

의사결정트리(Decision Tree) 알고리즘 기반 데이터 시각  
화 추천(Visualization recommendation) 알고리즘 개발  
2019.10~2019.12

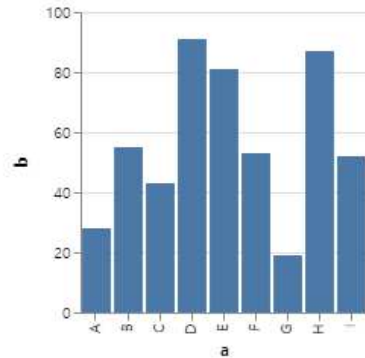
Table of Contents	Page No
CSV와 같은 데이터 셋으로부터 자동적으로 차트를 생성할 수 있는 알고리즘에 대해 알아보고, 적용가능성을 검토한다.	3
시각화 추천에 관한 논문을 찾아보고, 구현 방법에 대해서 간략히 정리하여 적용가능성을 검토한다.	5
시각화 추천에 관한 논문을 찾아보고, 구현 방법에 대해서 간략히 정리하여 적용가능성을 검토한다.	9
시각화 추천에 관한 논문을 찾아보고, 구현 방법에 대해서 간략히 정리하여 적용가능성을 검토한다.	11
시각화 추천 논문에서 기존 'RULE-BASE SYSTEM'과 'ML-BASED SYSTEM'을 비교하고, 제안한 방법을 실험하기 위한 실험환경을 구성한다.	13
구성한 실험환경을 바탕으로 데이터 시각화 추천 흐름도에 따른 데이터 분석을 수행한다.	15
데이터 분석과 관련하여 특징 추출을 기술하고 추출한 특징값을 이용하여 실험을 수행한다.	17
실험결과를 바탕으로 신경망의 입력 및 타 알고리즘에 사용되는 특징의 중요도를 분석한다.	22
로컬과 클라우드 환경 각각에서 은닉층 수 및 데이터 셋 크기에 따른 훈련 정확도와 테스트 정확도를 측정한다.	23
데이터 시각화 추천 논문을 조사하고, 적용가능성을 검토한다.	25
데이터 시각화 추천 알고리즘을 적용하기 위한 알고리즘 구조도 및 클래스 정의하고 세부사항에 대하여 명시한다.	27
데이터 시각화 추천 논문에서 제안하는 방법에 대하여 서술하고 구현한다.	28
데이터 시각화 추천 논문에서 제안하는 방법에 대하여 추가로 서술하고 DB와 연동하기 위한 코드를 구현한다.	30
Partial order-based visualization selection 방법과 Hybrid ranking method를 각각 실험하고 결과를 비교한다.	32
화면에 추천차트를 출력하기 위한 표준출력으로 나타내고, DB에서 해당 차트를 관리하기 위한 인터페이스를 설계한다. 또한 영문 한글 버전의 차트 추천 UI/UX를 고안한다.	33
차트 추천 아이콘을 디자인하고, 차트 추천 결과를 도출하기 위한 최종 스크립트 전달인자에 따라 출력결과를 확인한다.	35

# 연구 노트

## 연구 목표

CSV와 같은 데이터 셋으로부터 자동적으로 차트를 생성할 수 있는 알고리즘에 대해 알아보고, 적용가능성을 검토한다.

## 개념



[Export as PNG](#)[View Source](#)[Open in Vega Editor](#)

그림 1. vega-lite(링크로 가져오는(CND) 방식)

데이터 시각화 관련 연구를 진행하는 washington University의 Interactive Lab은 D3.js 기반 차트 라이브러리인 Vega와 vega-lite(그림 1)를 개발하였고, 해당 차트를 기반으로 데이터 특성에 따른 차트 추천 서비스인 Voyager를 개발하였음.

vega & vega-lite는 D3.js와 Typescript 기반으로 개발한 것으로 수동적인 차트 추천과 차트 추천을 보완하는 대화식 시스템임. 데이터에서 추천을 위한 차트 사양을 만들기 새로운 언어와 시스템. 20만 건의 데이터를 그리는데 canvas 기반으로 그려지고 4~5초 가량 소요됨.

## Voyager: Unifying Chart Specification & Recommendations

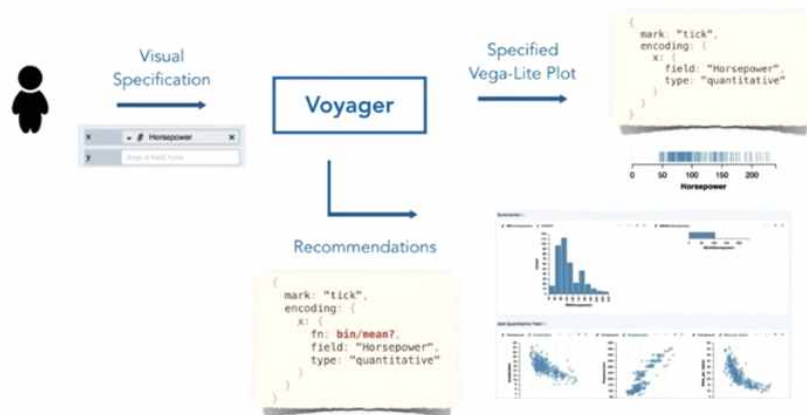


그림 2. 차트 구성 방법

Voyager의 특징으로는 데이터의 특성에 따른 자동 차트 추천 기능이 있으며 사용자가 Drag&Drop 방식으로 직접 차트를 구성할 수 있음(그림 2). 사용자가 specification을 지정하면 Voyager는 vega-lite 설정을 자동으로 만들고, CompassQL을 통해 추천 Recommendation 스펙도 생성함.

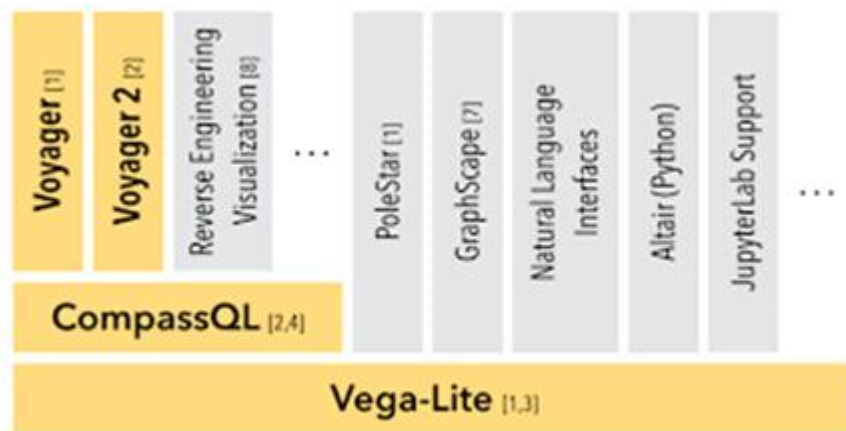


그림 3. Voyager Architecture

Voyager 3개의 layer로 구성됨(그림 3). Application인 Voyager2를 제공, CompassQL로 차트 추천엔진 배치, vega-lite로 차트 제공하는 과정을 layer-3, layer-2, layer-1에서 각각을 수행함.

\*참고문헌: <https://vega.github.io/voyager/>

# 연구 노트

## 연구 목표

시각화 추천에 관한 논문을 찾아보고, 구현 방법에 대해서 간략히 정리하여 적용가능성을 검토한다.

## 개념

Towards A general-Purpose Query Language for Visualization Recommendation 은 2016년 Human-In-the-Loop Data Analytics 워크샵에서 발표된 논문임.

Data Query	Completely or Partially Suggested	<b>Data Query Recommenders</b> SeeDB [21] Rank-by-Feature Framework [16] Scagnostics [18,24] ...	<b>Hybrid Recommenders</b> Voyager [25] VizDeck [14] Small Multiples, Large Singles [19] ...
	Completely Specified	<b>Manual Specification Tools</b> Polaris [17] ggplot2 [22] Vega-Lite [3] ...	<b>Encoding Recommenders</b> APT [12] Tableau's Show Me [13] Spotfire Recommendation [2] ...
		Completely Specified	Completely or Partially Suggested
		Visual Encoding	

Figure 2: Matrix of visualization tools grouped by the type of recommendations.

그림 1. 추천 타입에 따른 그룹화된 시각화 툴 매트릭스

Tableau의 Show me, Spotfire Recommendations와 같은 인코딩 추천자는 데이터에 대한 효과적인 그래픽 표현을 제안하나 데이터 쿼리를 수동으로 지정해야 함(그림 1).

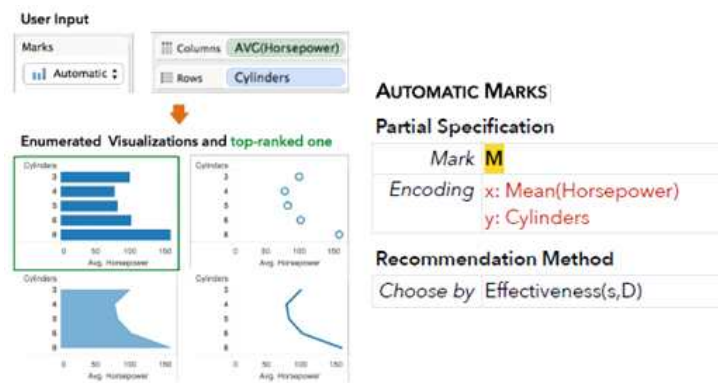


그림 2. Tableau's Show Me

Tableau는 Enumeration, Ranking, Reducing Redundancy의 추천과정을 거침. Enumeration 단계에서는 사용자의 의도와 일치하는 시각화 집합을 검색하여 열거함. Ranking 단계에서는 정렬된 권장 목록을 생성함. Reducing Redundancy 단계에서는 중복 추천을 피하기 위해 유사한 후보를 그룹화하고, 각 그룹에서 최상의 등급의 대표를 선택함. 기본 차트 유형 집합에 대해 각각 하나의 인스턴스만 제안함(그림 3).

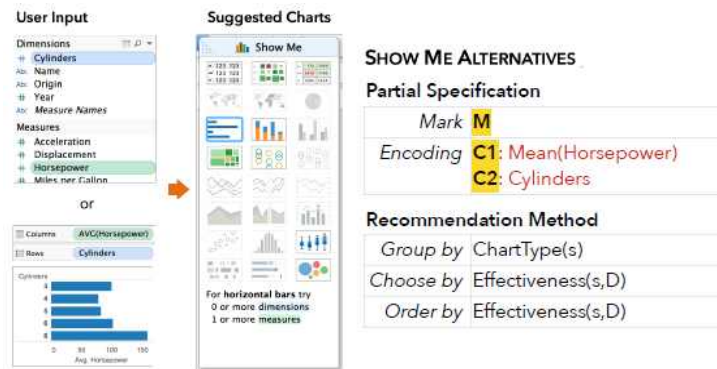


그림 3. Show Me Alternatives

논문에서 제안하는 Visualization Query language는 Partial Specification, Choosing and Ordering, Grouping 순서로 진행함. Partial Specification 단계에서는 Mark 속성 설정에 따라 시스템의 가능한 모든 마크 유형을 열거함(그림 4).

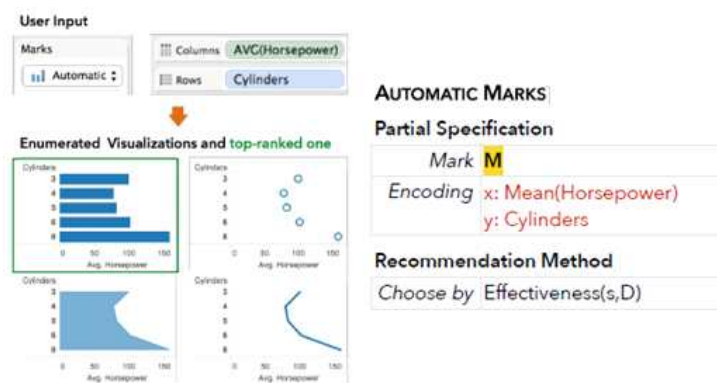


그림 4. Partial Specification 과정1

그림 5의 query는 1D 및 2D 시각화를 모두 열거함. 결과 집합의 조합 폭발을 피하기 위해 열거 지정자에 대한 제약 조건이 포함될 수 있음. 집합 함수 A는 집합 없음 또는 평균으로 제한함(그림 6).



그림 5. Partial Specification 과정2

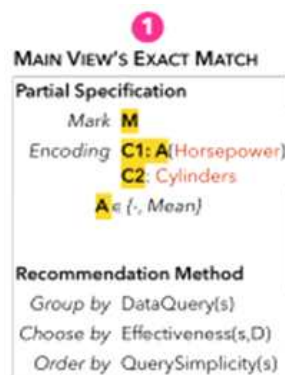


그림 6. Partial Specification 과정3

Choosing and Ordering 단계에서 쿼리는 기준 속성을 지정하여 최고 권장 사항을 선택하는 방법을 정의하거나 속성별 순서를 설정하여 점수 순서대로 순위가 매겨진 목록 권장 사항을 생성할 수 있음. 선택 기준 및 순서 기준 속성은 쿼리 엔진에서 기본적으로 제공하거나 개발자가 사용자 정의 함수로 정의한 순위 지정 방법을 나타냄(그림 7).



그림 7. Choosing and Ordering 과정

Grouping 단계에서 중복성을 줄이기 위해 쿼리에는 그룹화 절을 포함시켜 그룹화를 유도하는 주요 기능을 제공함. 데이터 쿼리를 매칭함으로써 시각화를 그룹화하고, 그룹을 나타내기 위해 가장 효과적인 시각화를 선택함(그림 8).

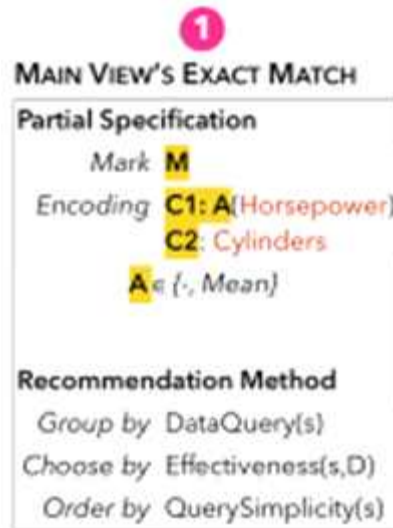


그림 8. Grouping 과정



# 연구 노트

## 연구 목표

시각화 추천에 관한 논문을 찾아보고, 구현 방법에 대해서 간략히 정리하여 적용가능성을 검토한다.

## 개념

Tableau의 Show me는 새로운 사용자와 숙련된 사용자 모두가 사용하도록 설계되었음. 자동 표시의 기본 뷰 표시 형식을 Default로 VizQL 식의 row 및 column 구조를 사용함. 시트에 추가 명령은 그래픽 설계 모범 사례를 사용하여 뷰에 추가할 필드의 속성과 뷰의 VizQL 사양을 사용함.

C = Categorical (discrete and dimension)  
 Cdate = Categorical date (date or date&time)  
 Q = Quantitative (continuous)  
 Qd = Quantitative dependent (measure)  
 Qi = Quantitative independent or Qdate (dimension)

그림 1. Automatic Marks

Automatic Marks는 그림 1과 같은 분류가 포함된 Tableau 데이터 모델을 활용함. Q 필드를 독립 또는 종속으로 분류하는 것이 특히 중요함. 종속 변수는 독립 변수의 함수로 고안되었음. 데이터 웨어하우스에서 사용되는 데이터 큐브의 규칙에 따라 독립 변수를 차원이라고 하고 종속 변수를 측정이라고 함.

Table 1: Automatic marks rules

Pane Type		Mark Type	View Type
Field	Field		
C	C	Text	Cross-tab
Qd	C	Bar	Bar view
Qd	Cdate	Line	Line view
Qd	Qd	Shape	Scatter plot
Qi	C	Gantt	Gantt view
Qi	Qd	Line	Line view
Qi	Qi	Shape	Scatter plot

그림 2. 자동 마크 규칙

Automatic Marks는 행 및 열 선반에서 가장 오른쪽 필드를 사용하여 그림 2와 같이 차 유형과 마크 유형을 결정함. 이 경우 일반적인 케이스와 동일한 mapping을 갖는 특정 케이스는 생략됨.

**Text Tables:** at least 1 field, rank 1



Text tables have the lowest rank because their primary utility is to look up specific values. The higher ranked commands present views that encode data graphically, which support other tasks such as comparison. Although text tables have a low rank, their condition is easily met. The text table command can handle a large number of fields and will always be available as a default for Show Me Alternatives. Heat maps are a related command that is not ranked.

**Aligned Bars:** at least 1 Q, rank 2



Bars are effective for comparing values because the human visual system is good at comparing bar lengths, particularly when they are aligned. Aligned Bars are a common default when the input includes a quantitative field unless the input also includes a date field or two quantitative fields. However, aligned bars can involve a lot of scrolling when multiple categorical fields are shown. The next command handles this case.

**Stacked Bars:** at least 2 C, at least 1 Q, rank 3 with at least 3 C



Stacked bars require less scrolling than aligned bars when there are multiple categorical fields in the view. There are two additional bar commands that are not ranked.

**Discrete Lines:** at least 1 Cdate, at least 1 Q, rank 4



A line view is a better default than a bar view when the input includes a date field because it is more effective for showing trends. This command treats the date field discretely. There is an unranked line command that treats the date field continuously. The primary difference is how the axes are drawn. We judged that the discrete version was more appropriate for the typical Tableau user.

**Scatter Plots:** between 2 and 4 Q, rank 5 for 2 Q



Scatter plots are very effective for comparing two values, particularly with categorical fields on the shape shelf. However, they grow less effective when the input has more than two quantitative fields. Lower ranked commands will be used as the default when the input has more than four quantitative fields.

**Gantt Charts:** at least 1 C, at least 1 Qi, 1 to 2 Q, rank 6



Although specialized, Gantt charts are effective for showing duration with respect to a quantitative independent. In Tableau, Gantt charts are also an effective way to show the distribution of a measure. Although highly ranked, the conditions for this command are rarely satisfied.

그림 3. The Show Me Default Ranking

이러한 규칙은 보기 유형을 나타내며 차트 및 그래프를 생성하는 가장 좋은 방법은 교육받은 사람이 이러한 유형의 필드에 대해 일반적으로 선택함.

각 대안 표시 명령은 특정 유형의 보기를 작성함. 각 명령에는 관련 조건이 있으며 일부 명령에는 순위가 있음. 순위가 높은 명령은 일반적으로 특정 뷰 유형을 적용하기 위해 보다 특수한 조건을 충족해야 함(그림 3).

# 연구 노트

## 연구 목표

시각화 추천에 관한 논문을 찾아보고, 구현 방법에 대해서 간략히 정리하여 적용가능성을 검토한다.

## 개념

VizML: A Machine Learning Approach to Visualization Recommendation은 2019년 Human Factors in Computing System 컨퍼런스에서 발표된 논문임.

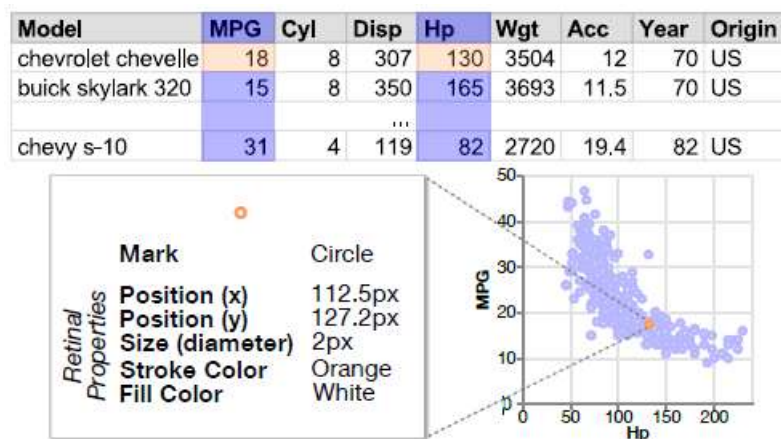


그림 1. Problem Formulation I

분석가는 2D 평면에서 원의 위치로 각 데이터 포인트 쌍을 인코딩하고 크기 및 색상과 같은 다른 많은 속성을 지정함(그림 1).

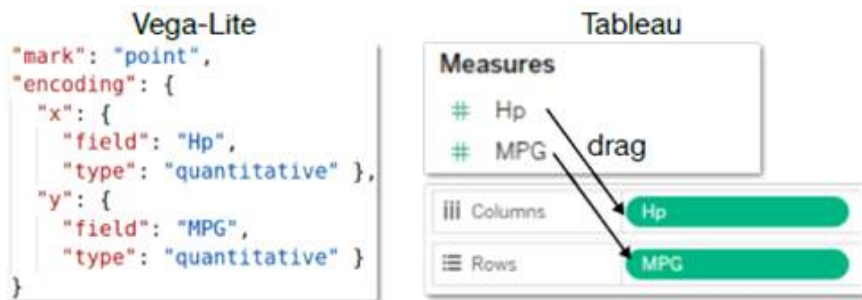


그림 2. Problem Formulation II

맞춤형 시각화를 만들려면 분석가가 표현 도구를 사용하여 인코딩을 자세하게 지정해야 함. 그러나 산점도는 X축 및 Y축을 따라 인코딩할 마크 유형 및 필드를 선택하고 Tableau에서 두 열을 해당 열 및 행 선반에 배치하여 vega-lite 문법으로 지정됨(그림 2).

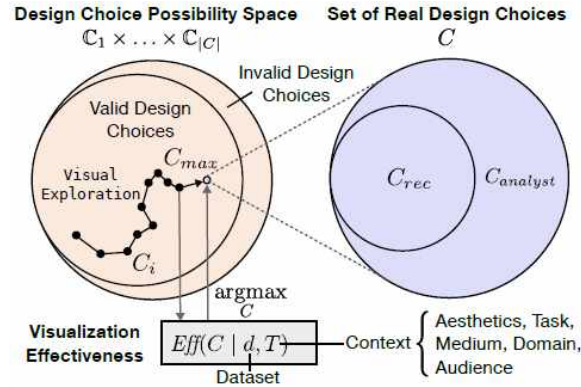


Fig. 1: Creating visualizations is a process of making design choices, which can be recommended by a system or specified by an analyst.

그림 3. 시스템 또는 분석가에 의해 추천된 시각화 생성 과정

모든 디자인 선택이 유효한 시각화를 생성하는 것은 아님. 분석가는 시각화 효과를 극대화하는 디자인 선택  $C_{max}$ 를 선택함(그림 3). 시각화 권장 사항의 목표는 디자인 선택의 일부를 자동 제안하여 시각화 생성 비용을 줄이는 것임.

제안한 방법은 신경망  $G_c$ 와 연결 가중치  $Q_c$ 로 모델링하고, 각  $G_c$ 를 최적화하여 추천 문제를 단순화하였음(그림 4).

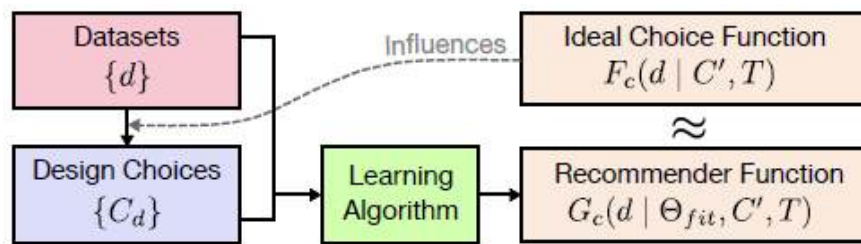


Fig. 2: Basic setup of learning models to recommend design choices with a corpus of datasets and corresponding design choices.

그림 4. 학습 모델 기본 설정

# 연구 노트

## 연구 목표

시각화 추천 논문에서 기존 'RULE-BASED SYSTEM'과 'ML-BASED SYSTEM'을 비교하고, 제안한 방법을 실험하기 위한 실험환경을 구성한다.

## 개념

### 1. RULE-BASE SYSTEM vs ML-BASED SYSTEM

System	Source	N <sub>data</sub>	Generation	Learning Task	Training Data	Features	Model
VizML	Public (Plotly)	10 <sup>6</sup>	Human	Design Choice Recommendation	Dataset-Visualization Pairs	Single + Pairwise + Aggregated	Neural Network
DeepEye	Crowd	1) 33.4K 2) 285K	Rules → Annotation	1) Good-Bad Classif. 2) Ranking	1) Good-Bad Labels 2) Pairwise Comparisons	Column Pair	1) Decision Tree 2) RankNet
Data2Vis	Tool (Voyager)	4,300	Rules → Validation	End-to-End Viz. Generation	Dataset Subset-Visualization Pairs	Raw	Seq2Seq NN
Draco-Learn	Crowd	1,100 + 10	Rules → Annotation	Soft Constraint Weights	Pairwise Comparisons	Soft Constraint Violation Counts	RankSVM

Table 1: Comparison of machine learning-based visualization recommendation systems. The major differences are that of Learning Task definition, and the quantity (N<sub>data</sub>) and quality (Generation and Training Data) of training data.

[그림 1] 머신러닝 기반 시각화 추천 시스템 비교

#### ◦ RULE-BASED VISUALIZATION RECOMMENDER SYSTEMS

-더 많은 데이터, 인코딩 및 작업 유형을 지원[SAGE, BOZ, Show Me]  
-시작적 인코딩 규칙과 선택되지 않은 열을 포함하는 시각화 권장 사항을 결합[Voyager, Explore in Google sheetes, VizDeck, DIVE]

-주요 제한 사항

- 1) 비선형 관계를 인코딩해야 하는 복잡한 과정
- 2) 간단한 규칙을 만드는 것조차 전문가의 판단에 의존
- 3) cold-start 문제
- 4) 권장 사항이 폭발적으로 증가

#### ◦ ML-BASED VISUALIZATION RECOMMENDER SYSTEMS

-DeepEye: 의사 결정 트리 훈련 / 33,412 개의 데이터에 대하여 100명의 학생들이 주석을 달았음.

-Data2Viz: 1-3개의 변수로 구성된 4,300개의 Vega-Lite 예제를 사용하여 학습

-Draco-Learn: 순위화된 시각화 쌍에 대해 훈련된 RankSVM을 사용



-기존 시스템과 차이점

- 1) VizML 모델은 설계 선택을 예측하는 방법으로써 정략적으로 검증
- 2) 데이터 수량 측면에서 DeepEye 및 Data2Vis보다 수십 배 큰
- 3) 기존 시스템의 소수 데이터 세트와 달리 모양, 구조 및 속성이 다양함

## 2. VizML 실험을 위한 환경구성

이름	수정한 날짜	유형	크기
.keep	2019-05-05 오후 3:45	KEEP 파일	0KB
plotly_full.tar.gz	2019-10-14 오후 10:00	압축(GZ) 파일	48,333,46...
plotly_plots_with_full_data_with_all_fields...	2019-10-14 오후 9:05	TSV 파일	226,564KB
plotly_subset_1k.tar.gz	2019-10-14 오후 9:05	압축(GZ) 파일	50,401KB

[그림 2] 실험 데이터

```

miniflow@DESKTOP-0UBK7G:~$
Installing this may take a few minutes.
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: root
adduser: The user 'root' already exists.
adduser: Please enter a username matching the regular expression configured
via the HOME_REGEX[SYSDBM] configuration variable. Use the --force-badname
option to relay this check.
Enter new UNIX username: miniflow
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo.root" for details.

miniflow@DESKTOP-0UBK7G:~$ ls -release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description: Ubuntu 18.04.2 LTS
Release: 18.04
Codename: bionic
miniflow@DESKTOP-0UBK7G:~$ sudo apt-get update && sudo apt-get upgrade
[sudo] password for miniflow:
get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [89.7 kB]
Hit:2 http://archive.ubuntu.com/ubuntu bionic InRelease
get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [98.7 kB]
get:4 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [529 kB]
get:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
get:6 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [5970 kB]
get:7 http://security.ubuntu.com/ubuntu bionic-security/main Translationen [177 kB]
get:8 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [5872 B]
get:9 http://security.ubuntu.com/ubuntu bionic-security/restricted Translationen [3286 B]
get:10 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [513 kB]
get:11 http://security.ubuntu.com/ubuntu bionic-security/universe Translationen [204 kB]
get:12 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [5250 B]
get:13 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translationen [2484 B]
get:14 http://archive.ubuntu.com/ubuntu bionic/universe Translationen [4341 kB]
get:15 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
get:16 http://archive.ubuntu.com/ubuntu bionic/multiverse Translationen [1108 kB]
get:17 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [752 kB]
get:18 http://archive.ubuntu.com/ubuntu bionic-updates/main Translationen [1270 kB]
get:19 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [15.7 kB]
get:20 http://archive.ubuntu.com/ubuntu bionic-updates/restricted Translationen [4256 B]
get:21 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1072 kB]
get:22 http://archive.ubuntu.com/ubuntu bionic-updates/universe Translationen [512 kB]
get:23 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [7354 B]
get:24 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse Translationen [3944 B]
get:25 http://archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [5512 B]
get:26 http://archive.ubuntu.com/ubuntu bionic-backports/main Translationen [1644 B]
get:27 http://archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [4260 B]
get:28 http://archive.ubuntu.com/ubuntu bionic-backports/universe Translationen [1396 B]
Fetched 18.0 MB in 1min 57s (185 kB/s)
Reading package lists... Done
E: Invalid operation upgrade
miniflow@DESKTOP-0UBK7G:~$
  
```

Windows 제품 인증  
(선택)으로 이동하여 Windows를 정품 인증하십시오

[그림 3] 패키지 설치 화면

-실험 데이터는 크기에 따라 50MB / 46GB로 나뉨(그림 2).

-필요 패키지 설치 및 환경설정은 그림 참고(그림 3).

\*참고문헌: VizML:A Machine Learning Approach to Visualization Recommendation  
(2019 CHI Conference on Human Factors in Computing Systems)

# 연구 노트

## 연구 목표

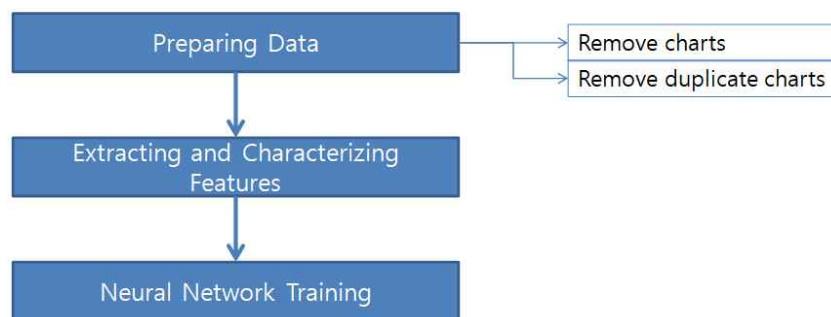
구성한 실험환경을 바탕으로 데이터 시각화 추천 흐름도에 따른 데이터 분석을 수행한다.

## 개념

### 1. VizML 실험을 위한 환경구성

plotly_plots_with_full_data_with_all_fields_and_header_1k.tsv	2019-10-14 오후 9:05	TSV 파일	226,564KB	221 MB
plotly_plots_with_full_data_with_all_fields_and_header_100k.tsv	2019-04-27 오전 3:56	TSV 파일	16,789,80...	16 GB
plotly_plots_with_full_data_with_all_fields_and_header_full.tsv	2018-08-21 오전 8:05	TSV 파일	214,921,1...	204 GB

[그림 1]



[그림 2] 데이터 시각화 추천 흐름도

```

fid chart_data layout table_data
maragones:847
"[{"ysrc": "maragones:848:e1002e",
  "xsrc": "maragones:848:65303e",
  "type": "bar", "opacity": 0.6,
  "marker": {"line": {"color": "rgb(8,48,107)",
    "width": 1.5},
  "color": "rgb(158,202,225)"}]}]"
{"title": "\u00daltima actualizaci\u00f3n: 2018-01-12 14:23:00"}
{"maragones:848": {"reason": "restricted", "cols": {"y": {"uid": "e1002e", "order": 1, "data": [153, 9, 1776]}, "x": {"uid": "65303e", "order": 0, "data": [{"account", "story", "hashtag"]}}}}]
  
```

[그림 3] 데이터 구조

- 중간 크기의 실험 데이터를 추가하여 압축을 해제한 실험 데이터의 크기는 그림 1과 같이 각 221 MB, 16 GB, 204 GB의 크기를 가짐.
- 데이터 시각화 추천의 흐름도는 그림 2와 같이 Data & Flow chart, Preparing Data, Extracting and Characterizing Features, Neural Network Training의 단계로 나뉘어짐. Preparing Data 단계에서 invalid한 차트 및 중복 차트를 제거하는 과정을 거침.

```

Anaconda Prompt
(keras_env) C:\Users\hbee.jsjo\mindflow\22_visualization_recommendation\data_cleaning>python remove_charts_without_all_data.py
Chunk 1: 10.87s (8.70s)
Final number of charts: 489 (0.49)
Empty fields: 0
Errors: 511
(keras_env) C:\Users\hbee.jsjo\mindflow\22_visualization_recommendation\data_cleaning>

```

[그림 4] invalid 차트 제거

```

Num skipped: 0
Num preserved FIDs: 0
Unique FIDs: 458
Duplicates: 31 (0.064)

```

[그림 5] 중복 차트 제거

hbee.jsjo > mindflow > 22_visualization_recommendation > data				data 검색
이름	수정된 날짜	유형	크기	
Preparing_Data_bak	2019-10-16 오후 7:16	파일 폴더		
.keep	2019-05-05 오후 3:45	KEEP 파일	0KB	
plot_data.tsv	2019-10-14 오후 9:05	TSV 파일	226,564KB	
plot_data_with_all_fields_and_header.tsv	2019-10-15 오후 8:18	TSV 파일	47,378KB	
plot_data_with_all_fields_and_header_deduplicated_one_per_user.tsv	2019-10-16 오후 7:24	TSV 파일	47,297KB	
plot_data_with_all_fields_and_header_1k.tsv	2019-10-15 오후 8:18	TSV 파일	1KB	
유형: TSV 파일	2019-10-14 오후 9:05	TSV 파일	226,564KB	
크기: 46.1MB	2019-10-16 오후 7:24	PKL 파일	10KB	
수정된 날짜: 2019-10-16 오후 7:24				

[그림 6] 사용자 별 차트 데이터

- 중복 차트 제거에 따른 유효한 차트의 수는 그림 4와 같이 221 MB 데이터에서 489건, 처리속도는 10.87초 소요되었음. 또한 중복된 차트를 제거함에 따라 그림 5와 같이 유효한 차트의 수는 458건으로 집계됨.



# 연구 노트

## 연구 목표

데이터 분석과 관련하여 특징 추출을 기술하고 추출한 특징값을 이용하여 실험을 수행한다.

## 개념

### 1. 특징 추출 기술

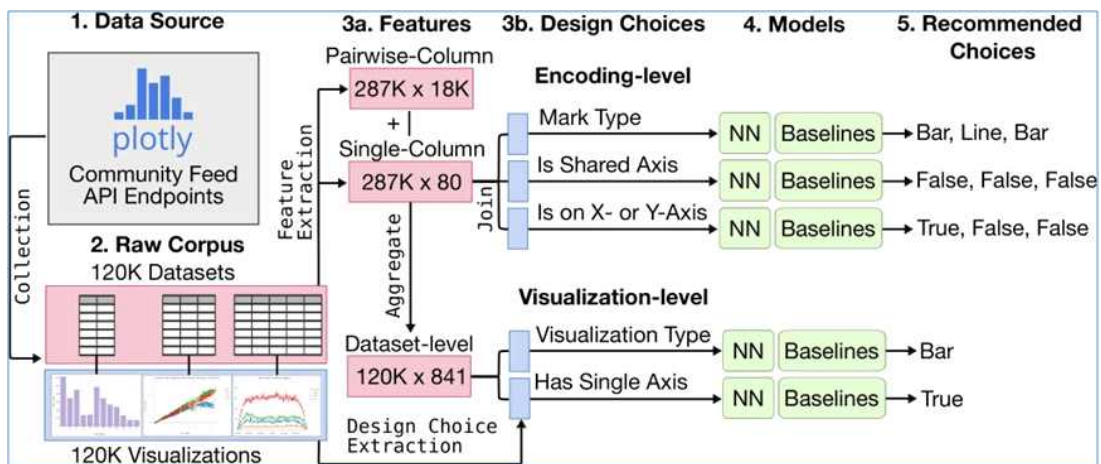


그림 1. 전체 흐름도

## Collection and Cleaning

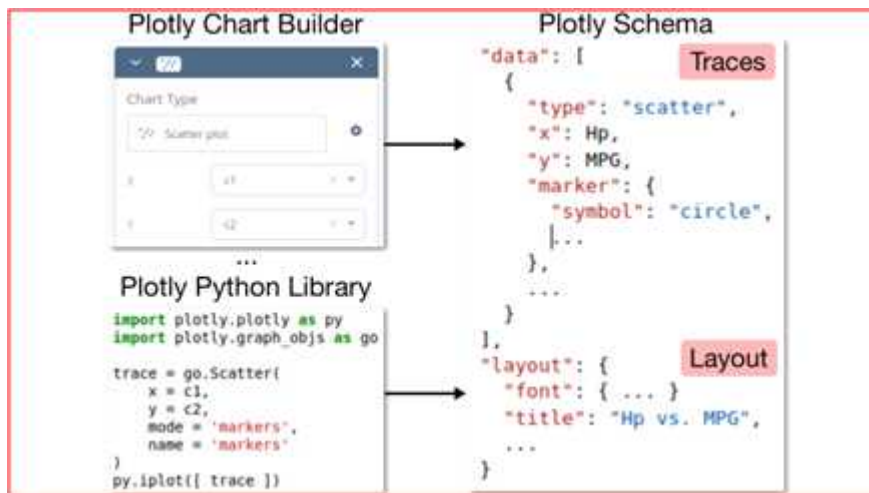


그림 2. Plotly 시각화 스키마

-Plotly의 시각화는 선언적 스키마로 지정함. 이 스키마에서 시각화는 두 개의 데이터 구조로 지정됨. 첫 번째는 데이터 모음이 시각화되는 방법을 지정하는 나열 목록이고, 두 번째는 축 레이블 및 주석과 같이 데이터에서 통일화된 시각화의 미학적 측면을 지정하는 것임.

## Data Description

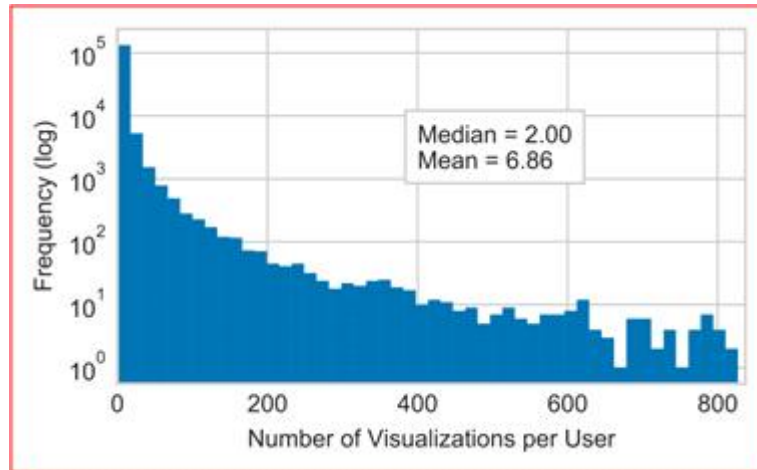


그림 3. 사용자 별 시각화 횟수

-Plotly corpus에는 사용량이 다양한 143,007명의 고유한 사용자가 만든 시각화가 포함되어 있음.

\*Plotly corpus: 데이터 셋 시각화 쌍 집합

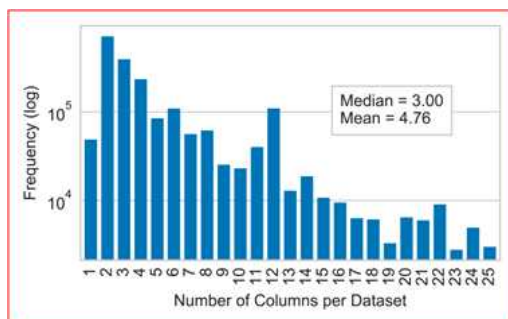


그림 4. 데이터 셋 별 열 숫자

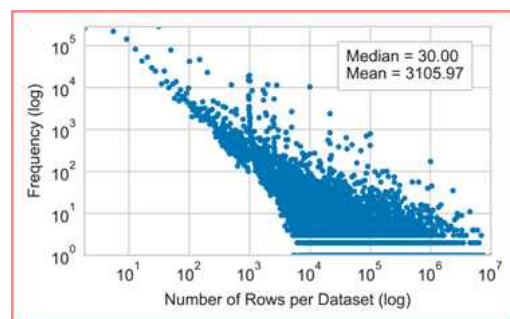


그림 5. 데이터 셋 별 행 숫자

-데이터 셋은 열과 행의 수에 따라 매우 다양함. 일부 데이터 셋에는 100개 이상의 열이 포함되지만 94.97%에는 25개 이하의 열이 포함됨.

-시각화의 98.32%는 하나의 데이터 셋만 사용하였음. 또한 시각화의 90% 이상이 데이터 셋의 모든 열을 사용하므로 데이터 쿼리 선택을 처리할 수 없음.

## Feature Extraction

(a) 81 single-column features describing the dimensions, types, values, and names of individual columns.

Dimensions (1)	
Length (1)	Number of values
Types (8)	
General (3)	Categorical (C), quantitative (Q), temporal (T)
Specific (5)	String, boolean, integer, decimal, datetime
Values (58)	
Statistical [Q, T] (16)	Mean, median, range (Raw/normalized by max), variance, standard deviation, coefficient of variance, minimum, maximum, (25th/75th) percentile, median absolute deviation, average absolute deviation, quantitative coefficient of dispersion
Distribution [Q] (14)	Entropy, Gini, skewness, kurtosis, moments (5-10), normality (statistic, p-value), is normal at ( $p < 0.05$ , $p < 0.01$ ).
Outliers (8)	(Has/%) outliers at ( $1.5 \times \text{IQR}$ , $3 \times \text{IQR}$ , 99%ile, $3\sigma$ )
Statistical [C] (7)	Entropy, (mean/median) value length, (min, std, max) length of values, % of mode
Sequence (7)	Is sorted, is monotonic, sortedness, (linear/log) space sequence coefficient, is (linear/space) space
Unique (3)	(Is/##/%) unique
Missing (3)	(Has/##/%) missing values
Names (14)	
Properties (4)	Name length, # words, # uppercase characters, starts with uppercase letter
Value (10)	("x", "y", "id", "time", digit, whitespace, "\$", "€", "£", "¥") in name

그림 6. 81 single-column features

(b) 30 pairwise-column features describing the relationship between values and names of pairs of columns.

Values (25)	
[Q-Q] (8)	Correlation (value, $p$ , $p < 0.05$ ), Kolmogorov-Smirnov (value, $p$ , $p < 0.05$ ), (has, %) overlapping range
[C-C] (6)	$\chi^2$ (value, $p$ , $p < 0.05$ ), nestedness (value, = 1, > 0.95%)
[C-Q] (3)	One-Way ANOVA (value, $p$ , $p < 0.05$ )
Shared values (8)	is identical, (has/##/%) shared values, unique values are identical, (has/##/%) shared unique values
Names (5)	
Character (2)	Edit distance (raw/normalized)
Word (3)	(Has, #, %) shared words

그림 7. 30 pairwise-column features

(c) 16 Aggregation functions used to aggregate single- and pairwise-column features into 841 dataset-level features.

Categorical (5)	Number (#), percent (%), has, only one (#=1), all
Quantitative (10)	Mean, variance, standard deviation, coefficient of variance (CV), min, max, range, normalized range (NR), average absolute deviation (AAD) median absolute deviation (MAD)
Special (1)	Entropy of data types

그림 8. 16 Aggregation functions

-81개의 단일 열 특징에 대해 그림 6에 나타냄. 이러한 특징은 4가지 범주로 나누어짐. 또한 30개 컬럼 쌍의 특징을 그림 7에 나타내었으며 값(value)과 이름(name)의 2가지 범주로 분류됨.

-그림 8의 16개의 집계함수를 사용하여 단일 열 및 쌍열 특징을 집계하여 그림 9와 같이 841 데이터 수준 특징을 만들.

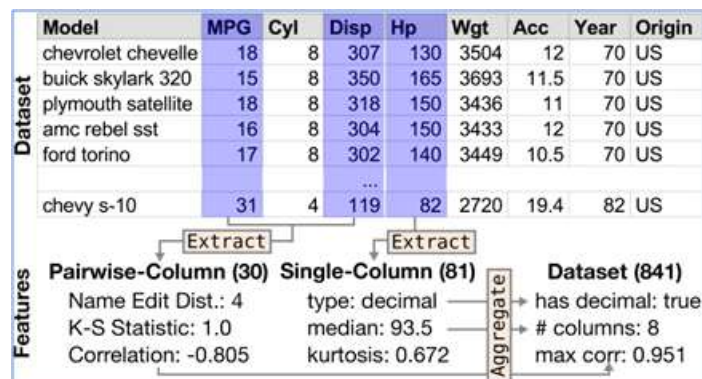


그림 9. 841 데이터 수준의 특징

## Design Choice Extraction

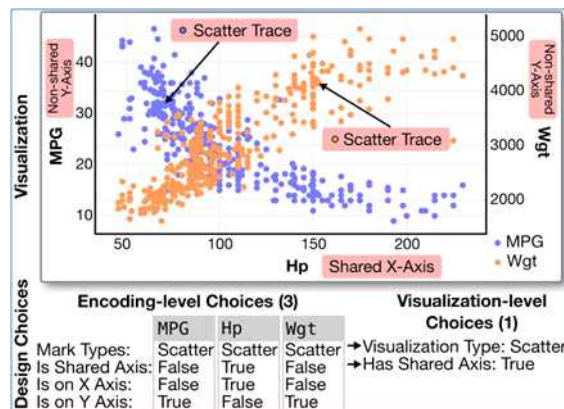


그림 10. Plotly 시각화 설계 선택 예시

-Plotly의 각 시각화는 데이터 컬렉션을 시각적 요소와 연결하는 단서로 구성. 따라서 이러한 추적을 파싱하여 분석가의 설계 선택을 추출(그림 10 참고).

\*인코딩 레벨 디자인 선택: 산포, 선, 막대와 같은 마크 유형.

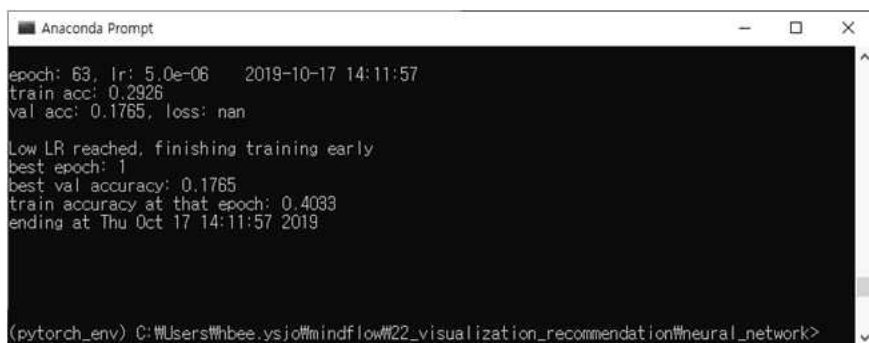
\*X 또는 Y 열 인코딩. 어떤 열이 어떤 축에 표시되는지.

\*X 또는 Y 열이 해당 축을 따라 표시: 단일 열인지 여부.

## 2. 추출한 특징값을 이용하여 실험

```
parameters = {  
    'batch_size': 200,  
    'num_epochs': 100,  
    'hidden_sizes': [800, 800, 800],  
    'learning_rate': 5e-4,  
    'output_dim': 3,  
    'weight_decay': 0,  
    'dropout': 0.00,  
    'patience': 20,  
    'threshold': 1e-3,  
    'model_prefix': 'agg',  
    'save_model': False,  
    'print_test': True,  
    'test_best': False  
}
```

그림 11. 신경망의 학습에 사용하는 파라미터



```
epoch: 63, lr: 5.0e-06   2019-10-17 14:11:57  
train acc: 0.2926  
val acc: 0.1765, loss: nan  
  
Low LR reached, finishing training early  
best epoch: 1  
best val accuracy: 0.1765  
train accuracy at that epoch: 0.4033  
ending at Thu Oct 17 14:11:57 2019  
  
(pytorch_env) C:\Users\hbee.jsjo\mindflow\22_visualization_recommendation\neural_network>
```

그림 12. 훈련 결과

-특징 추출 기술한 부분을 바탕으로 신경망에 입력되는 841개의 특징값은 그림 11과 같은 신경망 구조에 훈련데이터로 사용됨. 신경망의 학습 횟수는 100, 학습률은 0.0005, 은닉층은 구조는 입력과 출력을 제외하고 800개의 뉴런있는 3개 은닉층으로 구성됨.

-그림 11의 신경망의 학습에 사용되는 파라미터에 따라 훈련결과는 그림 12와 같음. 총 100 epoch까지 수행하나 63 epoch에서 정확도 개선이 없어서 훈련이 중지되었음. 이때 훈련 정확도와 평가 정확도는 각각 29.3%와 17.7%를 달성하였고 손실은 측정되지 못함.



# 연구 노트

## 연구 목표

실험결과를 바탕으로 신경망의 입력 및 타 알고리즘에 사용되는 특징의 중요도를 분석한다.

## 개념

(a) Prediction accuracies for two visualization-level tasks.							(b) Prediction accuracies for three encoding-level tasks.						
Model	Features	d	Visualization Type			HSA	Model	Features	d	Mark Type			XY
			C=2	C=3	C=6					C=2	C=3	C=6	
NN	D	15	66.3	50.4	51.3	84.1	NN	D	1	65.2	44.3	30.5	49.9
	D+T	52	75.7	59.6	60.8	86.7		D+T	9	68.5	46.8	35.0	57.3
	D+T+V	717	84.5	77.2	87.7	95.4		D+T+V	66	79.4	59.4	76.0	95.5
	All	841	86.0	79.4	89.4	97.3		All	81	84.9	67.8	82.9	98.3
NB	All	841	63.4	49.5	46.2	72.9	NB	All	81	57.6	41.1	27.4	70.0
KNN	All	841	76.5	59.9	53.8	81.5	KNN	All	81	72.4	51.9	37.8	65.6
LR	All	841	81.8	64.9	69.0	90.2	LR	All	81	73.6	52.6	43.7	79.1
RF	All	841	81.2	65.1	66.6	90.4	RF	All	81	78.3	60.1	46.7	83.4
N <sub>raw</sub> (in 1000s)			42.2	87.0	99.3	119	N <sub>raw</sub> (in 1000s)			94.7	163	183	287

Table 2: Design choice prediction accuracies for five models, averaged over 5-fold cross-validation. The standard error of the mean was  $< 0.1\%$  for all results. Results are reported for a neural network (NN), naive Bayes (NB), K-nearest neighbors (KNN), logistic regression (LR), and random forest (RF). Features are separated into four categories: dimensions (D), types (T), values (V), and names (N). N<sub>raw</sub> is the size of the training set before resampling, d is the number of features, C is the number of outcome classes. HSA = Has Shared Axis, ISA = Is Shared X-axis or Y-axis and XY = Is on X-axis or Y-axis.

그림 1. 5가지 모델의 예측 정확도 비교

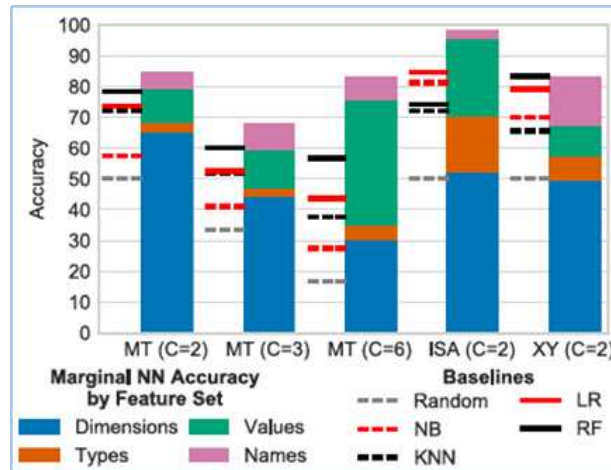


그림 2. 예측작업 정확도에 따른 특징 별 중요도

- 모델의 성능은 일반적으로  $NB < KNN < LR < RF < NN$  순서로 나타남. 대부분의 경우 RF 및 LR의 성능이 NN의 성능보다 크게 떨어지지 않음.

# 연구 노트

## 연구 목표

로컬과 클라우드 환경 각각에서 은닉층 수 및 데이터 셋 크기에 따른 훈련 정확도와 테스트 정확도를 측정한다.

## 개념

1. 로컬 환경에서 10만 건의 데이터 셋에 대하여 신경망을 학습시키고 훈련정확도 및 테스트 정확도를 측정함.

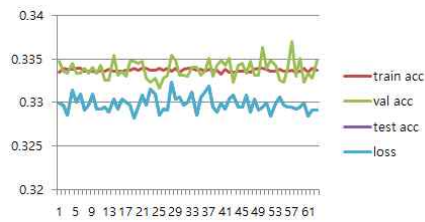


그림 1. Hidden layer 800

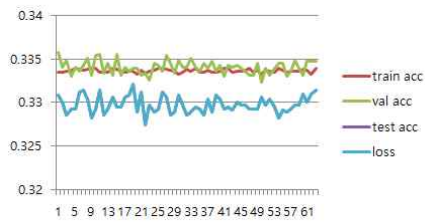


그림 2. Hidden layer 1,000

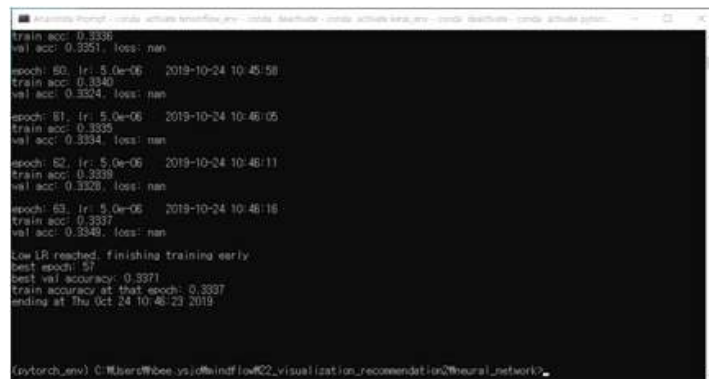


그림 3. Hidden layer 800

Hidden layer	800	1000
Accuracy		
Train Accuracy	0.3337	0.3339
Test Accuracy	0.3349	0.3347

```

Anaconda Prompt - conda activate tensorflow_env - conda deactivate - conda activate keras_env - conda deactivate - conda activate pytorch...
train acc: 0.3339
val acc: 0.3339, loss: nan
epoch: 60, lr: 5.0e-06 2019-10-24 12:56:38
train acc: 0.3338
val acc: 0.3332, loss: nan
epoch: 61, lr: 5.0e-06 2019-10-24 12:56:46
train acc: 0.3337
val acc: 0.3349, loss: nan
epoch: 62, lr: 5.0e-06 2019-10-24 12:56:55
train acc: 0.3339
val acc: 0.3347, loss: nan
epoch: 63, lr: 5.0e-06 2019-10-24 12:57:02
train acc: 0.3339
val acc: 0.3347, loss: nan
Low LR reached, finishing training early
best epoch: 1
best val accuracy: 0.3358
train accuracy at that epoch: 0.3335
ending at Thu Oct 24 12:57:10 2019

(pytorch_env) C:\Users\bee-yj\OneDrive\work\K2_visualization_recommendation\neural_network>

```

그림 4. Hidden layer 1,000

2. 클라우드 환경(P100)에서 200만 건의 데이터 셋에 대하여 신경망을 학습시키고 훈련정확도 및 테스트 정확도를 측정함.

```

root@ec2-238f0325: /home/bee-yj/OneDrive\work\K2_visualization_recommendation\neural_network
train acc: 0.9936
val acc: 0.9263, loss: 0.7763
epoch: 45, lr: 5.0e-06 2019-10-24 17:02:10
train acc: 0.9936
val acc: 0.9261, loss: 0.7804
epoch: 46, lr: 5.0e-06 2019-10-24 17:03:10
train acc: 0.9936
val acc: 0.9263, loss: 0.7811
epoch: 47, lr: 5.0e-06 2019-10-24 17:04:10
train acc: 0.9936
val acc: 0.9263, loss: 0.7822
epoch: 48, lr: 5.0e-06 2019-10-24 17:04:09
train acc: 0.9936
val acc: 0.9264, loss: 0.7827
Low LR reached, finishing training early
best epoch: 48
best val accuracy: 0.9264
train accuracy at that epoch: 0.9937
ending at Thu Oct 24 17:07:29 2019

[Name: /root@ec2-238f0325: neural_network]#

```

그림 5. Hidden layer 1,000

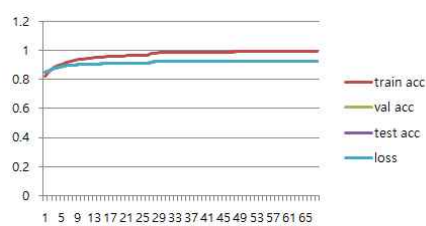


그림 6. Hidden layer 1,000

Hidden layer	
Accuracy	1000
Train Accuracy	0.9936
Test Accuracy	0.9264



# 연구 노트

## 연구 목표

데이터 시각화 추천 논문을 조사하고, 적용가능성을 검토한다.

## 개념

\*참고문헌: DeepEye: Towards Automatic Data Visualization

LUO, Yuyu, et al. DeepEye: Towards Automatic Data Visualization. In: *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 2018. p. 101-112.

‘흥미로운’ 차트는 세 가지 관점에서 정의됨.

- (1) Deviation-based
- (2) Similarity-based
- (3) Perception-based

(1), (2)는 통계적인 편차나 상관관계에 의해 정량화 될 수 있음. (3)은 다른 참조와 비교하지 않고 데이터를 이해함으로써 매력적인 스토리를 전달할 수 있는 차트를 말함.

## Problems

- DeepEye는 세 가지 문제를 다룸.

- (1) Visualization Recognition (2) Ranking (3) Selection

## Challenges

I. Capturing Human Perception -> Learning from examples, Expert Knowledge

II. Large Search Space -> Database optimization

III. Lack of Ground Truth -> Online chart & Manually annotate to create 'ground truth'

## Search Space

-SELECT:  $m \times (m-1)$

-TRANSFORM:

-ORDERBY: 3 Possibilities

-Together with the four visualization types, the number of all possible visualizations for two columns is:

$$m \times (m-1) \times 44 \times 4 \times 3 = 528m(m-1)$$

## Extensions for One Column and Multiple Columns

$$m^3 \times 44 \times 4 \times 4 = 704 \times m^3$$

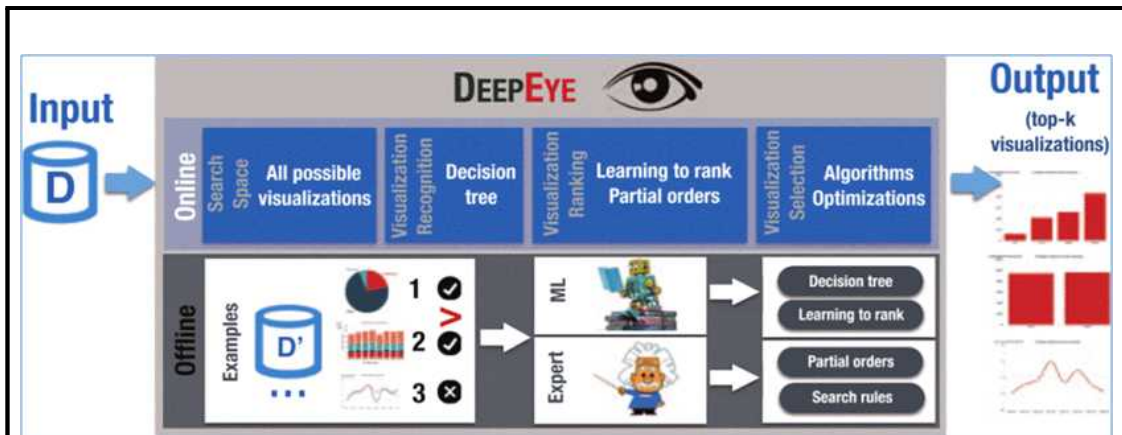


그림 1. DeepEye

Online component: : 가능한 모든 시각화를 식별하고, 훈련 된 분류기를 사용하여 시각화가 좋은지 여부를 결정하고, 학습-순위 모델 또는 전문가가 제공 한 부분 순서를 사용하여 최상위 k 시각화를 선택함.

Offline component: 주어진 데이터 세트와 관련 시각화가 좋은지 여부를 결정하는 이진 분류기 (예 : 의사 결정 트리)와 시각화의 순위를 매기는 학습-등급 모델 등 두 가지 ML 모델을 학습하기위한 예를 사용함.

• Features.

- |  |    |
|--|----|
| (1) The number of distinct values in column $X$ , $d(X)$ .                   | +1 |
| (2) The number of tuples in column $X$ , $ X $ .                             | +1 |
| (3) The ratio of unique values in column, $X$ , $r(X) = d(X) /  X $          | +1 |
| (4) The $\max(X)$ and $\min(X)$ values in column $X$ .                       | +2 |
| (5) The data type $T(X)$ of column $X$ : Categorical, Numerical, Temporal    | +1 |
| (6) The correlation of two columns, $c(X, Y)$ , is a value between -1 and 1. | -1 |
| (7) The visualization type: bar, pie, line, or scatter charts.               | +1 |

We have a feature vector of 14 features.

그림 2. 알고리즘에 사용하기 위한 특징 벡터

**시각화 인식:** 데이터 집합과 지정된 시각화 유형의 열 조합이 제공되어 출력의 양호 여부를 결정합니다. 따라서 의사 결정 트리를 사용하는 이진 분류기가 필요함.

**시각화 순위:** 두 개의 시각화 노드가 주어지면 어느 것이 더 나은지 결정하기 위해 순위 매김 모델을 사용함. 이 모델은 순위 지정 작업에서 모델을 학습하기 위한 ML 기술.

**시각화 선택:** 시각화 노드 세트를 입력으로 지정하면 순위가 매겨진 목록을 출력함.

# 연구 노트

## 연구 목표

데이터 시각화 추천 알고리즘을 적용하기 위한 알고리즘 구조도 및 클래스 정의하고 세부사항에 대하여 명시한다.

## 개념

DeepEye						
main	features	instance	myGraph	table	table_l	view

features.py	Type class	데이터 유형에 따라 세가지 범주로 분류 ex) categorical, numerical, temporal
	Features class	데이터의 특성 정보를 저장 ex) name, type, origin, min, max..

instance.py	ViewPosition class	테이블의 위치와 차트의 위치를 목록에 기록 ex) table_pos, view_pos
	Instance class	각 차트의 점수를 매기는 작업 ex) M, W

view.py	Chart class	차트 목록 정의 ex) bar, line, scatter, pie
	View class	차트 제목, 가로 및 세로 축 이름, 차트 유형, M, Q, W값 등의 차트 정보를 기록

table_l.py	Table class	원본 데이터를 분류하여 여러 유형의 차트를 생성(머신러닝에 사용)
------------	-------------	--------------------------------------

# 연구 노트

## 연구 목표

데이터 시각화 추천 논문에서 제안하는 방법에 대하여 서술하고 구현한다.

## 개념

### Visualization Ranking Principle

Definition 1: [Visualization Node] A visualization node consists of the original data X, Y, the transformed data X', Y', features F, and the visualization type T.

Case 1.  $X1=X2$  and  $Y1=Y2$

(I)  $X'1=X'2$  and  $Y'1=Y'2$

(i)  $X'1=X'2$  are categorical: pie/bar charts are better

-If Y'1 and Y'2 are obtained by AVG, the **bar charts** are better

-It would better to use **bar charts** if there are many categories(example $\geq 10$ )

(ii)  $X'1=X'2$  are numerical: scatter/line charts are better

-If there is a correlation between X' and Y', the **scatter charts** are better

-If there is no correlation, **line charts** are better

(II)  $X'1 \neq X'2$  or  $Y'1 \neq Y'2$

Case 2.  $X1 \neq X2$  or  $Y1 \neq Y2$ , and  $\{X1, Y1\} \cap \{X2, Y2\} \neq \Phi$

Case 3.  $\{X1, Y1\} \cap \{X2, Y2\} \neq \Phi$

### Partial Order

The matching quality between data and chart  $M(v)$

(i) Pie chart

$$M(v) = \begin{cases} 0 & |d(X)| = 1 \\ & \text{or } \min(Y') < 0 \\ & \text{or } Y' = \text{AVG}(Y) \\ \sum_{y \in Y} -p(y) \log(p(y)) & 2 \leq |d(X)| \leq 10 \\ \frac{10}{|d(X)|} \sum_{y \in Y} -p(y) \log(p(y)) & |d(X)| > 10 \end{cases} \quad (1)$$

(ii) Bar chart

$$M(v) = \begin{cases} 0 & |d(X)| = 1 \\ 1 & 2 \leq |d(X)| \leq 20 \\ \frac{20}{|d(X)|} & |d(X)| > 20 \end{cases} \quad (2)$$

(iii) Scatter chart

$$M(v) = c(X, Y) \quad (3)$$

(iv) Line chart

$$M(v) = \text{Trend}(Y) \quad (4)$$

Normalized Significance

$$M(v) = \frac{M(v)}{\max M} \quad (5)$$

The quality of transformation  $Q(v)$

$$Q(v) = 1 - \frac{|X'|}{|X|} \quad (6)$$

The importance of columns  $W(v)$

$$W(v) = \sum_{X \in v} W(X) \quad (7)$$

$$W(v) = \frac{W(v)}{\max W} \quad (8)$$

구현결과

```
Anaconda Prompt
{"order":59,"order2":1,"describe":"GROUP BY carrier, BIN arrdelay BY '
ZERO'", "x_name": "arrdelay", "y_name": "CNT(arrdelay)", "chart": "bar", "classify": ["AA",
"EY", "MQ"], "x_data": [{">0", "<=0"}], "y_data": [[1809, 1071], [60,
84], [41, 15], [207, 112], [640, 718]]}
full_time: 63.53793978681101

(keras_env) C:\Users\hbee.ysjo\minid\owl\Q3_deepeye>
```

```
*작업 애플릿 - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말
Data: FlightDelayStatistics, 3700건
Column number: 6column(date,carrier,destcity,depdelay,arrdelay,time)
Visualization Recommendation: 59 건
Running time: 63.5 s

Ln 1, Col 36 100% Windows (CRLF) UTF-8
```

## 연구 노트

### 연구 목표

데이터 시각화 추천 논문에서 제안하는 방법에 대하여 추가로 서술하고 DB와 연동하기 위한 코드를 구현한다.

### 개념

#### Partial Order-based Visualization Selection

- 1) Enumerate all visualizations
- 2) Decide the 'valid' charts
- 3) Conform to the partial order
- 4) Add a directed edge
- 5) Get a graph  $G(V, E)$ , where  $V$  is all valid visualization nodes and  $E$  indicates visualization pairs that satisfy partial orders

The weight between  $u$  and  $v$ , where  $u \geq v$ , is defined as:

$$\frac{M(u) - M(v) + Q(u) - Q(v) + W(u) - W(v)}{3} \quad (9)$$

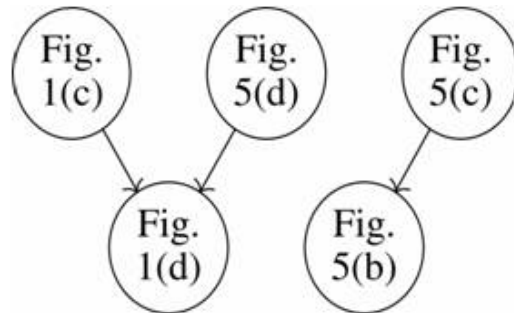
$M(v)$ : The matching quality

$Q(v)$ : The quality of transformation

$W(v)$ : The importance of columns

Examples about how to rank visualization nodes based on the graph.

Fig	1(c)	1(d)	5(b)	5(c)	5(d)
1(c)	$\succeq$	$\succ$	none	none	none
1(d)	none	$\succeq$	none	none	none
5(b)	none	none	$\succeq$	none	none
5(c)	none	none	$\succ$	$\succeq$	none
5(d)	none	$\succ$	none	none	$\succeq$



### Hybrid Ranking Method

HybridRank to linearly combine these two methods(Learning-to-rank and Partial order) as follows.

Consider a visualization  $v$ . Suppose its ranking position  $lv$  by learning-to-rank and its ranking position is  $pv$  by partial order

Then assigns  $v$  with a score of  $lv + \alpha pv$ , where  $\alpha$  is the preference weight of two methods which can be learned by some labelled data, and rank the visualizations by the score.

### 차트 추천 결과 확인

electricityConsumptionOfEasternChina.csv	2019-10-07 오후 7:30	Microsoft Office E...	348KB
FlightDelayStatistics2015.csv	2019-10-07 오후 7:30	Microsoft Office E...	124KB
FlightDelayStatistics2015_2.csv	2019-10-31 오후 7:26	Microsoft Office E...	1KB
FlightDelayStatist...	유형: Microsoft Office Excel 실효로 구분된 값 파일 크기: 119바이트 수정된 날짜: 2019-10-31 오후 7:26	오전 11:38 Microsoft Office E...	1KB
FlightDelayStatist...	2019-10-07 오후 7:30	Microsoft Office E...	124KB
Foreign Visitor Arrivals By Purpose(Jan-Dec 2015).csv	2019-10-07 오후 7:30	Microsoft Office E...	6KB
happinessRanking(2015-2016).csv	2019-10-07 오후 7:30	Microsoft Office E...	33KB
HollywoodsMostProfitableStories.csv	2019-10-07 오후 7:30	Microsoft Office E...	5KB
MostPopularBabyNames(NewYork).csv	2019-10-07 오후 7:30	Microsoft Office E...	892KB
titanicPassenger.csv	2019-10-07 오후 7:30	Microsoft Office E...	55KB

```

1 [{"order":1, "x_name":"depdelay", "y_name":"arrdelay", "chart":"scatter", "classify":""},
2 {"order":2, "x_name":"depdelay", "y_name":"time", "chart":"scatter", "classify":""},
3 {"order":3, "x_name":"arrdelay", "y_name":"time", "chart":"scatter", "classify":""},
4 {"order":4, "x_name":"date", "y_name":"CNT(date)", "chart":"line", "classify":"date"},
5 {"order":5, "x_name":"date", "y_name":"AVG(depdelay)", "chart":"line", "classify":"date"},
6 {"order":6, "x_name":"date", "y_name":"AVG(arrdelay)", "chart":"line", "classify":"date"},
7 {"order":7, "x_name":"date", "y_name":"SUM(time)", "chart":"line", "classify":"date"},
8 {"order":8, "x_name":"date", "y_name":"AVG(time)", "chart":"line", "classify":"date"},
9 {"order":9, "x_name":"date", "y_name":"AVG(depdelay)", "chart":"line", "classify":"date"},
10 {"order":10, "x_name":"date", "y_name":"AVG(arrdelay)", "chart":"line", "classify":"date"},
11 {"order":11, "x_name":"date", "y_name":"SUM(time)", "chart":"line", "classify":"date"},
12 {"order":12, "x_name":"date", "y_name":"AVG(time)", "chart":"line", "classify":"date"},
13 {"order":13, "x_name":"date", "y_name":"CNT(date)", "chart":"line", "classify":"date"},
14 {"order":14, "x_name":"date", "y_name":"CNT(date)", "chart":"bar", "classify":"date"},
15 {"order":15, "x_name":"date", "y_name":"AVG(depdelay)", "chart":"bar", "classify":"date"},
16 {"order":16, "x_name":"date", "y_name":"AVG(arrdelay)", "chart":"bar", "classify":"date"},
17 {"order":17, "x_name":"date", "y_name":"SUM(time)", "chart":"bar", "classify":"date"},
18 {"order":18, "x_name":"date", "y_name":"AVG(time)", "chart":"bar", "classify":"date"},
19 {"order":19, "x_name":"carrier", "y_name":"CNT(carrier)", "chart":"pie", "classify":"carrier"},
20 {"order":20, "x_name":"carrier", "y_name":"CNT(carrier)", "chart":"bar", "classify":"carrier"},

```

### MonetDB 연동 구현 결과

```

98 ##### data import function
99
100 def load_data(self):
101     # set up a connection, arguments below are the defaults
102     connection = pymonetdb.connect(username="monetdb", password="monetdb", hostname="192.168.0.210", database="bcsdb")
103
104     # create a cursor
105     cursor = connection.cursor()
106
107     # increase the rows fetched to increase performance (optional)
108     cursor.arraysize = 20000
109
110     # execute a query (return the number of rows to fetch)
111     cursor.execute('select fl_date, day_of_week, op_unique_carrier, carrier_grade, carrier_group, origin, dest, dep_delay, arr_delay, distance from bcs.BCS_20190828_090316_1170 sample 100')
112

```

# 연구 노트

## 연구 목표

Partial order-based visualization selection 방법과 Hybrid ranking method를 각각 실험하고 결과를 비교한다.

## 개념

### 1. Partial order-based visualization selection

실험 데이터: Flightdelay(10 column) / Sales(8 column) dataset



그림 1. flightdelay dataset

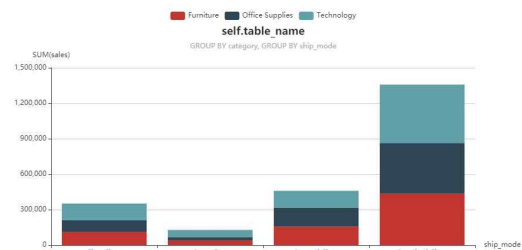


그림 2. sales dataset

실험결과:

- Learning to rank 방법과 동일한 숫자의 차트 추천(flightdelay 63건, sales 156건)
- 순서 및 추천 차트 수의 변화(\*sales dataset의 경우 temporal, categorical한 column 이 상대적으로 많기 때문에 더 많은 수의 차트 추천 결과가 나옴)

### 2. Hybrid Ranking Method

실험 데이터: Flightdelay(10 column) / Sales(8 column) dataset

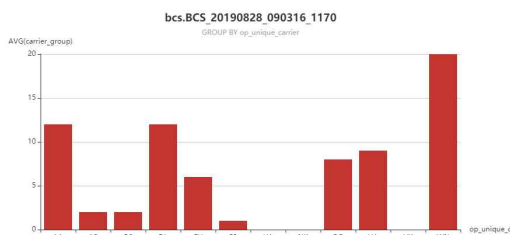


그림 3. flightdelay dataset

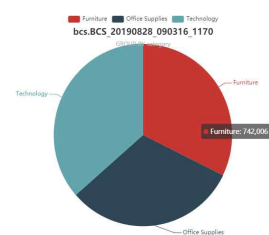


그림 4. sales dataset

실험결과:

- 기존 실험과 동일한 숫자의 차트 추천(flightdelay 63건, sales 156건)
- 순서 및 추천 차트 수의 변화(\*flightdelay 순서가 다름)



# 연구 노트

## 연구 목표

화면에 추천차트를 출력하기 위한 표준출력으로 나타내고, DB에서 해당 차트를 관리하기 위한 인터페이스를 설계한다. 또한 영문 한글 버전의 차트 추천 UI/UX를 고안한다.

## 개념

### 1. 표준출력

```
C:\Users\hbee.jsjo\mindflow\23_deepeye_20191118_flightdata_learning_rank/json/  
C:\Users\hbee.jsjo\mindflow\23_deepeye_20191118_flightdata_learning_rank/html/
```

그림 1. Json/html 표준출력

차트 추천을 위한 표준출력으로 Json파일과 html 파일을 표준출력으로 나타나도록 함. Json 파일에는 추천 차트에 대한 세부사항이 key와 value값을 가지도록 구성함. 즉, 추천 차트 종류, x축/y축 컬럼명, legend 속성 정보 등이 기록됨.

### 2. 차트 추천 인터페이스

파라미터명	사용가능한값 또는 예시	필수유무
rank_func	learning, partial, diversified	필수
db_host	192.168.0.210	필수
db_port	5000	필수
db_name	bcsdb	필수
db_user	monetdb	필수
db_passwd	monetdb	필수
db_query	SQL 쿼리문 전체	필수
ui_width	한차트당 넓이	필수
ui_height	한차트당 높이	필수

그림 2. 차트 추천 인터페이스

그림 2는 추천 차트 인터페이스로 DB에 기록하기 위한 파라미터 명을 나타내었음. rank\_func는 learning, patial, diversified에 따라 차트 추천을 위한 컬럼의 순서와 차트의 종류가 결정됨. 추천된 차트의 예시는 그림 3에 나타난 그림과 같음. 추천자(Recommender)와 컬럼 선택(Data column)에 따라 데이터를 분석하여 최적의 차트를 추천함.

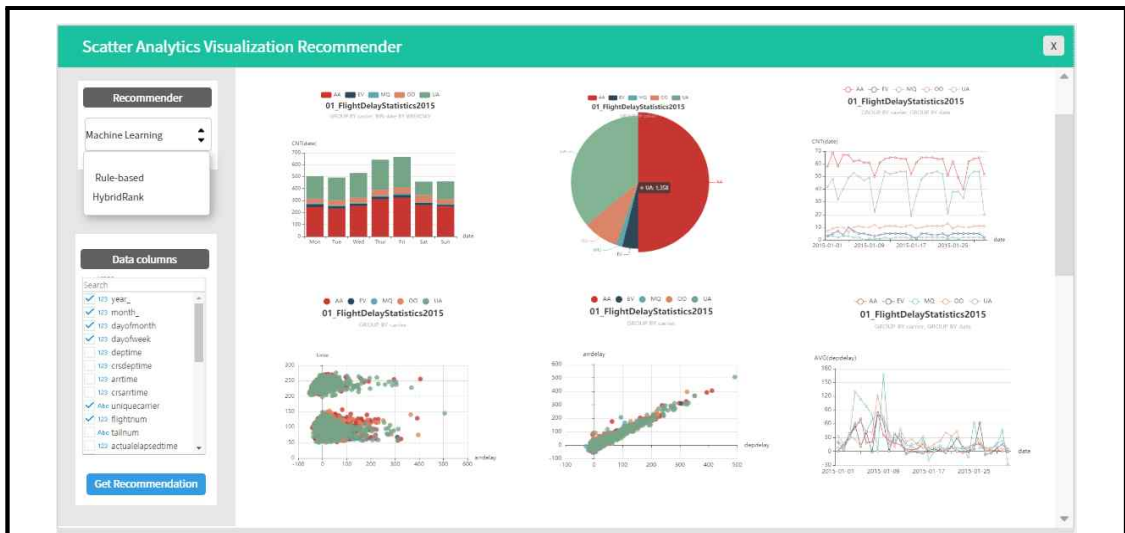


그림 3. 차트 추천 영문 UI/UX



그림 4. 차트 추천 한글 UI/UX

# 연구 노트

## 연구 목표

차트 추천 아이콘을 디자인하고, 차트 추천 결과를 도출하기 위한 최종 스크립트 전달 인자에 따라 출력결과를 확인한다.

## 개념

### 1. 차트 추천 아이콘 디자인



그림 1. 차트 추천 아이콘

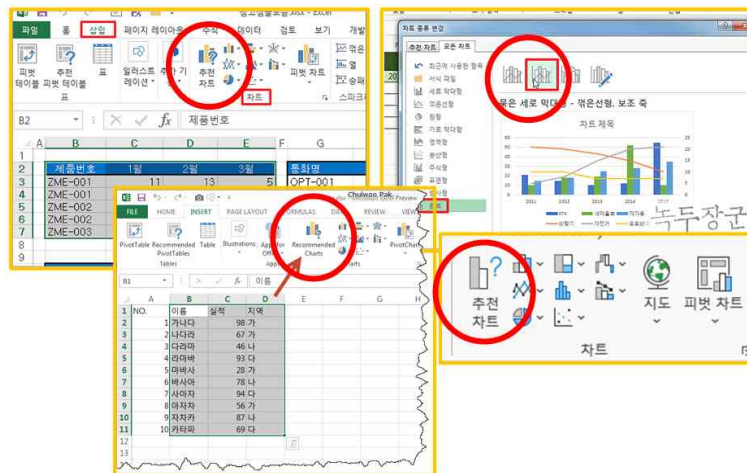


그림 2. 차트 추천 아이콘

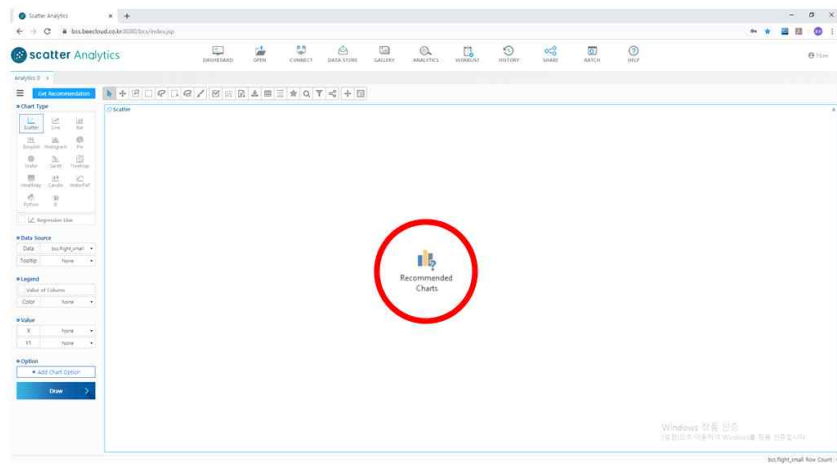


그림 3. 차트 추천 아이콘

```

Anaconda Prompt
(keras_env) C:\Users\hbee.jsjo\mindflow\23_deepeye_20191203_final>python recommender_script.py --rank_func="learning" --csv_path="C:/Users/hbee.jsjo/mindflow/23_deepeye_20191203_final/datasets/FlightDelayStatistics2015.csv" --json_path="C:/Users/hbee.jsjo/mindflow/23_deepeye_20191203_final/json/" --html_path="C:/Users/hbee.jsjo/mindflow/23_deepeye_20191203_final/html/"
C:/Users/hbee.jsjo/mindflow/23_deepeye_20191203_final/datasets/FlightDelayStatistics2015.csv
learning
C:\Python_core\display_HTML_object>
Score file not found. You may run the program again to get the result.
rank_func_time: 1.1970294094085693
html_time: 0.4089442538215088
json_time: 0.010935864888184062
C:/Users/hbee.jsjo/mindflow/23_deepeye_20191203_final/json/
C:/Users/hbee.jsjo/mindflow/23_deepeye_20191203_final/html/
  
```

그림 4. 차트 추천 스크립트 실행 결과

```

python recommender_script.py
--rank_func="learning"
--csv_path="C:/Users/hbee.jsjo/mindflow/23_deepeye_20191203_final/datasets/FlightDelayStatistics2015.csv"
--json_path="C:/Users/hbee.jsjo/mindflow/23_deepeye_20191203_final/json/test.json"
--html_path="C:/Users/hbee.jsjo/mindflow/23_deepeye_20191203_final/html/test.html"
  
```

그림 5. 차트 추천 스크립트 실행 명령어

차트를 추천 하는 기존방식을 참고하여 Scatter Analytics의 차트 추천 아이콘을 고안하였음. 고안된 아이콘은 그림 3과 같이 추천 아이콘을 중앙에 배치하여 사용자로 하여 특별한 지식 없이도 추천된 차트를 볼 수 있도록 설계하였음.