디먹스(Demux, 키-값 쌍으로 척와 레코드 생성 및 저장)
      - 척와는 비정형, 반정형 데이터 수집 둘 다 사용됨

- 데이터 수집 방식 및 기술 요약

정형 데이터 수집	비정형 데이터 수집	반정형 데이터 수집
- ETL: 추출 / 변환 / 적재 - FTP: 파일 송수신 프로토콜 - API: 실시간 데이터 수신 인터페이스 - DBToDB: 데이터베이스간 동기화 - Rsync: 일대일 동기화 - Sqoop: RDBMS와 하둡간 데이터 전송	- 크롤링: 웹사이트에서 데이터 수집 - RSS: XML기반 프로토콜 활용 - Open API: 실시간 데이터 수신 - 스크래파이: 파이썬 기반 크롤링 - 아파치 카프카: 대용량 실시간 로그 처리	- 센싱: 센서 데이터- 스트리밍: 미디어 실시간 수집 - 플럼: 분산형 대량 로그 수집 기술 - 스크라이브: 대량 실시간 로그 수집 기술 - 척와: 대규모 분산 시스템 모니터링

==========

2) 데이터 유형 및 속성 파악

(1) 데이터 유형

- 구조 / 시간 / 저장 형태 관점에 따라 분류
- 구조 관점
  - 정형 / 비정형 / 반정형스키마 구조 또는 연산 가능 여부에 따라 분류
  - 정형 데이터

        - 스키마(형태) 구조 기반 형태 / 고정된 필드에 저장 / 일관성 O / 칼럼, 로우 구조
        - 관계형 데이터베이스(RDB), 스프레드시트(SpreadSheet) / ERP / CRM / SCM
    - 반정형 데이터
        - 스키마 구조 형태 가짐 / 메타데이터 포함 / 일관성 X
        - XML / HTML / 웹 로그 / 시스템 로그 / 알람 / JSON / RSS / 센서 데이터
    - 비정형 데이터
        - 스키마 구조 형태 X / 고정된 필드 X
        - SNS / 웹 게시판 / 텍스트 / 이미지 / 오디오 / 비디오

- 시간 관점
    - 실시간 / 비실시간시간 관점 또는 활용 주기에 따라 분류
    - 실시간 데이터
        - 생성된 이후 수 초 ~ 수 분 이내에 처리되어야 의미있는 현재 데이터
        - 센서 데이터 / 알람 / 시스템 로그 / 네트워크 장비 로그 / 보안 장비 로그
    - 비실시간 데이터
        - 생성된 이후 수 시간 or 수 주 이후에 처리되어야 의미있는 과거 데이터
        - 통계 / 웹 로그 / 서비스 로그 / 구매 정보 / 디지털 헬스케어 정보

- 저장 형태 관점
    - 파일/ 데이터베이스/ 콘텐츠/ 스트림 데이터
    - 파일 데이터
        - 파일 형식으로 저장 / 크기가 대용량 or 개수가 다수인 데이터
    - 데이터베이스 데이터
        - 데이터 종류 or 성격에 따라 데이터베이스의 컬럼 또는 테이블 등에 저장된 데이터
        - 관계형 데이터베이스(RDBMS) / NoSQL / 인메모리 데이터베이스
    - 콘텐츠 데이터
        - 개별적 객체로 구분될 수 있는 미디어 데이터
        - 텍스트 / 이미지 / 오디오 / 비디오 등
    - 스트림 데이터
        - 네트워크를 통해 실시간 전송되는 데이터
        - 센서 데이터 / HTTP 트랜잭션 / 알람 등

(2) 데이터 속성 파악

- 수집 데이터 종류: 정형/ 반정형/ 비정형
    - 정형: 고정된 컬럼 / 행열에 의해 속성 구별 / 스키마를 지원함
    - 반정형: 정형 데이터의 스키마에 해당하는 메타데이터를 가짐
    - 비정형: 대표적으로는 텍스트 데이터나 멀티미디어 데이터
- 데이터 형태에 따른 분류: 정성적 / 정량적
    - 정성적: 언어, 문자 형태 / 저장, 검색, 분석에 많은 비용 소모
    - 정량적: 수치, 도형, 기호 형태 / 정형화된 데이터이므로 비용 소모 적음

- 데이터 속성 파악
    - 범주형(Categorical, 질적변수)
        - 특성에 따라 범주로 구분하여 측정되는 변수
        - 연산 불가 / 각 범주에 속한 개수, 퍼센트를 다룸 / 원그래프, 막대그래프 등
        - 명목형(Nominal)

        - 명사형 / 순서없음 / 이름만 의미를 부여함 -> 같다(=), 다르다(!=)만 가능
    - 순서형(Ordinal)
        - 순서가 의미를 부여함 (ex. 상태 양호=3, 보통=2, 나쁨=1) -> 대소관계(<, >)만 비교 가능

- 수치형(Measure, 양적변수)
    - 양적인 수치로 측정되는 변수
    - 연산 가능 / 히스토그램, 시계열그래프 등
    - 이산형(Discrete)
        - 하나하나 셀 수 있음 (ex. 맞은 문제 개수, 방문 횟수)
    - 연속형(Continuous)
        - 변수가 구간 안의 모든 값을 가질 수 있는 경우 (ex. 키, 몸무게)
        - ※ 나이: 시간이 지나면서 계속 늘어나는 연속형 변수지만, 1년 단위로 측정한다면 이산형 변수

- 데이터 속성에 대한 측정 척도
    - 범주형 변수는 명목, 서열, 등간 척도
    - 수치형 변수는 비율, 간혹 등간 척도

    - 명목 척도
        - 임의의 범주로 분류 -> 기호나 숫자를 부여(분류의 수치화)
        - 척도 값은 "분류"의 의미만 가짐
        - (ex) 혈액형, 지역 번호, 출신 국가, 직업 구분 등
        - 같다, 다르다만 비교 가능

    - 서열 척도/ 순위 척도
        - 임의의 기준에 따라 상대적인 비교 및 순위화
        - 척도 값이 분류와 서열 순서를 가짐 / 수치의 크기나 차이는 의미 없음
        - (ex) 맛집 별점, 선호도 조사, 이용자 등급 등
        - 대소 관계만 비교 가능

    - 등간 척도/ 간격 척도/ 거리 척도
        - 비계량적인 변수를 정량적인 방법으로 측정
        - 각 대상을 별도로 평가 / 동일 간격화로 크기 간 차이를 비교할 수 있음
        - (ex) 온도, 미세먼지 수치, 당뇨 수치, 5점 척도 등
        - 순서뿐만 아니라 간격도 의미 있음

    - 비율 척도
        - 균등 간격 / 절대 영점 있음 / 비율 계산 가능한 척도
        - 순서 의미 있음 / 간격 의미 있음 / 사칙연산 가능

    - (ex) 나이, 키, 금액, 거리, 넓이, 소득, 부피, 질량 등
    - 속성 값을 연산했을 때 의미 있으면 비율척도, 의미 없으면 등간척도

변수 유형	척도 유형	범주	순위	같은 간격	절대 영점
범주형	명목척도	○			
범주형	순위척도	○	○		
연속형	등간척도	○	○	○	
연속형	비율척도	○	○	○	○

==========

3) 데이터 변환

(1) 데이터 저장 전처리 절차

- 데이터 저장 전처리 절차
    - 데이터 저장 전, 후로 활용 목적에 맞도록 적절한 처리가 필요
    - 데이터 저장관리: 데이터 전/후처리 -> 저장 -> 보안관리 -> 품질관리
- 데이터 저장 전·후처리 시 고려사항
    - 전처리: 데이터 유형 분류 기준을 적용할 수 있는 기능 / 데이터 변환 기능 / 변환 여부 확인 기능 / 변환 실패 시 재시도 및 취소 기능/ 변환된 데이터 저장 기능을 제공해야 함
    - 후처리: 이상값 변환 또는 자동 추천 기능 / 집계 시 데이터 요약 기능 / 변환, 패턴, 이벤트 감시 기능 / 변환 로그 저장 관리 기능을 제공해야 함
- 데이터 처리 방식 선정
    - 전처리 단계: 수집한 데이터를 저장하기 위한 작업(데이터 필터링 / 유형 변환 / 정제 등의 기술 활용)
    - 후처리 단계: 저장된 데이터를 분석하기 좋게 가공하는 작업(변환 / 통합 / 축소 등의 기술 활용)

(2) 데이터 변환 기술

- 데이터 변환: 데이터의 특정 변수를 정해진 규칙에 따라 바꿔주는 것
- 데이터 변환 기술: 평활화 / 집계 / 일반화 / 정규화 / 속성 생성
    - 평활화(Smoothing)
        - 잡음 제거를 위해 추세에서 벗어나는 값들을 변환
        - 구간화, 군집화 → 거칠게 분포된 데이터를 매끄럽게 만듦
    - 집계(Aggregation)
        - 다양한 방법으로 데이터를 요약
        - 복수 개의 속성을 하나로 줄임
        - 유사한 데이터 객체(Data Object) 줄이고 스케일 변경
    - 일반화(Generalization)
        - 특정 구간에 분포하는 값으로 스케일 변화
        - 특정 데이터가 아니라, 범용적인 데이터에 적합한 모델을 만드는 기법
        - 이상값, 노이즈에 크게 영향받지 않아야 잘된 일반화
    - 정규화(Normalization)

        - 정해진 구간 내에 들도록 함
        - 최단 근접 분류, 군집화와 같은 거리 측정 등을 위해 유용함
        - 최소-최대 정규화 / z-score 정규화 / 소수 스케일링 등
    - 속성 생성(Attribute/Feature Construction)
        - 데이터 통합을 위해 새로운 속성 or 특징을 만듦
        - 여러 데이터의 분포를 대표할 수 있는 새로운 속성/특징을 활용
        - 선택한 속성을 하나 이상의 새 속성으로 대체

- 정규화 기법 3가지
    - 최소-최대 정규화: 최솟값 0, 최댓값 1, 다른 값들은 0과 1사이의 값으로 변환
    - Z-스코어 정규화
        - 데이터가 평균 대비 몇 표준편차만큼 떨어져 있는지 점수화
        - 이상값 문제를 피하는 정규화 전략
    - 소수 스케일링: 특성값의 소수점을 이동하여 데이터 크기 조정

==========

4) 데이터 비식별화

(1) 데이터 보안 관리

- 수집 데이터 보안
    - 개인정보 / 데이터 연계 / 빅데이터 보안 관점에서 고려해야 함
    - 개인정보 보안 관점: 개인정보가 포함되어 있을 경우 삭제 혹은 비식별 조치
    - 데이터 연계 보안 관점: 다양한 데이터의 연계 처리 시 보안 취약점 제거
    - 빅데이터 보안 관점: 데이터 흐름에 대한 보안 고려/ 암호화를 통해 유출 시 무결성 유지 필요
- 빅데이터 수명 주기별 보안 관리
    - 데이터 수집 / 저장 / 분석 단계별 보안 관리 필요
    - 수집 보안 관리
        - 데이터 수집 기술 취약성 / 수집 서버 및 네트워크 보안 / 개인정보 및 기밀정보 유출 방지
    - 저장 보안 관리
        - 데이터 저장소 취약성 / 보안 등급 분류 / 보안 모니터링
        - 보안 등급: 기밀 수준(Confidential) / 민감 수준(Sensitive) / 공개 수준(Public) 등
    - 분석 보안 관리
        - 내부 사용자는 데이터 유출 방지 / 외부 침입자의 유출행위 차단 / 접근기록 등의 보안로그 관리
        - 분석가의 윤리의식 중요 / 분석목적에 따라 분석가의 접근권한, 접근통제 등을 관리해야 함
- 빅데이터 보안 대응 방안
    - 빅데이터 수명 주기 전반(수집 -> 저장 -> 분석 -> 활용)에 걸쳐서 보안 적용 방안 고려
    - 개인정보 처리 / 사용자 인증 / 접근 제어 / 암호화 / 보안 모니터링 / 보안 인프라 등을 수명 주기에 따라 관리해야 함

(2) 데이터 비식별화

- 데이터 비식별화

    - 수집된 개인정보의 일부 or 전부를 삭제 or 다른 정보로 대체하여 다른 정보와 결합해도 특정 개인 식별이 어렵도록 만듦
- 데이터 비식별화 적용 대상
    - 그 자체로 개인을 식별할 수 있는 정보: 이름 / 생년월일 / 사진 / 주민등록번호 / 여권번호 / 생체정보 / 계좌번호 등
    - 다른 정보와 함께 결합하여 개인을 알아볼 수 있는 정보: 성별 / 나이 / 국적 / 신체특성 / 신용특성 / 경력특징 등
- 데이터 비식별화 처리 기법
    - 가명처리/ 총계처리/ 데이터값 삭제/ 범주화/ 데이터 마스킹
      -> 데이터 활용성 고려하여 기법을 선택

< 데이터 비식별화의 처리 기법에 활용되는 세부 기술 >

- 가명처리(Psuedonymisation)

    - 다른 값으로 대체 -> 완전 비식별화 가능 / 데이터 변형 수준 낮음 / 분석에 한계 존재
      -> 세부기술: 휴리스틱 익명화 / K-익명화 / 암호화 / 교환방법
        - 휴리스틱 익명화 (Heuistic Anonymization): 정해진 규칙에 따라서 or 사람의 판단에 따라서 개인정보 숨김
        - K-익명화 (K-anonymity): 같은 속성값 가지는 데이터를 K개 이상으로 유지, 지정된 속성이 가질 수 있는 값을 K개 이상으로 유지
        - 암호화(Encryption): 일정 규칙의 알고리즘을 적용하여 암호화하여 대체, 복호화 값(key)에 대한 보안 방안도 함께 필요
        - 교환방법(Swapping): 추출된 표본 레코드에 대해 교환

- 총계처리(Aggregation)

    - 통계값 적용 -> 통계분석용 데이터셋 작성에 유리 / 정밀한 분석 어려움
      -> 세부기술: 기본 방식 / 부분집계 / 라운딩 / 데이터 재배열
        - 총계처리 기본 방식: 데이터 집합 or 부분적으로 총합 or 평균 처리
        - 부분 집계(Micro Aggregation): 부분 그룹만 처리 (다른 속성값에 비해 오차범위가 큰 항목 등)
        - 라운딩(Rounding): 올림 or 내림 기준을 적용
        - 데이터 재배열(Rearrangement): 기존 정보값은 유지 / 개인정보와 연관되지 않도록 재배열 개인 정보와 타인 정보가 뒤섞임 -> 전체 정보의 손상없이 비식별 처리

- 데이터값 삭제(Data Reduction)

    - 특정 데이터값을 삭제 -> 분석 다양성 / 결과의 유효성 / 신뢰성 저하 가능성
      -> 세부기술: 속성값 삭제/ 속성값 부분 삭제/ 데이터 행 삭제/ 준 식별자 제거를 통한 단순 익명화
        - 속성값 삭제(Reducing Variables): 개인식별항목 단순 제거
        - 속성값 부분 삭제(Reducing Partial Variables): 일부 값 삭제 -> 대표성을 가진 값으로 보이도록 함
        - 데이터 행 삭제(Reducing Records): 민감한 속성값을 가진 개인정보 내용 전체를 제거함
        - 준식별자 제거: 식별자 뿐만 아니라 준 식별자를 모두 제거 -> 프라이버시 침해 위협 줄임

- 범주화(Data Suppression)
    - 범주화(대표값 변환) or 범위화(구간값 변환) -> 정확한 수치 분석은 어려움
      -> 세부기술: 기본 방식 / 랜덤 올림 / 제어 올림 / 범위 방법 / 세분 정보 제한
        - 범주화 기본 방식(은폐화): 평균 or 범주의 값으로 변환 → 명확한 값을 숨김
        - 랜덤 올림(Random Rounding): 임의의 수 기준으로 올림(Round up) or 절사(Round down)
        - 제어 올림(Controlled Rounding): 랜덤 올림의 단점 해결 -> 행과 열이 맞지 않는 것을 제어하여 일치시킴
        - 범위 방법(Data Range): 해당 값의 분포(범위, 구간)으로 표현
        - 세분 정보 제한 방법(Sub-divide Level Controling): 민감 항목, 높은 시각 항목을 상한, 하한 코딩, 구간 재코딩

- 데이터 마스킹(Data Masking)
    - 전체 or 부분적으로 대체값으로 변환 -> 완전비식별화 가능 / 원시데이터 구조변형 적음
      -> 세부기술: 임의 잡음 추가 / 공백과 대체
        - 임의 잡음 추가 방법(Adding Random Noise): 임의의 숫자 등의 잡음을 더하거나 곱하여 노출 방지
        - 공백(Blank)과 대체(Impute) 방법: 비식별 항목을 공백으로 바꿈 -> 대체법 적용하여 공백을 채움

(3) 개인정보 비식별 조치 가이드라인

- 개인정보 비식별 조치 가이드라인
    - 정보 일부 or 전부를 삭제 or 대체하거나 다른 정보와 쉽게 결합하지 못하도록 하여 특정 개인을 알아볼 수 없도록 하는 수행지침
- 단계별로 조치 기준 있음(사전검토 -> 비식별 조치 -> 적정성 평가 -> 사후 관리)
- 사전 검토: 개인정보 해당 여부 검토
- 비식별 조치
    - 식별자 조치 기준: 식별자는 원칙적으로 삭제
    - 속성자 조치 기준: 이용 목적과 관련없는 속성자도 원칙적으로 삭제
    - 비식별 조치 방법: 여러 조치 방법을 단독 or 복합적으로 활용

- 적정성 평가
    - 기초 자료 작성 -> 평가단 구성(3명 이상) -> 평가 수행 -> 추가 비식별 조치 -> 데이터 활용
    - 평가 수행: 프라이버시 보호 모델을 활용하여 비식별 수준 적성성 평가
    - k-익명성: 주어진 데이터 집합에서 준식별자 속성들이 동일한 레코드가 적어도 k개 존재하도록 하는 모델
    - l-다양성: k-익명성의 동질성 문제, 배경지식의 문제를 극복하여 익명성을 향상시키는 보완기술

    - t-근접성: (동질 집합에서 민감정보의 분포)와 (전체 데이터 집합에서 민감정보의 분포)가 유사한 차이를 보이게 하는 모델

- 사후 관리: 비식별 정보 안전조치/ 재식별 가능성 모니터링

==========

5) 데이터 품질 검증

(1) 데이터 품질 특성

- 유효성 & 활용성
- 데이터 유효성: 정확성 / 일관성으로 정의
    - 데이터 정확성: 정확성 / 사실성 / 적합성 / 필수성 / 연관성
    - 데이터 일관성: 정합성 / 일치성 / 무결성
- 데이터 활용성: 유용성 / 접근성 / 적시성 / 보안성으로 정의
    - 데이터 유용성: 충분성 / 유연성 / 사용성 / 추적성
    - 데이터 보안성: 보호성/ 책임성/ 안정성

(2) 데이터 변환 후 품질 검증 프로세스

- 수집 데이터 분석 프로세스: 빅데이터 수집 -> 메타데이터 수집 -> 메타데이터 분석 -> 데이터 속성 분석
    - 메타데이터 수집: 테이블 정의서 / 컬럼 정의서 / 도메인 정의서 / 데이터 사전 / ERD(ER-Diagram) 등
    - 메타데이터를 통한 데이터 속성(유효성) 분석 방안
        - 누락값 분석: NULL / 공백 / 숫자 0 의 분포 확인
        - 값의 허용 범위 분석: 해당 속성의 도메인 유형에 따라서 범위 결정
        - 허용 값 목록 분석: 허용 값 목록, 집합에 포함되지 않는 값을 발견
        - 문자열 패턴 분석: 컬럼 속성값의 특성을 문자열로 도식화 -> 특성을 파악하기 쉽게 해 놓은 표현 방법
        - 날짜 유형 분석: DATETIME 유형, 문자형 날짜 유형을 활용
        - 유일 값 분석: 유일해야 하는 컬럼에 중복이 있는지 확인
        - 구조 분석: 관계 분석 / 참조 무결성 분석 / 구조 무결성 분석기 등을 활용하여 구조 결함 발견
            - 참조 무결성(Referential Integrity): 관계형 데이터베이스 모델에서 참조 관계에 있는 두 테이블의 데이터가 항상 일관된 값을 가지도록 유지되는 것

- 데이터 유효성 여부를 검증할 수 있는 규칙 설정 기능 개발 -> 일반적으로 정형 데이터에 대해 수행
- 정규표현식을 활용한 검증 수행 -> 값 유무, 중복 여부 검증 외에도 / 데이터 양식, 규칙을 적용할 수 있음

표현기호	기능	예시
\	특수 문자 표기	\t(탭), \s(스페이스), \d (숫자)
|	OR	a|b -> a 혹은 b가 존재하면 참
^	시작	^abc -> abc로 시작하는 문자열 등장
$	종료	def$ -> def로 종료되는 문자열 등장

표현기호	기능	예시
( )	묶음 처리	a(bc)+ -> a 뒤에 bc가 1번 이상 등장
[ ]	▢ 에 있는 문자열 중 1개와 매칭	[a-d] → a, b, c, d 중 1개 이상 등장
*	0개 이상의 문자열 매칭	a(bc)* -> a 뒤에 bc가 0번 이상 등장
+	1개 이상의 문자열 매칭	d(ef)+ -> d 뒤에 ef가 1번 이상 등장
{n}	n개 이상의 문자열 매칭	＼s{1,3} → 공백이 1번 이상 3번 이상 등장

(3) 품질 검증 방안

- 빅데이터 수집 시스템의 요구사항 관련 자료 수집: 수집 단계에서 품질관리를 해야 하는 요건 도출
- 수집된 빅데이터의 특성을 고려한 품질 검증 기준 정의
    - 수집 데이터의 복잡성 / 완전성 / 유용성 등에 대한 품질 검증 기준 정의
    - 복잡성 기준 정의: 데이터 구조 / 형식 / 자료 / 계층 측면에서 정의함
    - 완정성 기준 정의: 메타데이터 / 개체 단위 / 변수 정의 등을 기준으로 질이 충분하고 완전한지
    - 유용성 기준 정의: 처리 용이성 / 자료 크기 / 하드웨어 및 소프트웨어의 제약 사항 측면에서 정의
    - 시간적 요소 및 일관성 기준 정의: 시간적 요소/ 일관성/ 타당성/ 정확성을 기준으로 품질 관리
        - 시간적 요소: 수집 기간/ 수집방법의 변화가 과거 자료 사용에 제약을 주는지 여부 등
- 데이터 변환 수 빅데이터 품질 검증 기준에 따라 검증 수행 -> 검증 후 잘못된 데이터는 다시 변환하여 저장

2.2 데이터 적재 및 저장

-	KeyWord
데이터 적재	데이터 적재, 데이터 적재 아키텍처, 서버 노드 아키텍처, 데이터 아키텍처, 네트워크 아키텍처, 플루언티드
데이터 저장	빅데이터 저장 시스템, 분산 파일 시스템, 데이터베이스 클러스터, NoSQL, BASE, CAP 이론

==========

1) 데이터 적재

(1) 데이터 적재 아키텍처 수립

- 아키텍처 정의: 요구사항을 구현하기 위한 기반 기술을 정의
    - 요구사항을 반영하여 하드웨어, 소프트웨어 아키텍처 정의 -> 정보시스템을 위한 기술적 기반이 됨
- 빅데이터 적재 아키텍처 요구사항 정의: 장비 / 소프트웨어 / 성능 / 인터페이스
- 장비 요구사항 정의: 서버 / 네트워크 / 스토리지 장비 규격 정의

- 소프트웨어 도입 요구사항 정의
    - 자체 구축(온프레미스): 상용, 오픈소스 소프트웨어 모두 고려
    - 자체 구축이 아닌 경우: 상용 클라우드 서비스 고려 (IaaS, PaaS, SaaS 중 선택)
    - 온프레미스(On-premise): 서버나 소프트웨어와 같은 기업의 솔루션 등을 원격 환경이 아닌 자체적으로 보유한 전산실에서 직접 설치하여 운영하는 방식
- 성능 요구사항 정의: 서버(용량) / 네트워크(트래픽, 대역폭) / DBMS(용량계획) / 응용 시스템(응답속도)
- 인터페이스 요구사항 정의: 내부/ 외부 연계 대상 시스템을 고려하여 정의
- 빅데이터 적재 하드웨어 아키텍처 정의: 서버 노드 / 데이터 / 네트워크 아키텍처
    - 서버 노드 아키텍처 정의: 관리를 위한 네임노드 / 데이터 처리를 위한 데이터노드
        - 단일 네임노드 + 다수 데이터노드 (+ 보조 네임노드)
        - 네임노드: 파일 시스템의 메타데이터를 관리 / 데이터를 블록 단위로 데이터노드에 분배
        - 데이터노드: 실제 데이터 저장, 처리가 수행되는 노드
    - 데이터 아키텍처 정의: RDB / NoSQL / 분산파일 시스템 등
        - 처리할 데이터 유형, 성격에 따라 아키텍처 구성
        - 정형 데이터의 경우: 관계형 데이터베이스(RDB)
        - 비정형 데이터의 경우: NoSQL
    - 네트워크 아키텍처: 목표 시스템 네트워크 구성/ 개별 장비 네트워크 환경

- 빅데이터 적재 소프트웨어 아키텍처 정의
    - 기반 소프트웨어 정의: 하둡 / 인 메모리 데이터베이스 / 데이터 분석 플랫폼 / 데이터 시각화 적용 검토
    - 빅데이터 적재 소프트웨어 아키텍처: 데이터수집 -> 적재&저장 -> 분석 -> 활용단계에 따른 아키텍처 정의

수집	적재 및 저장	분석	활용
- ETL - 크롤러 - 연계/수집 플랫폼	데이터 구성 플랫폼 - RDB 저장소 - NoSQL 저장소 - Object 저장소	빅데이터 분석 모델/플랫폼	- 데이터 시각화 - 데이터 활용 플랫폼 - Open-API 서비스

(2) 데이터 적재

- 데이터 적재 특징
    - 수집한 데이터는 빅데이터 시스템에 적재
    - 빅데이터 유형, 실시간 처리 여부에 따라 -> RDBMS / HDFS / NoSQL 저장 시스템
    - 분산된 여러 서버에서 데이터를 수집하는 데이터 수집 플랫폼, 저장 방법의 중요성이 점점 확대되고 있음
- 데이터 적재 도구
    - 데이터베이스가 제공하는 적재 도구로 직접 적재 / 데이터 수집 도구 이용하여 적재
      -> 데이터 수집 도구: 플루언티드 / 플럼 / 스크라이브 / 로그스태시
    - 플루언티드(Fluentd)

        - 크로스 플랫폼 오픈소스 데이터 수집 소프트웨어
        - 각 서버에 플루언티드 설치 -> 서버에서 로그 수집 -> 중앙 로그 저장소에 전송
        - 플루언티드가 로그 수집 에이전트 역할만 수행하는 가장 간단한 구조
        - 중간에 두는 플루언티드: 로그 저장소에 넣기 전에 로그 트래픽을 조정하기 위함
        - 여러 저장소에 로그를 복제해서 저장 / 로그 종류에 따라 다른 저장소로 라우팅 가능
    - 플럼(Flume): 대용량 로그 데이터를 수집, 집계, 이동 / 이벤트, 에이전트 활용
    - 스크라이브(Scribe): 다수 서버로부터 실시간 스트리밍 로그 데이터 수집
    - 로그스태시(Logstash): 모든 로그 정보를 수집하여 하나의 저장소에 출력해주는 시스템

==========

2) 데이터 저장

(1) 빅데이터 저장기술

- 빅데이터 저장 시스템: 대용량 데이터 집합을 저장, 관리하는 시스템
    - 대용량 저장공간 / 빠른 처리성능 / 확장성 / 신뢰성 / 가용성 등을 보장해야 함
- 비대칭성(Asymmetric) 클러스터 파일 시스템: 메타데이터를 별도의 전용서버로 관리함 (접근 경로가 분리되어 있음)

(2) 빅데이터 저장기술 분류

- 분산 파일 시스템 / 데이터베이스 클러스터 / NoSQL / 병렬 DBMS / 네트워크 구성 저장 시스템 / 클라우드 파일 저장 시스템

(3) 빅데이터 저장기술 - 분산 파일 시스템

- 컴퓨터 네트워크를 통해, 공유하는 여러 호스트 컴퓨터의 파일에 접근할 수 있게 하는 시스템

- 구글 파일 시스템(GFS)
    - 구글 대규모 클러스터 서비스 플랫폼의 기반이 되는 파일 시스템
    - 고정된 크기 64MB의 청크들로 파일을 나눔 -> 각 청크와 여러 복제본을 청크 서버에 분산 저장
    - 구성요소: 클라이언트 / 마스터 / 청크 서버
    - 구조: 클라이언트가 마스터에게 파일 요청 -> 마스터는 청크 서버에 요청 -> 청크 서버는 클라이언트에게 청크 데이터 전송

- 하둡 분산 파일 시스템(HDFS)
    - 대용량 파일을 분산된 서버에 저장 -> 데이터를 빠르게 처리할 수 있게 하는 시스템
    - 저사양 서버를 다수 이용하여 스토리지 구성 -> 비용 관점에서 효율적
    - 블록 구조의 파일 시스템: 파일을 특정 크기의 블록(하둡 2.0에서 128MB)으로 나누어 분산 서버에 저장함
    - 구성: 하나의 네임노드 + 하나 이상의 보조 네임노드 + 다수의 데이터노드
    - 구성요소
        - 네임노드: 마스터 역할 / 모든 메타데이터 관리 / 데이터노드들로부터 하트비트를 받아 상태 체크
        - 보조 네임노드: 상태 모니터링을 보조함
        - 데이터노드: 슬레이브 역할 / 데이터 입출력 요청 / 데이터 유실방지를 위해 블록을 3중 복제

        - 노드(Node): 컴퓨터 과학에 쓰이는 기초적인 단위, 대형 네트워크에선 장치나 데이터 지점(포인트)를 의미. 예를 들면, 개인용 컴퓨터, 휴대전화, 프린터, 서버 같은 장치를 말함

- 러스터(Lustre)
    - 객체 기반의 클러스터 파일 시스템
    - 구성요소: 고속 네트워크로 연결된 클라이언트 파일 시스템 / 메타데이터 서버 / 객체 저장 서버
    - 계층화된 모듈 구조로 TCP/IP, 인피니밴드와 같은 네트워크를 지원함

(4) 빅데이터 저장기술 - 데이터베이스 클러스터

- 하나의 데이터베이스를 여러 개의 서버 상에 분산하여 구축
- 성능, 가용성 향상을 위해 데이터베이스 파티셔닝 또는 클러스터링을 이용함
    - 데이터베이스 파티셔닝: 데이터베이스를 여러 부분으로 분할하는 것
- 데이터베이스 클러스터 구분
    - 리소스 공유 관점에서 공유 디스크 / 무공유 디스크로 구분
    - 공유 디스크 클러스터: 모든 데이터에 접근 가능 / 모든 노드가 데이터 수정 가능 / 높은 수준의 고가용성 제공
    - 무공유 디스크 클러스터: 데이터 파일을 로컬 디스크에 저장 / 파일을 노드 간 공유하지 않음
- 데이터베이스 클러스터 종류
    - Oracle RAC / IBM DB2 ICE / SQL Server / MySQL
    - Oracle RAC: 공유 클러스터 / 고가용성 / 쉬운 확장
    - IBM DB2 ICE: 무공유 클러스터 / 한 노드에 장애 발생 시 복구할 수 있도록 공유 디스크 방식 사용
    - SQL Server: 연합 데이터베이스 형태 / 여러 노드로 확장할 수 있는 기능 제공
    - MySQL: 비공유형 / 메모리 기반 데이터베이스의 클러스터링을 지원함

(5) 빅데이터 저장기술 - NoSQL

- NoSQL
    - 전통적인 RDBMS와 다른 DBMS를 지칭하기 위한 용어
    - 데이터를 저장하는 데에 고정된 테이블 스키마가 필요없음
    - 조인 연산을 사용할 수 없으며 수평적으로 확장이 가능
- NoSQL 일반적 특성
    - 관계형 모델을 사용하지 않음
    - 대규모 데이터를 처리하기 위한 기술
    - 확장성 / 가용성 / 높은 성능 제공
    - Schema-less: 자유롭게 필드 추가 가능
- NoSQL의 특성("BASE")
    - Basically Available: 언제든지 접근 가능(항상 가용성)
    - Soft-state: 노드의 상태는 "외부"에서 전송된 정보를 통해 결정되는 속성
      -> 특정 시점에는 데이터 일관성이 보장되지 않음
    - Eventually consistently: 일정 시간이 지나면 데이터 일관정이 유지됨
- NoSQL의 유형
    - 저장되는 데이터 구조에 따라 Key-Value / Column Family Data / Document / Graph Store
    - Key-Value Store: 유니크한 Key에 하나의 Value를 가진 형태
    - Column Family Data Store: Key 안에 (Column, Value) 조합의 여러 필드를 가진 형태
    - Document Store: Value의 데이터 타입이 Document 타입
    - Graph Store: 시맨틱 웹과 온톨로지 분야에서 활용되는 그래프로 데이터를 표현

- CAP 이론: Consistency / Availability / Partition Tolerance
    - 분산 컴퓨팅 환경은 유효성 / 일관성 / 분산 가능성 3가지 특성을 가지는데, 이 중 2가지만 만족할 수 있음
    - NoSQL은 CAP이론을 기반으로 함
    - 일관성: 모든 사용자에게 같은 시간에는 같은 데이터를 보여줘야 함
    - 유효성: 모든 클라이언트가 읽기/쓰기가 가능해야 함
    - 분산 가능: 물리적 네트워크 분산환경에서 시스템이 원활하게 동작해야 한함
- NoSQL 제품 종류: 구글 BigTable/ 아파치 HBase/ 아마존 SimpleDB/ 마이크로소프트 SSDS 등

(6) 빅데이터 저장 고려사항

- 요구사항 분석 절차
    - 요구사항 수집(도출) -> 분석 -> 명세 -> 검증
    - 사용자 요구사항을 분석하여 빅데이터 저장을 위한 제품 검토
- 기존 시스템 기술 검토 절차 수립
    - 데이터가 대부분 테이블로 정의될 수 있는 형태 & 기존에 RDBMS 도입된 형태 -> 기존 시스템 그대로 활용
    - 기존에 HDFS만 활용함 & SQL-like 분석환경 구축하고자 함 -> HBase 추가 도입 권장
- 데이터 저장의 안정성, 신뢰성 확보 방안 수립: 용량 산정 / 데이터 파악 / 시스템 구축 방안
- 유형별 데이터 저장방식 수립

데이터 유형	요구 데이터 종류	저장 시스템
정형	RDB / 스프레드시트	RDB
반정형	HTML/XML/JSON/웹 문서/웹 로그/센서 데이터	RDB, NoSQL
비정형	소셜 데이터/문서/이미지/오디오/비디오/IoT	NoSQL, HDFS

- 저장방식 결정
    - 저장방식 선정 시 고려요소: 저장기술의 기능성 / 분석방식 및 환경 / 분석 대상 데이터 유형 / 기존 시스템과의 연계