데이터마이닝응용_중간고사

01. 데이터마이닝 개론

① 데이터마이닝 생긴 이유(Motivation)

- 필요하니 생겼다

*Data mining 정의 = **KDD(Knowledge Discovery form Data base)**

- 데이터 폭발적 증가 : database의 수용 용량 초과
  → 처리의 필요성 대두
- 데이터 多 but 정보는 부족 = 찾기 위한 기술력 필요

⇨ 해결책 : data warehousing and data mining

- data warehousing and OLAP (on-line analytical processing)
- 데이터베이스의 데이터로부터 관심 있는 정보(규칙, 정규, 패턴, 제약) 추출

② 데이터베이스 기술의 발전

- 이전 : file 단순 쌓아둠, 분류 X = 찾기 어려움

-1960s : 데이터 수집, 데이터베이스 생성, IMS and network DBMS

- 1970s : 상관관계형 데이터 모델, 상관관계가 있는 DBMS 실행(HDB)
- 1980 : RDBMS, 발전된 데이터 모델(확장된 관계, OO(객체지향), OR(객체관계형) 등), DBMS(데이터베이스 관리 시스템) 중점을 둔 활용(RDB)
- 1990s-2000s : 데이어마이닝과 데이터 웨어하우스, 멀티미디어(이미지,오디오,비디오) 데이터베이스, 웹 데이터베이스

- data base

- 1) 계층형 (IMS,CICS) : 과거에 주로 사용, 빠름
- 2) 망형 : X , 상용화되지 못함
- 3) 관계형 (Oracle, DB/2, MY SQL) : 현재 사용, 쓰기 쉬움, 느림(기술력으로 극복)

*IBM이 IMS, System-R만들고 관계형 사용x, 계층만 Oracle이 관계형 만들자 DB/2만듦

③ 데이터마이닝이란?

- 관심있는(가치 있는, 숨겨진, 이전에 알려지지 않은, 사용 유용성이 있는) 정보, 패턴을 데이터베이스의 데이터로부터 추출

데이터마이닝 = 데이터 고고학, Business intelligence

- 데이터마이닝이 아닌 것

- Query processing : 데이터베이스에서 단순 처리
- 전문가 시스템, 머신러닝, 통계 프로그램

④ KDD Process(데이터마이닝 프로세스)

1) 데이터베이스에서 출발 : 거래(Transation) 기록
   ⇨ 데이터베이스 축적
2) 데이터 웨어하우스 구축(Data Cleaning) : 노이즈와 미싱들을 제거하고, 형식을 통합하여 큐브(다차원)로 구축
3) Task-relevant data 선택 : 대용량 데이터 웨어하우스에서 필요한 것만 뽑음(Selection)
4) 데이터마이닝 ⇨ 패턴 뽑아내기 ⇨ 의사결정에 사용

- KDD 프로세스 단계

1) 활용 기본 정보에 대해 공부하기 (알아야 분석 가능)
2) 선행처리 & Data cleaning
3) 데이터 축소(Reduction) or 변화(Transformation)

- 축소) 차원(dimension) 축소 / 샘플링(구간별 통합)
- 변화) 단위 통합

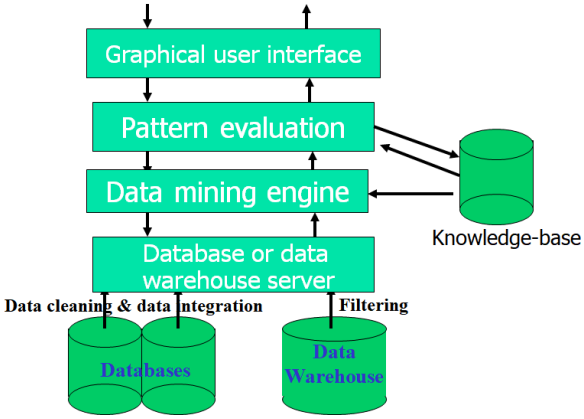
4) 데이터마이닝의 기능 선택

- Summarization, Classification, Regression, Association, Clustering

5) 데이터마이닝의 알고리즘 선택
6) 데이터마이닝 : 흥미있는 패턴 찾기
7) 패턴을 평가하고 지식을 발표(presentation)

- 의사결정에 사용되기 때문에
- 시각화(Visulization), 변환(transformation), 쓸모없는 패턴 제거(removing redundant patterns), 등

- 데이터마이닝 구조



⑤ 데이터마이닝에서 다루는 데이터
   ⇨ 다양한 데이터 소스에 적용가능

- 관계형 데이터베이스
- 데이터 웨어하우스(Data Warehouses)
- Transactional Database RDB : 매일매일 기록
- 발전된 DB와 정보저장소(information Repositories)

- OOD, ORD
- 시계열 데이터
- 문서 데이터
- 여러 성격이 혼재된 기존 데이터
- 웹 문서 데이터

⑥ 데이터마이닝의 기능

- Characterization(Generalization) : 숫자들로부터 간단한 컨셉 도출(연간 강수량 ⇨ 강수량 별 의미부여)
- Discrimination(Specialization)
- **Association(연관관계, correlation and causality)**

- 상관관계(correlation) : 기저귀를 사면 맥주를 산다
- 인과관계(causality) : 병원에서 진단 후 약을 산다
- A→B [sup = (rule 중요도) , conf =(rule의 신뢰도)]

*Support = A^B / 전체

*Confidence = A^B / A IF 100%, 떨어뜨려 놓자

- **Classification and Prediction**

- 분류(Classification) - 머신러닝

어떤 성향을 가진 사람이 구매하는지 샘플을 통해 분류 model을 만들어서 살지 안 살지 분류해서 DM발송

⇨범주형 데이터(Categorical data) 종교, 성별, ⇨ label

- 예측(Prediction)

과거의 자료를 기반으로 model을 만들어서 발생될 상황 즉, 알지 못하는 숫자나 빠진 숫자를 예측

⇨숫자형 데이터(numeric data) ⇨ 날씨 예측

- **Cluster analysis**

- 군집화 : 유사성향 데이터를 그룹별로 묶음
- classification은 이미 알려진 label에 따라서 분류 supervised learning(지도학습)
- cluster는 유사성향끼리 만들고 label을 붙여줌 unsupervised learning(비지도학습)
- inter class similarity : 클래스 간의 유사도 ⇩
- intra class similarity : 한 클래스 내의 유사도 ⇧

- **Outlier analysis**

보통의 패턴에서 동 떨어진 것 = 노이즈 취급하다가 이제는 중요한 information을 포함함 = 분석방법 중요

ex) 신용카드의 비정상적인 사용

- **Trend and evolution analysis**

= regression analysis(회귀분석), periodicity analysis(주기성 분석), similarity-based analysis(유사성분석)

- 다른 패턴 양상 혹은 통계적 분석

✓ 데이터마이닝을 통해 얻는 정보는 다 관심 있는 것?

놉, 모든 패턴에 관심X = 중요한 패턴 추출 필요

- Interestingness measures

1) Objective(객관적인) : support 낮음 = 흥미 X

2) Subjective(주관적인) : 개인의 신념, 기대가 기준

- 의미 있는 패턴의 조건

쉽게 이해됨, 새로운/시험 데이터 셋에서도 돌아감, 의미있음, 새로움, 검정 가능

✓ 패턴을 완벽히 다 찾을 수 있는가?

- 완전성 : 모든 패턴 다 찾아내기(Completeness)
- 최적화 : 관심 있는 패턴만 찾아내기(Optimization)

1) 모든 패턴 찾고 관심 없는 패턴 제거

2) 처음부터 관심 있는 패턴만 찾기

⇨ 데이터마이닝은 여러 학문의 합류점(통계, 기계, etc)

⑦ 데이터마이닝의 분류

- 기능상 분류

- Descriptive data mining : 숨겨진 형상 찾기
- Predictive data mining : 미래의 데이터 예측

- 관점별 분류

- 데이터베이스 / 관계형, 객체지향형
- 지식 / 관계, 분류, 군집화
- 기술 / OLAP, 머신러닝, 통계, 신경망
- 응용분야 / 물류, 은행, DNA mining

⑧ 데이터마이닝의 이슈(참고)

- 데이터마이닝의 방법론, 사용자의 개입의 유무
- 알고리즘의 응답시간 & 규모
- 데이터마이닝의 다양성, 응용분야, 사회적 영향

02. 데이터 선행처리(Data Preprocessing)

✓ 왜 데이터 선행처리를 진행하는 가?

- 실제 세계 속 데이터는 완전하지 않고, 흠이 있다

불완전(Incomplete) : 데이터의 누락

노이즈(noisy) : 아웃라이어나 오류 포함

일관성 없음(inconsistent) : 단위가 통합X

⇨ data cleaning이 필요

- 질 좋은 데이터가 아니면 질 좋은 결과 못 만듦

Quality decisions must be based on quality data

① 데이터 선행처리의 주요 역할

- data cleaning
  누락된 데이터 찾고, 값 보정, 일관성 찾기
- data integration : 데이터 통합
- data transformation : 정규화
- Data reduction : 데이터 축소(차원 줄이기, 샘플링)
- Data discretization : 이산화 시키기

② Data cleaning

- 결측치(missing data) 채우기
- outlier 찾고 노이즈 지우기
- 일관성 없는 데이터 수정하기

*coloum = field, attribute, variable, dimension

*row = record, tuple, instance

- Missing data

- 발생원인) 측정 장비의 문제, data 기록셋에 맞지 않아서 삭제, 이해하지 못해서 삭제, 입력당시 중요하지 않았음, data history와 변화 누락

⇨ missing data handling 중요 : 추론을 통해서 채움

- Missing data handling

- missing data 관련 tuple 무시 (자주사용)
  data 건수 中 missing data가 많으면 문제 발생
- 수작업으로 채워 넣기
- 평균값으로 채워 넣기
- "unknown"으로 채워 넣기 : 나머지 값은 사용 가능
- 유사집단끼리 모아서 집단 별 평균으로 채워 넣기
- 추론을 통해 값 채우기

- Noisy Data(random error)

- 기록의 중복, 측정상의 문제, 기술적 한계

- Noisy Data handling

- Binning method

data 구간으로 나눠서 noisy data에 구간 평균값 넣기

- Clustering(군집화)

아웃라이어 찾고 제거하기

- 컴퓨터와 사람이 협업

컴퓨터가 이상한 결과값 찾고 사람이 판단

- Regression(회귀 분석)

수치데이터 ⇨ 회귀식 찾고 smoothing 하기

= error 제곱의 합 최소화하기

③ Simple Discretization Methods : Binning

- Equal-width(구간의 길이를 동일하게 나누기

W[A,B] = (B-A)/N ⇨ 가장단순한 방법

- 아웃라이어, 치우져진 데이터가 있으면 사용 어려움

- Equal-depth(각 구간의 데이터 수 동일하게 하기)

- 범주형 데이터 다루기 어려움(숫자형만 가능)

④ Cluster Analysis

유사한 데이터 별로 군집화 후 어느 곳에도 속하지 않은 데이터를 noisy로 판단 ⇨ 삭제

⑤ Regression

산점도를 그리고 회귀 방정식을 활용하여 각 데이터와 방정식 간의 거리를 이용하여 noisy 판단

⑥ Data integration

- data integration

여러 데이터 소스로부터 데이터 웨어하우스 구축

- schema integration

동일한 자료를 다른 방식으로 표현했을 때, 하나로 통합

- 데이터 value의 모순점

같은 측정치를 다른 단위로 표현한 경우 하나로 통합

⑦ Data Transformation

- Smoothing : 노이즈 제거
- Aggregation : 데이터 집계 = 큐브 구축
- Generalization(⇔specialization)

데이터 큐브를 만들 때, 계층 구조(Concept Hierarchy)된 자료를 상위 개념을 통합(도로명, 읍, 면, 동 ⇨ 시로 통합)

- Normalization(정규화)

- min-max 정규화
- z-core 정규화
- decimal scaling

- Attribute, feature constriction

분석에 따라서, 새로운 attribute 만들기(주민번호로 나이 attribute 만들기)

⑧ Data Transformation : Normalization

✓ 왜 정규화? 데이터마다 scale이 다르기 때문

- min-max 정규화

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

0~1사이로 정규화 ⇨ 극단치의 영향 多

- z-score 정규화

$$v' = \frac{v - mean_A}{stand\_dev_A}$$

극단치가 있더라도 커버 가능(평균 빼고, 표준편차로 나눔)

- decimal Scaling

$$v' = \frac{v}{10^j} \quad \text{Where } j \text{ is the smallest integer such that Max}(|v'|)<1$$

⑨ Data Reduction Strategies

✓ 데이터 크기가 너무 커지면 처리하기 어려움

- 축소된 데이터 결과나 원본 결과나 동일해야한다

• 데이터 큐브 만들기

데이터 큐브의 디테일을 적합한 수준(level)에 통합

• 차원 축소

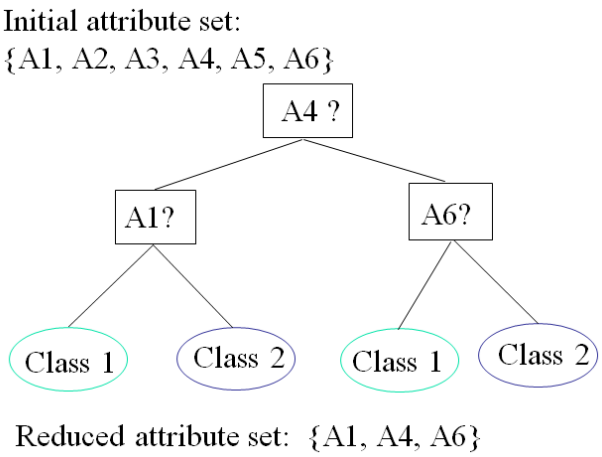
- feature(coloum) selection

10개의 feature 중에서 5개 선택 = 어떻게 뽑을 것이냐

⇨ 원본 분포와 비슷하도록 feature을 선택해야 함

- Heuristic methods(데이터는 기하급수적 늘어남)

1. 중요한 순서대로 선택(10 중 중요한 순서로)
2. 안 중요한 자료부터 제거
3. 뽑고 기존의 자료와 비교하여 기존 자료 중 불필요는 제거(1번 2번 합쳐서 사용)
4. decision-tree induction



Discrimination power : 데이터별로 구별력(높아야 함)

⇨ 엔트로피를 사용해서 높임

• 샘플링(데이터 크기 자체를 축소)

• 이산화(히스토그램 만들기)

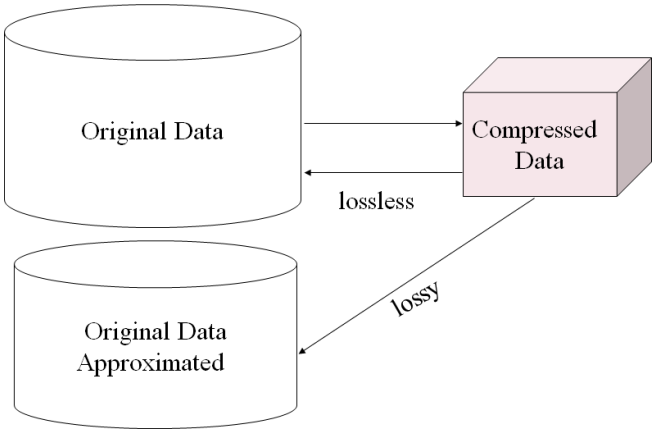
⑩ Data Compression : 데이터 압축

• String compression

lossless compression(무손실 압축) ⇨ 문자압축

• Audio/video compression

lossy compression(손실 압축)



⑪ Numerosity Reduction(데이타 량을 줄이기)

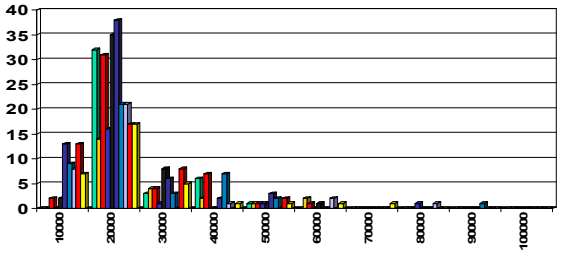
• Parametric methods(모수적 방법)

데이터 특성을 잘 반영한 수학적 모델 찾기, 찾기 어려움

• Non-parametric methods(비모수 방법)

데이터 분포 자체를 이용, 데이터를 여러 방법으로 표현

- 히스토그램



개별 데이터로 의사결정에 사용하기는 어렵지만,

이렇게 히스토그램(분포)은 사용가능(숫자형, 범주형 가능)

⑫ Clustering(군집화)

데이터를 파티션별로 분류하고, 파티션의 특징만 기록

• 전체적으로 퍼져있는 경우는 사용하기 어려움

• 계층적 군집(hierarchical clustering), 단위별 가능

구조화 시켜서 저장(보다 자세히, 조직화 저장 가능)

*index 구조 = tree 구조

⑬ 샘플링(Sampling)

전체의 데이터에서 일부를 뽑아서 진행

⇨ 원본 분석 결과와 샘플 분석결과가 유사/동일해야 함

✓ 샘플링은 data 수를 줄이나요?

놉, 흩어진 data를 다 읽고 샘플링 진행함

⇨ sampling may not reduce database I/Os

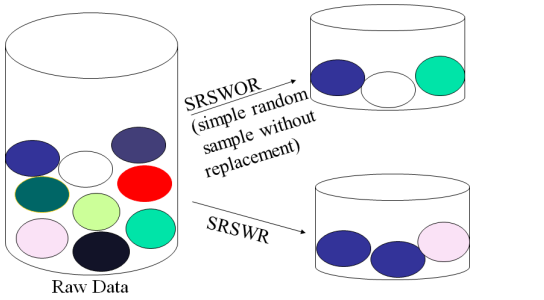
• random sampling 사용 : 샘플 수 많아야함

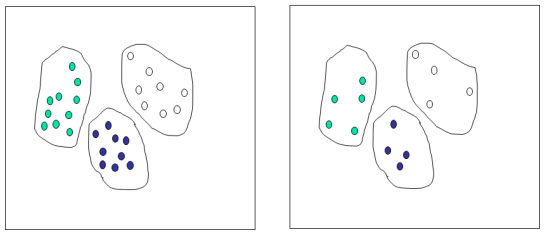
• Adaptive sampling 방법

전제 data 분포에 따라 적응시켜서 사용

- Stratified sampling(계층화 샘플링)

data를 계층 별로 나누고 계층별로 일정 비율 뽑기





⇨ 비율대로 뽑기 = 최대한 원본 유지

03. 데이터 웨어하우스와 OLAP

① 데이터 웨어하우스

- 근간 : 데이터큐브 기술
- trend를 읽고 의사결정을 위해 사용되는 기술

⇨ 운영계 DB(Operation DB)와는 별도로 운영

- data warehouse : TV의 지역별 판매 추이
- 운영계 DB : 홍길동이 지난 달에 구매한 전자제품(즉각)

- 데이터웨어하우스의 특징

- 주제별 분류
- 통합된 데이터
- 시간에 따른 변화(분기별 업데이트)
- 비휘발성

- **OLTP(거래 기록용 ⇨ 업무처리)**

Online transactional processing

- transaction 처리를 위해 day to day operation
- 고객 중심, 현재, 상세한 정보
- ER 모델링, 현재, 지역적, 업데이트

- **OLAP(분석용 ⇨ 의사결정 지원)**

Online analytical processing

- 시장 전체 통합 내용, 과거 data, 통합된 정보
- Star 모델링, 어떻게 변화, read-only(주기적으로만 업데이트) & 복잡한 쿼리

- **헤테로지니어스 DBMS(Hetreogeneous)**

원하는 정보가 분산(성격이 달라짐)되어 있을 때 원활히 뽑아내기 위해 만들어짐

⇨ wrapper(DB 통합), mediator(조정자)

- 어떤 정보가 어디에 있다는 index 기록, 따로 데이터 없음

⇨ Meta dictionary에서 찾아서 분산된 위치로 query를 변환해서 보내고 이에 대한 답 통합해서 을 산출

- 따로 저장 필요 X
- but processing 오래 걸리고, 운영계 DB transaction 발생 시, 자신 + meta dictionary까지 변화시켜야 함

⇨ 이 단점을 극복하려 **데이터 웨어하우스 등장**

- 운영계 영향 없음, 신속한 반응&결과산출
- but 저장 공간 필요, 운영계 변화에 즉각 반응X

✓ 왜 데이터 웨어하우스를 분리했을까?

- 높은 퍼포먼스 유지 위해

DBMS(OLTP) vs Warehouse(OLAP)

- 다른 기능과 다른 데이터

Warehouse : 결측치, 통합된 데이터, 데이터의 질 중요

DBMS : 별로 중요하지 않음, 현제 변화 처리에 집중

② 데이터큐브(Data Cube) ★★★

1) Transaction 발생

- 홍길동이 분당에서 LG TV 50만원에 2대 구입(9.20)
- 일지매가 모현에서 삼성 냉장고 75만원에 3대 구입(9.15)

2) DB 만들기(요소별 분리 ERD_개체 관계 모델)

- customer table : 새로운 고객이 생길 때 기록
- location table : 위치 기록
- time table : 시간 기록
- item table : 신상품이 생길 때 기록

= 기준 테이블 작성(= dimension table)

⇨ transaction table(fact table) : transaction이 생길 때 마다 update해서 작성

- sales table : dimension attribute(고객 ID, time, item), measure attribute(가격, 수량) 기록

= 기존 테이블을 수정하여 기록 / 고유 내용도 기록

- 각 기준 테이블에 있는 key와 measure attribute와 합쳐져서 만들어짐

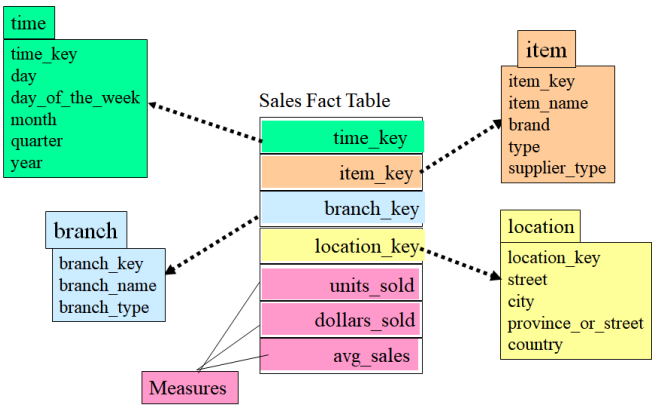
3) 만들어진 DB를 통해서 data warehouse 만듦

- 각 transaction에 해당하는 cell을 찾아서 가격, 수량 추가
- time X item X location 3차원 큐브 형성 : 총 3개의 2차원 큐브, 3개의 1차원 큐브, 0차원 큐브(모든 cell의 내용을 합친 것, apex cube) 1개

③ 데이터 웨어하우스의 스키마

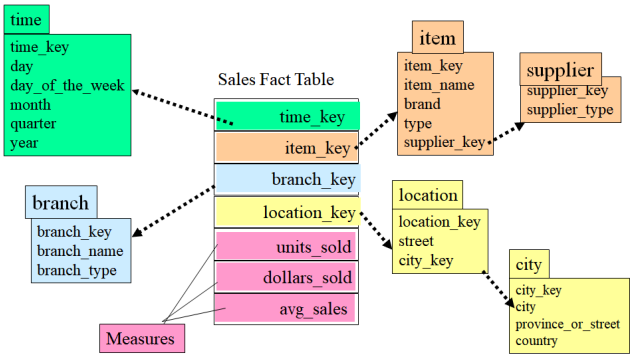
- star schema(별형)

fact table 1개 -dimension table 연결



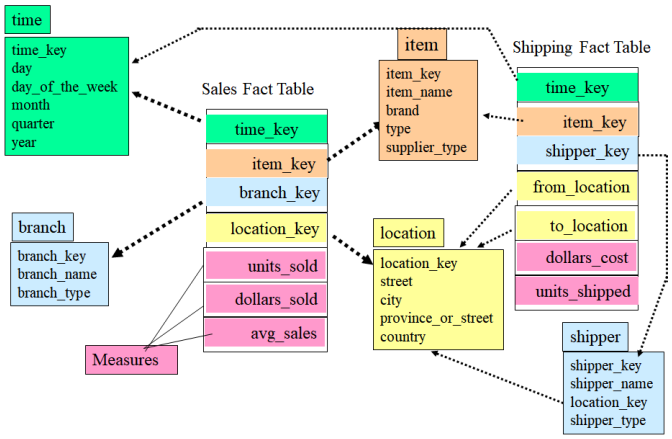
- snowflake schema(눈송이형) dimension 분화

fact table 1개 - dimension table 두 단계로 연결



- Fact constellations(성운형)

fact table 2개 이상 - dimension table 공유



④ 데이터마이닝 쿼리 언어 (DMQL)

- star schema in DMQL

```
define cube sales_star [time, item, branch, location]:
        dollars_sold = sum(sales_in_dollars), avg_sales =
          avg(sales_in_dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week,
    month, quarter, year)
define dimension item as (item_key, item_name, brand,
    type, supplier_type)
define dimension branch as (branch_key, branch_name,
    branch_type)
define dimension location as (location_key, street, city,
    province_or_state, country)
```

- snowflake schema in DMQL

```
define cube sales_snowflake [time, item, branch, location]:
        dollars_sold = sum(sales_in_dollars), avg_sales =
          avg(sales_in_dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week, month,
    quarter, year)
define dimension item as (item_key, item_name, brand, type,
    supplier(supplier_key, supplier_type))
define dimension branch as (branch_key, branch_name,
    branch_type)
define dimension location as (location_key, street,
    city(city_key, province_or_state, country))
```

- Fact constellation in DMQL

```
define cube sales [time, item, branch, location]:
        dollars_sold = sum(sales_in_dollars), avg_sales =
          avg(sales_in_dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week, month, quarter, year)
define dimension item as (item_key, item_name, brand, type, supplier_type)
define dimension branch as (branch_key, branch_name, branch_type)
define dimension location as (location_key, street, city, province_or_state,
    country)
define cube shipping [time, item, shipper, from_location, to_location]:
        dollar_cost = sum(cost_in_dollars), unit_shipped = count(*)
define dimension time as time in cube sales
define dimension item as item in cube sales
define dimension shipper as (shipper_key, shipper_name, location as location
    in cube sales, shipper_type)
define dimension from_location as location in cube sales
define dimension to_location as location in cube sales
```

⑤ Measures : 3가지 종류

- distribute

분할해서 값을 구하고, 통합해서 처리 가능한 attribute

ex) count(),sum(),min(),max()

- algebraic

attribute의 연산으로 값을 구할 수 있는 attribute

ex) avg(),min_N(),standard_deviation()

- holistic(⇔distribute)

분할이 불가능한 attribute, 전체를 계산해서 얻을 수 있는 attribute

ex) median(),mode(),rank()

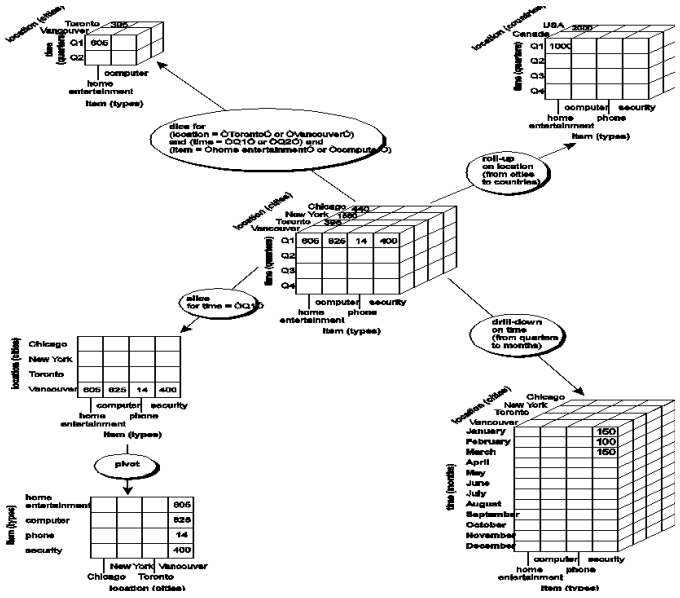
⑥ OLAP Operation

- Roll up(drill-up) : summarize data (계층 올라감)
- Drill down(roll down) : reverse of roll up
- Slice and dice : project and select

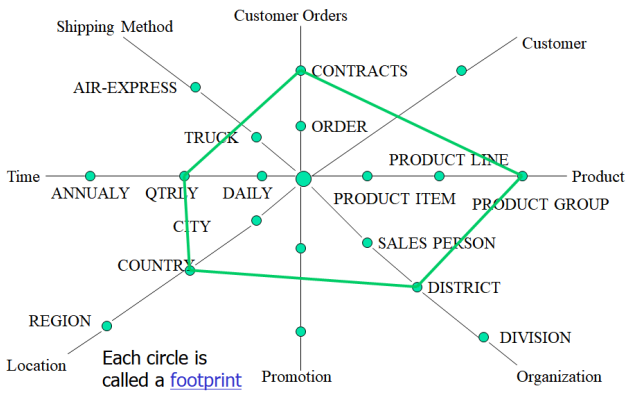
3차원 데이터 큐브에서 2차원으로 자르기(Slice), 구간별 measure attribute(dice)

- Pivot(rotate) : 축을 바꿔가면서 봄
- Drill across : fact table 간을 왔다,갔다하면서 봄
- Drill through : 더 세부적으로 cell 내용을 봄

계산된 pointer를 가지고 operation DB로 찾아갈 수 있게 함



- Star-Net Query Model(어느 레벨에서 도식화)



04. Association Rule Mining(연관규칙마이닝)

① Association Rule Mining

데이터베이스에 있는 아이템 사이에서 자주 발생하는 패턴, 연관 규칙 찾기, correlations찾기 or causal structures 등 관련 있는 아이템들을 찾는 것 ⇨ A 아이템과 관련 있는 B아이템 찾기

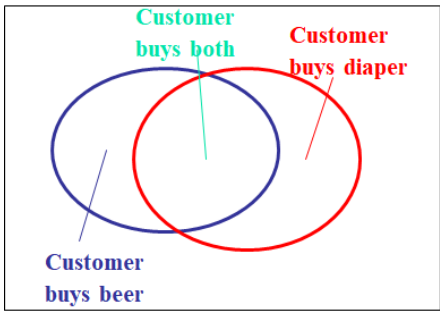
- 응용

장바구니 분석, 카탈로그 디자인, 어떤 상품이 우위에 있는지, 클러스터링, 분류 등

- 예시

Rule form : "Body → Head [support, confidence]"
buy(x, "diapers") → buys(x, "beer")[0.5%, 60%]
major(x,"CS")^takes(x,"DB") → grade(x,"A")[1%,75%]

- Support & Confidence



diaper ⇨ beer

- support(Rule의 중요도) : diaper ∩ beer / U
- confidence(신뢰도) : diaper ∩ beer / diaper

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

최소 support 50%, and 최소 confidence 50%인 경우,
- A ⇨ C [S = 2/4 = 50%, C = 2/3 = 66.7%]
- C ⇨ A [S = 2/4 = 50%, C = 2/2 = 100%]

*Support는 방향성 중요 X, Confidence는 방향성 중요

- 용어 정리

- Boolean association(text)
  true/false, Yes/No
- Quantitative associations(number)

- buys(x, "SQLServer") ^ buys(x, "DMBook") → buys(x, "DBMiner") [0.2%, 60%]
- age(x, "30..39") ^ income(x, "42..48K") → buys(x, "PC") [1%, 75%]

- Single dimension vs Multiple dimension
- Single level vs multiple-level analysis

ex) 기저귀를 산 사람이 맥주 산다 (Single)
기저귀를 산 사람이 [브랜드 별] 맥주 산다(Multiple)

- Correlation, casuality analysis
- Constraints : 제약식을 줄 수 있다

ex) small sales (sum 〈 100) trigger big buy(sum 〉 1,000) ?

② Mining Association Rules - Example

1) Transaction DB와 최소 sup, conf 주어져야 함

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

min sup 50%
min conf 50%

2) Frequent item set 찾기 (min support로 판단)★

Frequent Itemset	Support	
{A}	75%	3/4
{B}	50%	2/4
{C}	50%	2/4
{A,C}	50%	2/4

*{D} support = 20 (1/4) ⇨ 탈락

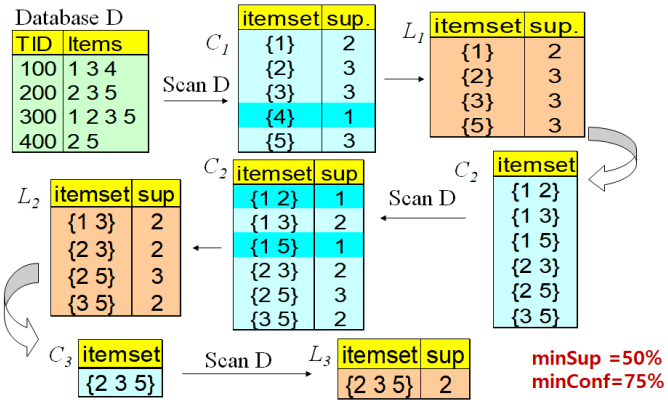
3) Rule 생성, 평가

- rule A ⇨ C

- support = support({A∩C}) = 50%
- confidence = support({A∩C}) / support({A})
  = {A∩C} / {A} = 2/3 = 66.6%
- Apriori principle(Apriori 원칙)★

frequent item set의 어떤 subset도 무조건 frequent
{A,C} frequent ⇨ {A}, {C} frequent (반대는 불가능)
[대우] {D} not frequent ⇨ {A,D} not frequent (참)
[역] : 참 or 거짓 가능 / [부정] : 거짓

③ The Apriori Algorithm - Example



* 탈락하면 대우 이용하여, 앞 단계 정리

- evaluate each Rules

1) frequent item set 만들기 { 2, 3, 5 }
2) Rule 만들기

- 2, 3 ➔ 5 conf = 100%
- 2, 5 ➔ 3 conf = 66.7%
- 3, 5 ➔ 2 conf = 100%
- 2 ➔ 3, 5 conf = 66.7%
- 3 ➔ 2, 5 conf = 66.7%
- 5 ➔ 2, 3 conf = 66.7%

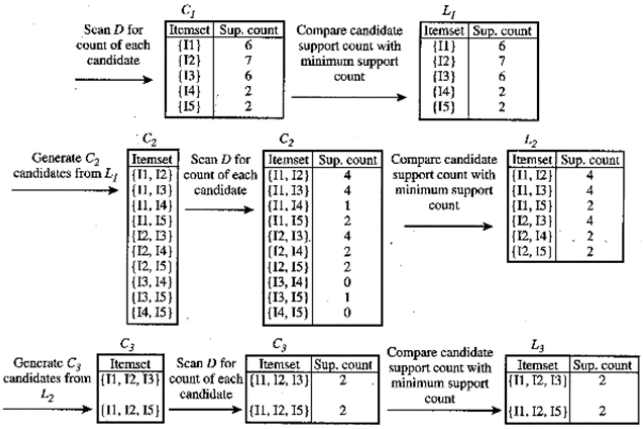
⇨ {2^3 ⇨ 5}, {3^5 ⇨ 2} 채택

④ The Apriori Algorithm – Example

다음 9개의 transaction으로 구성된 DB에서 minsup = 20%, minConf = 70% 일 때, frequent item set 및 association rule을 구하시오

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

- finding frequent item set



- finding rules

$I1 \wedge I2 \Rightarrow I5$, $confidence = 2/4 = 50\%$

$I1 \wedge I5 \Rightarrow I2$, $confidence = 2/2 = 100\%$

$I2 \wedge I5 \Rightarrow I1$, $confidence = 2/2 = 100\%$

$I1 \Rightarrow I2 \wedge I5$, $confidence = 2/6 = 33\%$

$I2 \Rightarrow I1 \wedge I5$, $confidence = 2/7 = 29\%$

$I5 \Rightarrow I1 \wedge I2$, $confidence = 2/2 = 100\%$

SPSS 실습자료

01. 데이터마이닝의 이해

① 데이터마이닝이란?

- 데이터베이스 환경의 변화 : 빅데이터의 등장
- 비즈니스 환경의 변화 : 군집화, 마케팅 타케팅 필요
- 의사결정 환경의 변화 : 실시간 의사결정의 필요

⇨ 데이터마이닝(KDD)의 출현

- 데이터마이닝의 정의

not a magic solution + advanced & repeated

⇨ 사전작업과 노력 ⇨ 상호작용과 반복 프로세스

② 데이터마이닝, OLAP 그리고 Query

- 데이터마이닝이란 대용량의 데이터베이스로부터 각종 통계기법 또는 인공지능기법을 이용하여, 데이터 내부에 숨어 있는 유용한 정보를 추출해 내는 일련의 과정

⇨ 숨겨진(hidden) 자료 찾기

- 분명한 미래의 분석 방향, 데이터마이닝을 이용한 분석 결과를 첨부하지 않은 보고서는 검토 가치X

  기본적인 자료 - 다차원 자료 - 숨겨진 자료

  (SQL, Query) (OLAP) (Data mining)

- SQL : Shallow knowledge, 찾으려는 데이터에 대해서 잘 알고 있을 때

ex) 2008년 4월의 매출

- OLAP : summeraize된 정보를 찾을 때

ex) 2008년 4월 지역별 매출 건수

- Data mining : 숨겨진 정보를 찾을 때

ex) 2008년 4월에 제품을 구매한 홍길동이 향후 6개월 이내에 추가 구매를 할 가능성

- OLAP와 Data mining

- CRM에서 데이터마이닝이 활용되는 단계

*CRM(Customer relationship management)

[고객 유치] - [고객 양육] - [고객 유지]

= OLAP ⇨ 고객의 실제 발생한 데이터 분석

  ex) 고객 구매성향 분석, 특성 분석, 요인 분석

= Data mining ⇨ 고객의 데이터 기반형 예측

  ex) 스코어기반의 고객세분화, 교차판매 대상 상품 예측

- Analysis trends

[기초데이터분석] 경영자의 직감 ⇨ 데이터 관리 도구에 의한 분석 ⇨ 데이터 웨어하우스 이용(OLAP) ⇨ 데이터 마이닝 ⇨ 텍스트 마이닝[고급 데이터 분석]

- 데이터마이닝과 통계기법, OLAP과의 차이점

- 통계기법 (기술통계, 분포특성, 가설검정)

각종 데이터의 통계적 분석은 데이터의 분포에 대한 가정을 바탕으로 기본 통계학적 속성을 파악하는데 초점을 맞추어 분석자가 확인하고 싶은 연구 가설을 검증하기 위하여 수행

- OLAP(다양한 관점에서 데이터 해석)

방대한 양의 데이터를 변수 기분의 다양한 관점 또는 차원을 통해 제시함으로써 데이터를 의미있는 형태로 해석할 수 있는 틀 제공

- 데이터마이닝(인공지능적인 요소 추가)

인공 지능적인 요소(의사결정나무, 연관 규칙, 인공신경망)를 가미하여, 특정 변수나 사건을 예측하게 하고 비즈니스 규칙을 세우기 위한 변수들 간의 관계를 파악하도록 하는 기술, 자료에 대한 이해와 가설 없이도 탐색적인 연구(가설 검증 또는 EDA)를 수행할 수 있게 함.

③ 데이터마이닝 프로세스 및 방법론

- 데이터마이닝의 주요 특징

1) 데이터 수집

데이터를 효율적이고 체계적으로 어떻게 수집 및 정의

2) 모형의 적합성

개발된 다양한 예측 모형들의 가치는 새로운 데이터에서 얼마나 잘 적용되는 가로 평가 ⇨ 강인성(robustness : 새로운 데이터에 적용해도 잘 맞음), 일반화

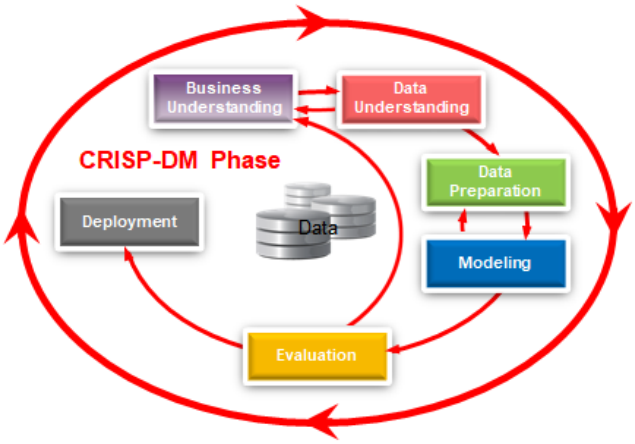
3) 이해 및 활용

실제 의사 결정에 활용하고, 기업의 전략과 경영목표에 맞추어 활용 ⇨ 충분한 이해와 적합한 활용 중요

4) 룰 관리

데이터 실시간 분석을 돕기 때문에 엄격한 룰의 설정 및 이행 등 관리

- 데이터마이닝 방법론



⇨ 피드백이 가능한 환류체계

1) 비즈니스 이해 : domain(기본 정보) 분석
2) 데이터 이해
3) 데이터 선행처리
4) 예측 모형 만들기 : 데이터마이닝 기법 사용
5) 평가
6) 실제로 구현하여 사용

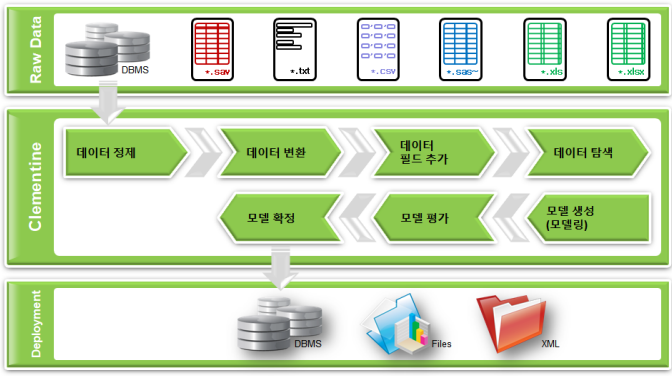
*CRISP-DM

(CRoss-Industry Standard Process for Data mining)

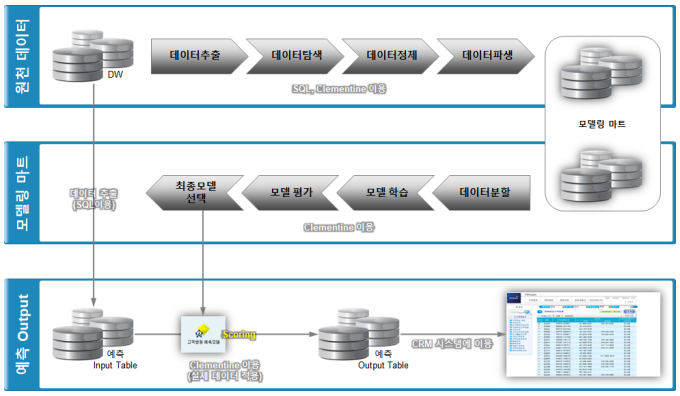
모든 domain에 적용되는 기준 = 금융, 물류 등 통용됨

- 클레맨타인(Clementine)의 활용

Clementine은 원천 데이터에 접근하여 각종 정제, 변환, 필드추가, 탐색, 모델링 및 평가를 통해 모델을 확정하고, 최종모델을 Clementine 이용하여 DB, Flat File, XML 코드로 전개



- Clementine의 예측모델 활용



④ 모델링 기법의 개요

- 데이터마이닝 기법과 알고리즘

Technique & Algorithm \ Problem Type & Technique		Supervised Modeling		Unsupervised Modeling	
		Predictive		Descriptive	
		Classification	Estimation	Clustering	Association/Affinity
Regression	Linear	✓	✓		
	Logistic	✓	✓		
Decision Tree	CART	✓	✓		
	C5.0	✓	✓		
	CHAID	✓	✓		
	QUEST	✓	✓		
Neural Network	MLP	✓	✓		
	RBF	✓	✓		
Clustering	K-Means			✓	
	Kohonen			✓	
	TwoStep			✓	
Association	Apriori				✓
	GRI				✓
	Sequence				✓

⇨ 지도학습의 경우 레이블과 같이 수집

- 데이터 분할(Data Partitioning)

- 학습데이터 : 모델만들기
- 검증데이터 : 생산된 모델 검증

⇨ 업무에 따라서 데이터 분할

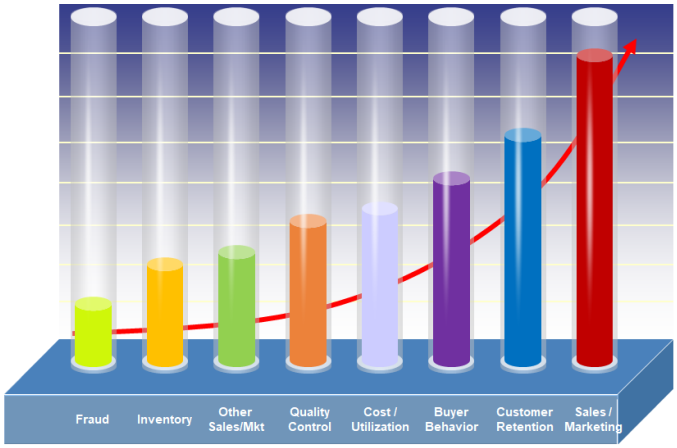
⑤ 데이터마이닝의 사례

- 다양한 데이터마이닝의 테마

궁극적으로 데이터마이닝을 하는 것은 일반적인 분석이 잘하지 못하는 분석을 할 수 있기 때문으로 분석적인 경쟁력을 확보하고, 더욱 더 과학적이고, 고도화된 분석 정보 획득을 위해 필요

- 예측 : 과거 데이터로 미래 예측
- 세분화 : 고객 특성집단별 마케팅
- 연관성/순차규칙 : 장바구니 분석
- 인과모형 : 데이터 상의 목표 변수에 대한 인과관계
- text 분석 : SNS 마케팅
- 기타 : 신용평가, 품질개선, 이미지 분석

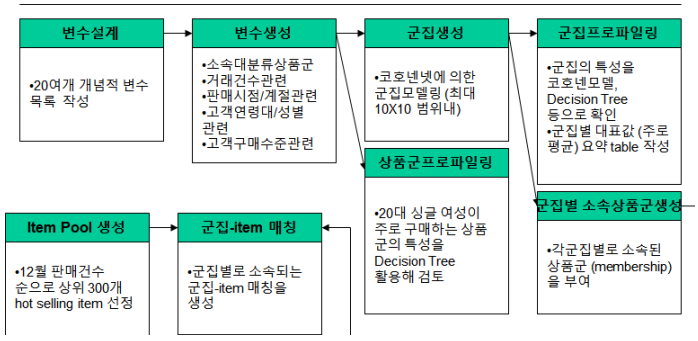
- 데이터마이닝의 활용 분야



- 활용사례 1) A사의 추천시스템

상품과 상품을 상품군 단위에서 군집화한 후 동일 군집에 속하는 어떤 개별 item을 선택하거나 구매한 고객에게 해당 item이 속한 상품군 군집내에 속한 다른 개별 item 추천

- 상품군 선택은 군집화, 개별 item 선정은 최근판매순위 활용



- 군집내에서 sorting 후, 상품관의 연관성 발견
- 상품추천에 연관성 규칙 적용

- 활용사례 2) B사의 URC-Selling 모델

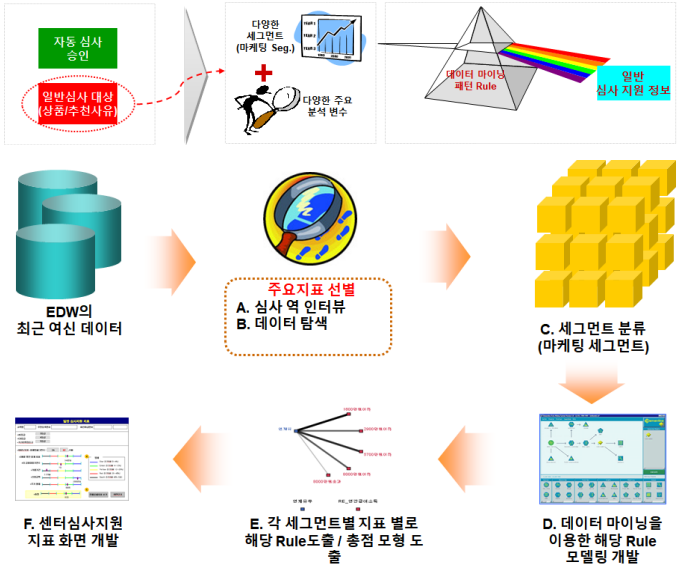
지도학습모델을 사용하여 Up-selling mining진행하여 적당한 고가의 제품 선정 및 홍보, Re-selling mining은 주기모델을 활용하여 상품/고객별 평균 주기를 계산하여 구매 예상 직전 시기에 홍보, Cross selling mining은 연관성 모델을 활용하여 특정 상품을 산 고객에게 가장 연관성이 높은 상품 추천

- 활용사례 3) C사의 URC-Selling 모델

기존의 RFM에서 상품이라는 축을 추가하여, 고객 지향 중심의 분류/분석관점에서 상품의 정보를 고객과 동일하게 하는 분석과점으로 이동, 상품의 재분류와 상품분류 속에 또 다른 분류가 있는 형태로, 수평적 분류와 수직적 분류로 세분화 범주화 시켜서 상품관련 분석을 강화 시킴

- 활용사례 4) K사의 대출연체고객판단모델

예상되는 심사 폭주와 다양한 심사기준에 대한 정보 제공을 위해서 센터심사지원지표개발의 필요성 증가, 자동심사승인을 위해 도울 수 있는 의사결정 지원 시스템 필요



- 의사결정나무 분석을 이용한 지표 구간화

연체유무를 이용하여 가장 연체 유무가 잘 나누어지는 값을 각종 지수에 기준으로 분류하여 주는 분석

주요 지표들에 대하여 단독 변수로 의사결정나무 분석을 이용하여 분석을 수행

연체 유무에 따른 비등간 구간화 의미

장점) 연체 유무에 따라 가장 효율적으로 분리, 특정한 값 이상/이하 등과 같은 이분법적인 부리가 아닌 특정한 case들에 대해서도 분리 가능

이상치 및 0으로 된 경우는 제외하고, 모델링 수행

- 다차원 회귀방정식 사용

```
0.0000000002941 * 육개월요구불평잔 +
0.00000002148 * 육개월저축성평잔 +
-0.01494 * [주거상황코드=01] +
-0.1009 * [주거상황코드=02] +
-0.1833 * [주거상황코드=03] +
0.006443 * [주거상황코드=04] +
-0.1064 * [주거상황코드=05] +
0.3379 * [주거상황코드=06] +
-2.194 * [주거상황코드=07] +
0.6749 * [동거가족코드=1] +
0.5529 * [동거가족코드=2] +
0.137 * [동거가족코드=3] +
1.041 * [동거가족코드=4] +
0.555 * [동거가족코드=5] +
0.7311 * [CB신용등급=0] +
2.612 * [CB신용등급=1] +
-0.983 * [CB신용등급=10] +
2.108 * [CB신용등급=2] +
2.068 * [CB신용등급=3] +
1.842 * [CB신용등급=4] +
1.386 * [CB신용등급=5] +
0.8954 * [CB신용등급=6] +
0.5976 * [CB신용등급=7] +
0.6793 * [CB신용등급=8] +
+ -1.629
```

attribute를 활용하여 수식개발

⇨ 대출 연체 가능성 판단

data mining을 통해서 attribute을 선정

⇨ 시각화된 틀로 보여줌