



Hacettepe Üniversitesi Endüstri Mühendisliği Tezli Yüksek Lisans Programı



EMÜ 660 ANALİTİK YÖNTEMLER İLE KARAR VERME

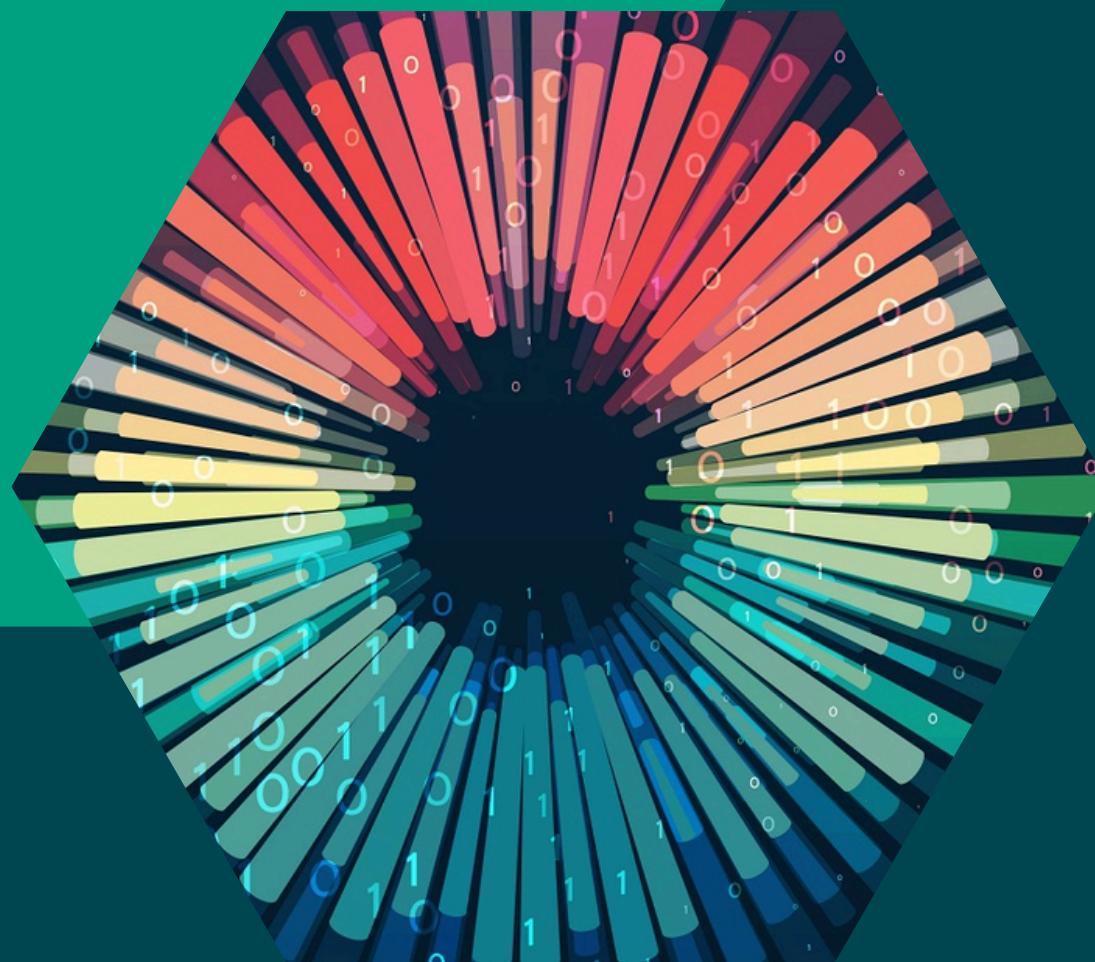
Makale Sunumları

Making data visualization more efficient
and effective: a survey

Tuncay Özkardeş - N24111741
İzzet Can Özbaş - N23139298



İÇİNDEKİLER



- Veri Görselleştirme Tarihi
- 1- Giriş
- 2- Veri Görselleştirme Spesifikasyonları
- 3- Veri Görselleştirme için Etkin ve Verimli Yaklaşımlar
- 4- Görselleştirme Öneri Sistemleri
- 5- Tartışmalar
- 6- Sonuç

Tarihçe

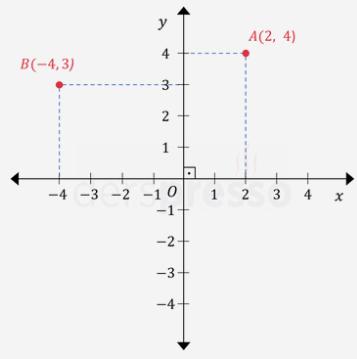


M.Ö. 5500

Mezopotamya ve Mısır'da ilk haritalar çizildi. İnsanlar nehir sistemlerini kaydetmek için resimler çizdi.

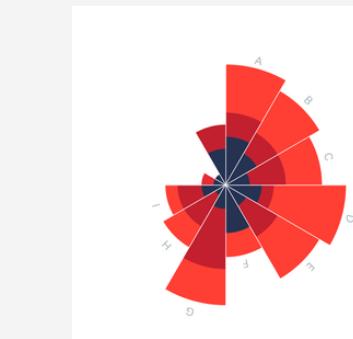
17. Yüzyıl

René Descartes kartezyen koordinat sistemini geliştirdi. William Playfair modern çubuk grafik, çizgi grafik ve pasta grafik gibi temel teknikleri icat etti.



19. Yüzyıl

Florence Nightingale sağlık verilerini görselleştirmek için "gül diyagramı" (polar area diagram) kullandı. Charles Minard Napolyon'un Rusya Seferi'ni anlatan ünlü haritasını çizdi .



20. Yüzyıl

- 1960'lар: Bilgisayarlar sayesinde interaktif veri görselleştirme başladı.
- 1970'lар: John Tukey "Exploratory Data Analysis" kavramını geliştirdi.
- 1980'lар: Edward Tufte, veri görselleştirmede tasarım prensiplerini içeren "The Visual Display of Quantitative Information" kitabı yayınladı.



21. Yüzyıl

- 2000'lар: Tableau, Power BI gibi araçlar interaktif görselleştirme sağladı.
- 2010'lар: D3.js gibi JavaScript kütüphaneleri ile web tabanlı dinamik görselleştirme yaygınlaştı.
- 2020'lар: Makine öğrenmesi ve yapay zeka destekli otomatik veri görselleştirme teknikleri gelişti.



1- Giriş

- Veri görselleştirme, soyut verileri fiziksel ve görsel unsurlara (uzunluk, konum, şekil, renk vb.) dönüştürerek verileri anlamayı kolaylaştıran güçlü bir yöntemdir.
- Veri görselleştirme, Excel, Google Sheets, Oracle Data Visualization Desktop, IBM DB2, Amazon Quicksight ve Microsoft Power BI gibi birçok veri tabanlı uygulamada yaygın olarak kullanılmaktadır.

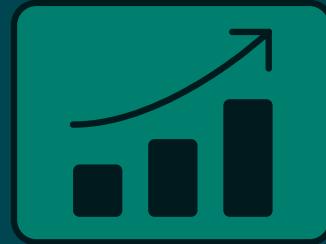
Katkı Sahipleri:

Bilgisayar grafikleri topluluğu	→ D3,	Web tarayıcılarında veri görselleştirme için kullanılan güçlü bir JavaScript kütüphanesidir.
Görselleştirme topluluğu	→ D3, Vega-Lite, VizQL	
Veri tabanı topluluğu	→ Hyper DB	

Veri Görselleştirme Süreci:



Veri görselleştirme sürecini daha etkili ve verimli hale getirmek için üç ana yön belirlenmiştir:



Görselleştirme Spesifikasyonları

Kullanıcıların istedikleri görselleri oluşturmasına yardımcı olan diller ve araçlar geliştirilmiştir. "Mapping" (eşleme) bileşenine odaklanarak verileri görsel elementlere nasıl çevireceğimizi belirler. Aynı zamanda "Veri Manipülasyonu" bileşeniyle de kesişir (örneğin, graplama ve sıralama işlemleri).

Veri tabanı dilleri (SQL gibi) ile görselleştirme dilleri arasındaki tasarım farklarını ele alır.



Veri Görselleştirme için Verimli Yöntemler

Sürecin ölçülebilir ve hızlı olması önemlidir, özellikle "Veri Manipülasyonu" ve "Mapping" bileşenleri için. SQL gibi güçlü veri işleme motorları ile entegrasyon sağlanarak sorgular daha verimli hale getirilmiştir. Tableau için HyperDB gibi özelleştirilmiş sistemler geliştirilmiştir. Büyük veri ile çalışırken yaklaşık (approximate) ve aşamalı (progressive) çözümler kullanılarak gerçek zamanlı yanıt alma hedeflenmiştir.

Veri Görselleştirme Tavsiyeleri

Görselleştirme süreci deneme-yanılma yöntemine dayandığından, kullanıcılarla akıllı öneriler sunulması önemlidir. Bazı sistemler belirsiz kullanıcı girdilerini tamamlayarak otomatik öneriler sağlar. Görselleştirme ve veri tabanı toplulukları, kullanıcıların daha anlamlı görseller oluşturmasına yardımcı olacak yöntemler geliştirmiştir.

2- Veri Görselleştirme Spesifikasyonları

Veri görselleştirme dilleri genellikle üç temel bileşenden oluşur:

Veri:

- **Kayıtlar (Records):** Görselleştirilecek veriler.
- **Dönüştürme (Transformation):** Veriler üzerinde yapılan işlemler (gruplama, filtreleme, sıralama vb.).

İşaretler (Marks) veya Görsel İpuçları:

- **Tür (Type):** Verinin görsel olarak nasıl temsil edileceği (çubuk, çizgi, nokta vb.).
- **Boyut (Size):** Görselleştirmenin genişliği ve yüksekliği.
- **Gösterge (Legend):** Görselleştirmenin açıklama bilgileri.
- **Diğer Özellikler:** Çubukların genişliği, rengi gibi ek nitelikler.

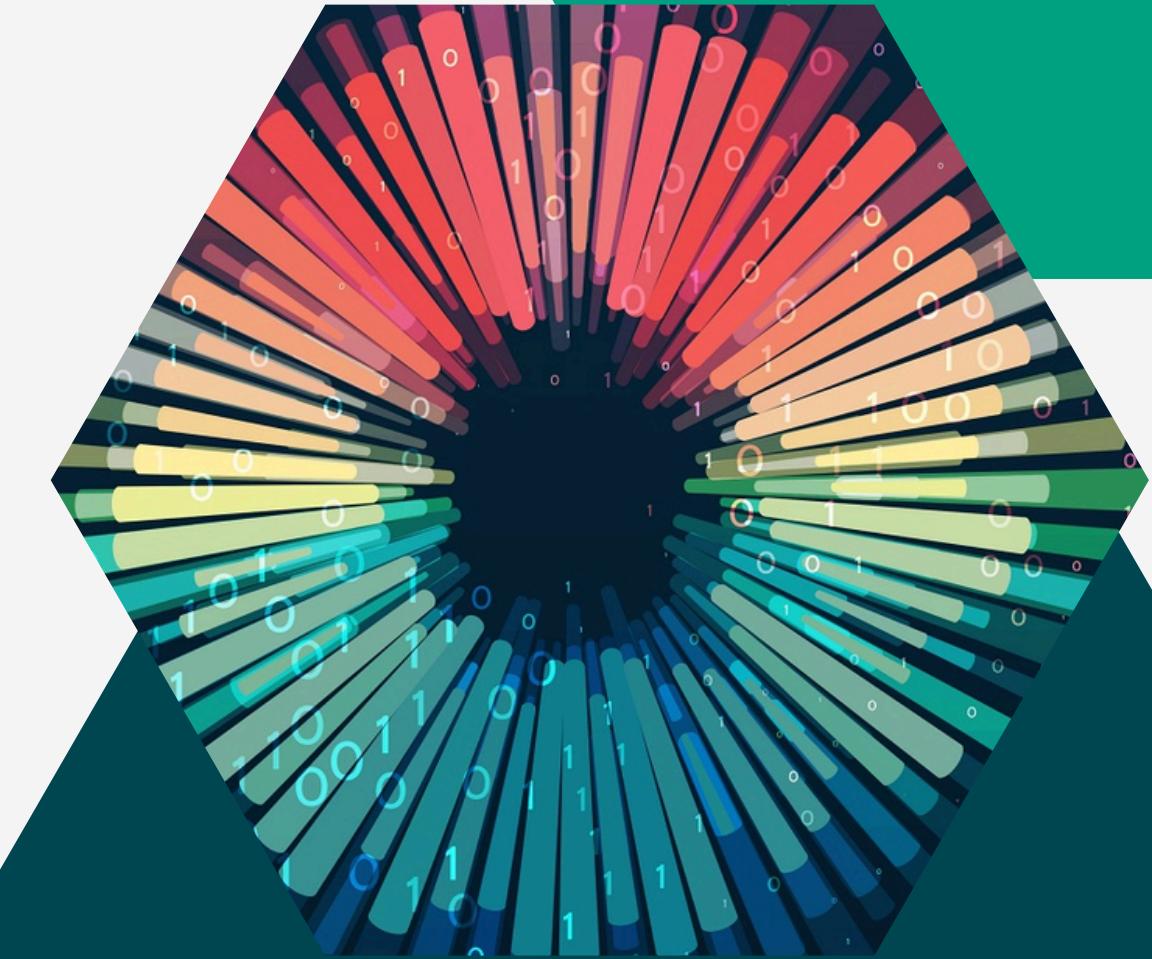
Eşleme (Mapping):

Verinin ilgili görsel bileşenlere nasıl dönüştürüleceğini belirler.



R - ggplot'daki AES() fonksiyonu bu görevi yapar.

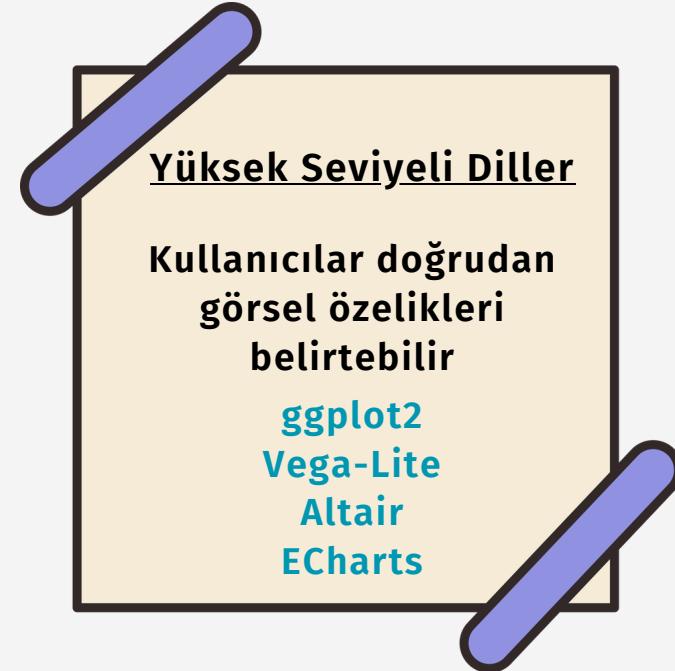
Veri görselleştirme dillerinin sınıflandırılması genellikle ifade gücüne (expressiveness) dayalı olarak yapılır. Düşük seviyeli diller daha fazla ifade gücüne sahipken, yüksek seviyeli diller bazı düşük seviyeli detayları kapsayarak (örneğin, varsayılan ayarlarla veya belirli kısıtlamalar ekleyerek) kullanımı kolaylaştırır. Aynı zamanda, yüksek seviyeli dillerin erişilebilirliği daha yüksektir ve kullanımları daha kolaydır.



2- Veri Görselleştirme Spesifikasyonları

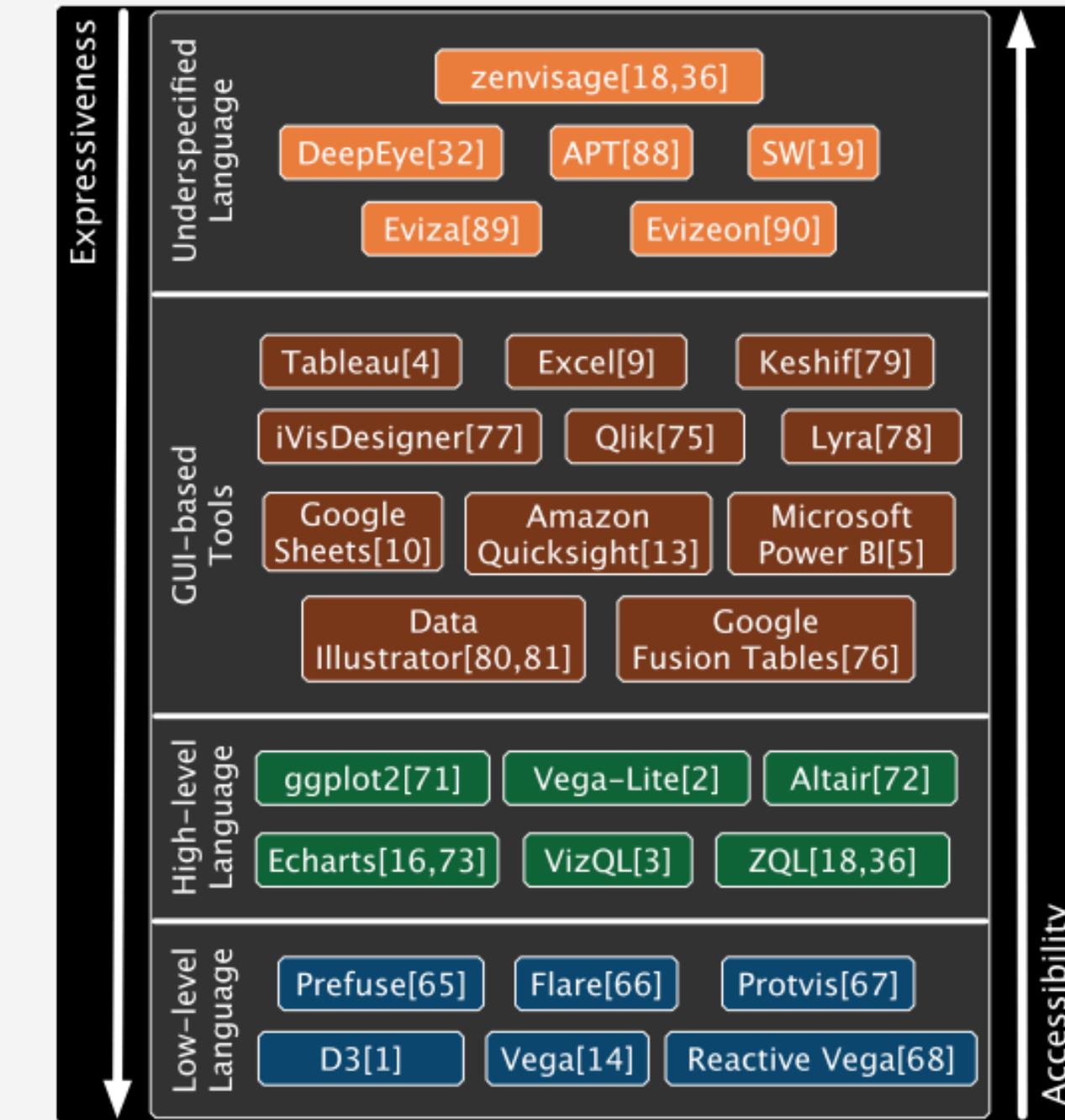


2.2) Veri Görselleştirme Dillerinin Sınıflandırılması



2.3) GUI Tabanlı Görselleştirme Araçları

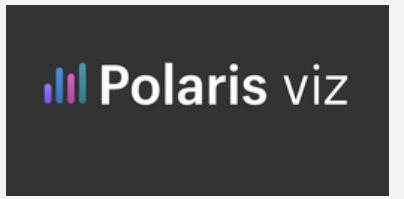
Kullanıcıların arayüzde doğrudan veri görselleştirmesi oluşturmasına olanak tanıyan grafiksel kullanıcı arayüzleri (GUI) kullanmayı ifade eder.



2- Veri Görselleştirme Spesifikasyonları

Etkileşimli Veri
Görselleştirme

Adım Adım Sorgu İyileştirme
(Stepwise Query Refinement)



Kullanıcılar, birden fazla
özellik satır, sütun veya
katmanlara sürükleyerek
görselleştirme oluşturabilir.

Fasetlenmiş Gezinme (Faceted Navigation)

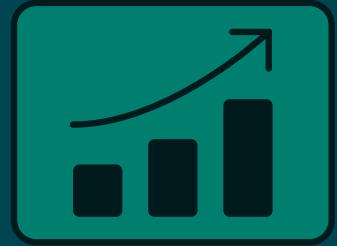


Kullanıcının yazdığı anahtar
kelimelere göre ilgili
görselleştirmeler önermektedir.
Facet (Grafik türü, Gruplama
seçenekleri vs.) seçilir.

Temel Mantık: Veri analizinin bir keşif süreci olmasıdır. Kullanıcılar, görselleştirmelerini adım adım
geliştirerek (ekleme/çıkarma/değiştirme gibi işlemler yaparak) istedikleri sonuçlara ulaşırlar.

GIU tabanlı dillerde grafik türleri sınırlıdır, şablon dışına çıkmak zor olabilir. Detayları özelleştirmek zordur (çubuk genişliği gibi)

2- Veri Görselleştirme Spesifikasyonları



2.4) Eksik Belirtilmiş Tanımlamalar

Çoğu durumda kullanıcılar verinin tüm yönlerini tam olarak bilemez. Bunun nedenleri şunlar olabilir:

- Veri çok büyük olabilir
- Veri sürekli güncellenebilir

Bu sebeple eksik belirtilmiş tanımlamaları (underspecified specifications) destekleyen sistemlere ihtiyaç vardır. 3 tür ipucuna bakalım:

1) Referansa Dayalı (Reference-Based) İpucu

Kullanıcı, bir örnek görselleştirme (seed) sağlar ve sistem buna benzer veya farklı görselleştirmeler önerir.



2) Anahtar Kelimeye Dayalı (Keyword-Based) İpucu

Google tarzı bir yaklaşımla kullanıcılar belirli bir sütun veya veri ilişkisini görmek istediklerini belirtebilir.



3) Doğal Dile Dayalı (Natural Language-Based) İpucu

Kullanıcının girdilerini ve sistemin mevcut durumunu bağlam içinde değerlendirir. Örneğin, Evizeon'da kullanıcı önce "İngiltere'de kızamık vakalarındaki ani artışı göster" yazabilir ve sistem bu eğilimi gösterir. Ardından kullanıcı "oradaki kabakulak vakalarını göster" dediğinde, sistem aynı bölgedeki kabakulak vakalarını gösterir.

Eviza ve Evizeon,

3- Veri Görselleştirme için Etkin ve Verimli Yaklaşımlar

3.1) Kesin (Exact) Veri Görselleştirme

Kesin veri görselleştirme, verileri hızlı ve doğru bir şekilde işlemenin yollarını ele alır. Veri görselleştirme sistemleri genellikle SQL sorgularıyla verileri işler ve ardından görselleştirme araçlarını kullanarak sonuçları oluşturur.

1) Soru Çevirimi (Query Translation):

Veri tabanı yönetim sistemleri (DBMS) ile uyumlu hale getirmek için görselleştirme sorgularının SQL ifadelerine çevrilmesi yaygın bir yaklaşımdır.

2) Veri Görselleştirme Sistemlerinin DBMS ile Entegrasyonu:

SQL sorgularını doğrudan kullanarak görselleştirme yapmak doğaldır, ancak bazı dezavantajları bulunmaktadır:

- İşlevlerin tekrar edilmesi: Veritabanı ve istemci tarafında aynı işlemler yapılabilir (örneğin, filtreleme hem SQL sorgusuyla hem de JavaScript ile yapılabilir). Bu durum, hangi yöntemin daha optimize olduğu konusunda kararsızlığa neden olabilir.
- Bağımsız sistemlerin yönetim zorluğu: Sunucu (veritabanı) ve istemci (görselleştirme aracı) farklı varsayımlar ve optimizasyon teknikleriyle çalıştığından, sistemin geliştirilmesi ve ölçeklenmesi zorlaşır.

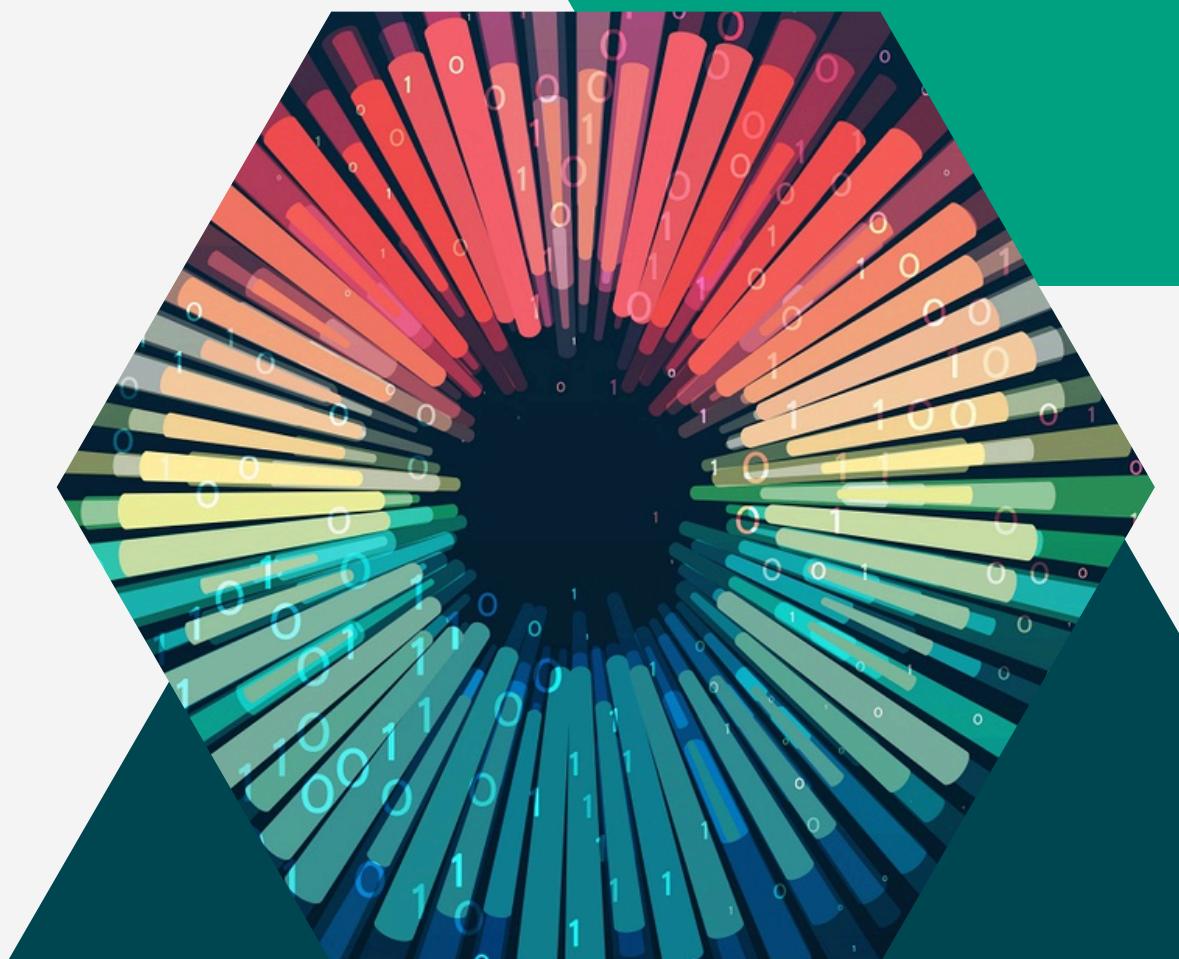
Çözüm: Veri Getirme (Retrieval) ve Görselleştirme İşlemlerinin Birleştirilmesi

Bu sorunu çözmek için veri alma ve görselleştirme işlemlerinin sıkı bir şekilde entegre edilmesi önerilmektedir.

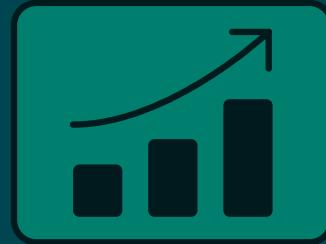
Ermac, bu yönde geliştirilmiş bir Veri Görselleştirme Yönetim Sistemi (DVMS - Data Visualization Management System) olarak öne çıkmaktadır.

Ermac'ın Temel Özellikleri:

- Veri (Data):** Görselleştirilecek veri kayıtları ve bunların görsel öğelerle ilişkisi.
- Ölçekler (Scales):** Veri aralıklarının görsel kodlama aralıklarına dönüştürülmesi.



3- Veri Görselleştirme için Etkin ve Verimli Yaklaşımlar



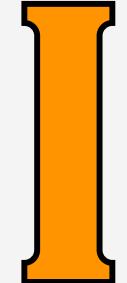
3) Sütun Tabanlı Depolama (Column Stores)

Sütun tabanlı depolama, veri görselleştirme işlemlerinde daha hızlı sorgulama ve veri işleme sağladığı için etkileşimli analiz ve OLAP (Online Analytical Processing) sistemlerinde daha verimli bir alternatif olarak öne çıkmaktadır.

4) İndeksleme

Tablolarda taranan kayıt sayısını azaltarak sorguların daha hızlı çalışmasını sağlar

- **Ağaç Tabanlı (Tree-Based) İndeksleme:** Hiyerarşik bir ağaç yapısı kullanarak, kullanıcı seçimlerini ve filtreleme işlemlerini hızlandırır.
- **Önceden Hesaplanan (Precomputed) Veri Küpleri:** Farklı veri dilimlerine ait toplama işlemlerini (aggregation) önceden hesaplayarak ham veriye erişmek yerine doğrudan bu özet verilerden sorgulama yapar.
- **Etkileşimli Görselleştirme için İndeksleme (Falcon):** Kullanıcının etkileşimde olduğu (aktif) görselleştirme için indeks oluşturur ve pasif görselleri anlık olarak vurgular. Diğer indeksleme sistemlerine göre daha küçük bellek kullanımı gerektirir.



5) Paralel Hesaplama (Parallel Computation)

- Büyük veri kümeleriyle çalışırken, işlemlerin paralel olarak gerçekleştirilmesi, görselleştirme sistemlerinin daha verimli ve hızlı çalışmasını sağlar.
- Örneğin SeeDB, Birden fazla SQL sorgusunu paralel olarak yürütür.



3- Veri Görselleştirme için Etkin ve Verimli Yaklaşımlar

6) Tahmin ve Önbellekleme (Prediction and Prefetching)

önceki görselleştirmelerinden ilham alarak yeni görselleştirmeler oluşturur. Kullanıcılar genellikle parametreleri değiştirerek veya yakınlaştırma/uzaklaştırma yaparak veri detaylarını keşfeder.

- Anlık Olarak Keşfedilen Görselleştirmeler (Currently Explored Visualizations): kullanıcının mevcut keşfettiği görselleştirmelere dayalı olarak veri önbelleklemesi yapar.
- Geçmiş Verilere Dayalı Önbellekleme (Historical Data Prefetching): Bu yaklaşım, kullanıcıların geçmiş hareketlerini analiz ederek bir sonraki adımda hangi veriye ihtiyaç duyulacağını tahmin eder.



XmdvTool

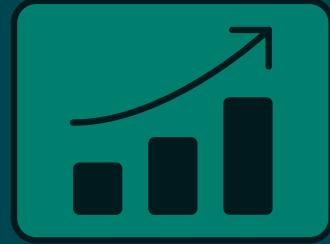


ForeCache

Makine öğrenmesini kullanır.

Problem	Technique	Target
Exact Data Visualization	Query Translation [17,20–23] Integrating Visualization Systems with DBMS [91,92] Column Stores [22,23,29,85] Indexes [93–95] Parallel Computation [22,23,31,96] Prediction and Prefetching [19,31,34,97,98]	Accelerate Visualization Exploration Process
Approximate Data Visualization	AQP [24,25] Incremental Sampling [26–28] Human Perception [26,99]	Enable Quick Visualization Creation
Progressive Data Visualization	Hierarchical Aggregation [31,93,100,101]	Enable Progressive Visualization Creation

3- Veri Görselleştirme için Etkin ve Verimli Yaklaşımlar



3.2) Yaklaşık Veri Görselleştirme (Approximate Data Visualization)

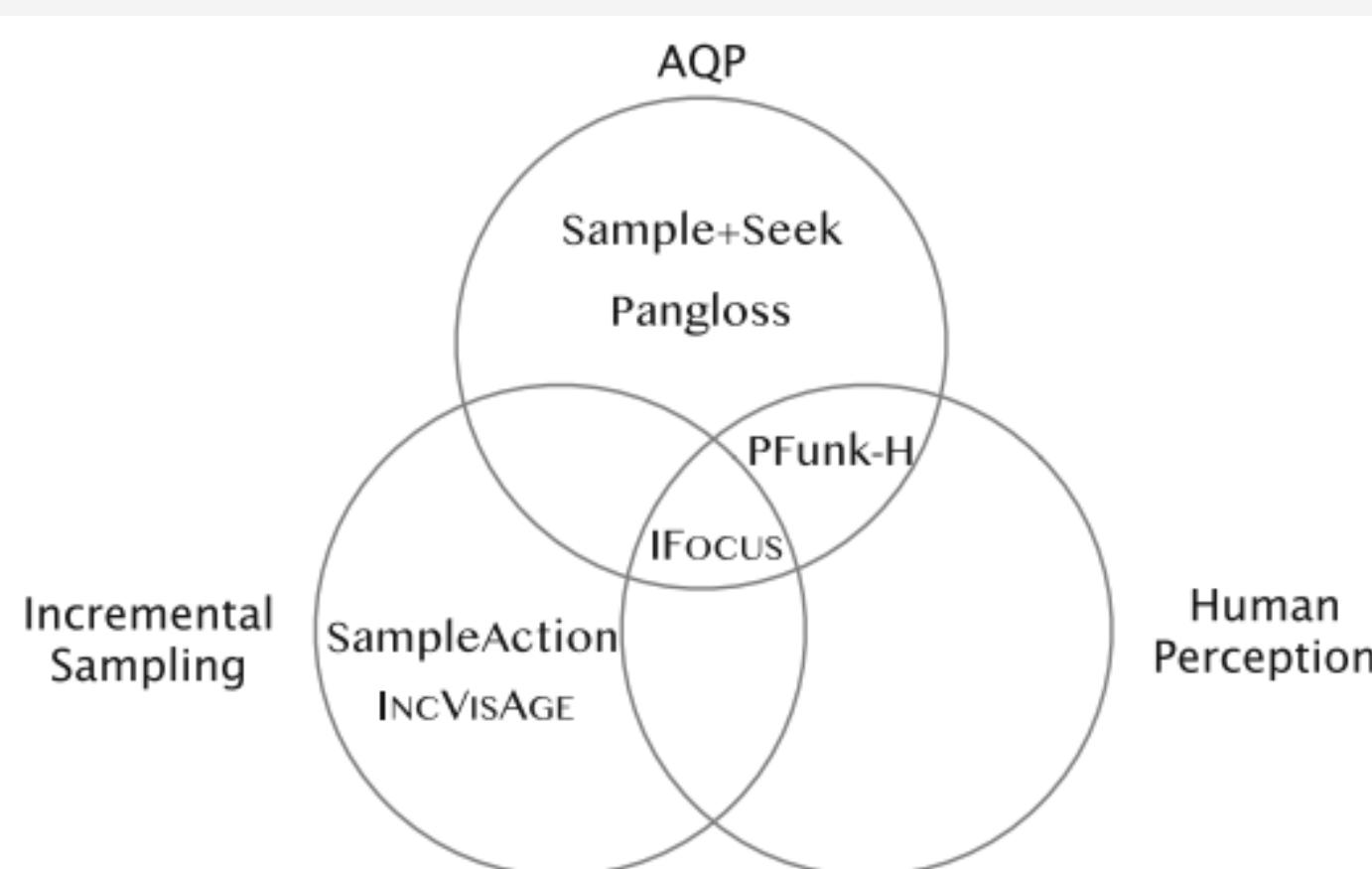
Büyük veri hacimleri hızla arttığında, geleneksel veri işleme yöntemleri hızlı ve etkileşimli görselleştirme sağlamakta yetersiz kalır. Bu problemi çözmek için yaklaşık sorgu işleme (AQP - Approximate Query Processing) teknikleri kullanılarak veri işleme süreci hızlandırılır.

Yaklaşık veri görselleştirme üç temel yaklaşıma ayrılmaktadır:

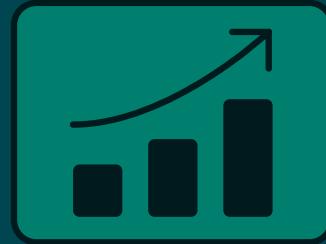
AQP Tabanlı Yöntemler: Doğrudan AQP tekniklerini kullanarak görselleştirme yapar.

Artan Örnekleme Tabanlı Yöntemler: Adım adım örnekleme (incremental sampling) ile sorgu işleme sürecini görselleştirme ile bağdaştırır.

İnsan Algısı Tabanlı Yöntemler: İnsan algısının sınırlamalarını dikkate alarak veri gösterimini optimize eder.



3- Veri Görselleştirme için Etkin ve Verimli Yaklaşımlar



3.2) Yaklaşık Veri Görselleştirme (Approximate Data Visualization)

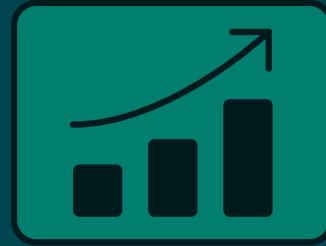
AQP tabanlı sistemler, belirli hata paylarıyla yaklaşık sonuçlar üretirken performans avantajı sunar

Sample+Seek:

- Dağılım hassasiyeti (distribution precision): Yaklaşık ve kesin görselleştirme arasındaki mesafeyi ölçerek doğruluk seviyesi belirler.
- Uniform Sampling: COUNT türündeki sorguları hızlandırır.
- Measure-Biased Sampling: SUM işlemleri için geliştirilmiş, verileri toplam değerlerine göre ağırlıklı seçen bir tekniktir.
- Measure-Augmented Inverted Index: Kategorik boyutları indeksleyerek toplama işlemlerini hızlandırır.

Bunun gibi sistemler, kullanıcılar hem yaklaşık hem de kesin görselleştirme seçenekleri sunarak keşif sürecini hızlandırır.

3- Veri Görselleştirme için Etkin ve Verimli Yaklaşımlar



3.2) Yaklaşık Veri Görselleştirme (Approximate Data Visualization)

Artan Örnekleme Tabanlı Yaklaşım (Incremental Sampling-based Approach), veri sorgulama tekniklerini veri görselleştirme süreciyle birleştirerek hızlı ve etkileşimli görselleştirmeler oluşturmayı amaçlar.

- Sistem başlangıçta küçük bir veri örneğine dayanarak yaklaşık bir görselleştirme oluşturur.
- Örneklem boyutu zamanla artırılarak görselleştirme kalitesi kademeli olarak iyileştirilir.

SampleAction

- Büyük veri kümeleri üzerinde toplama sorgularını (aggregation queries) görselleştirmek için geliştirilmiştir.
- Belirli bir hata payı ile (error bounds) çubuk grafikler (bar charts) oluşturur.
- Her saniye örneklem boyutunu artırarak hata payını azaltır ve görselleştirmeyi kademeli olarak iyileştirir.

3- Veri Görselleştirme için Etkin ve Verimli Yaklaşımlar



3.2) Yaklaşık Veri Görselleştirme (Approximate Data Visualization)

İnsan Algısına Dayalı Yaklaşık Veri Görselleştirme (Human Perception-Based Approximate Visualization)

Bazı durumlarda, örneklem büyülüğünü artırmak görselleştirme kalitesini her zaman iyileştirmez. Bunun iki temel nedeni vardır:

- Ekran piksel sayısı sınırlıdır, dolayısıyla daha fazla veri eklemek görsel olarak fark yaratmaz.
- İnsan algısı, küçük ayrıntıları algılama konusunda sınırlamalara sahiptir.

Bu nedenle, [insan algısına dayalı görselleştirme yaklaşımları](#), görselleştirmenin kalitesini algılanabilir fark yaratmayacak şekilde optimize eder. Örnekleme işlemi, kullanıcıların fark edemeyeceği bir noktaya ulaştığında durdurulur.

IFocus

- Çevrimiçi bir örnekleme algoritmasıdır ve çubuk grafiklerde (bar chart) çubuk sıralamalarının doğruluğunu koruyarak yaklaşık görselleştirme üretir.
 - Nasıl çalışır? Tüm gruplar başlangıçta "aktif" olarak kabul edilir ve her grup için örnekleme yapılır.
 - Her çubuk için bir güven aralığı hesaplanır.
 - Eğer bir çubuğun güven aralığı diğer çubuklarla örtüşmüyorsa, sıralaması kesinleşmiş demektir ve "aktif" olmaktan çıkar.
 - Tüm çubukların sıralaması kesinleştiğinde algoritma durur ve yaklaşık görselleştirme sunulur.
 - **Avantajı:** Görselleştirmede algılanabilir bir hata yaratmadan örneklem büyülüğünü optimize eder.
 - Çubuk sıralamalarını koruyarak insan algısına en uygun sonucu üretir.

3- Veri Görselleştirme için Etkin ve Verimli Yaklaşımlar

3.3) Aşamalı Veri Görselleştirme (Progressive Data Visualization)

Veri, farklı seviyelerde özetlenerek bir hiyerarşik yapı oluşturulur. Örneğin:

- Farklı boyutlardaki veri kümeleri (bins)
- Farklı zaman dilimleri (temporal values)
- Farklı coğrafi bölgeler (spatial zones)

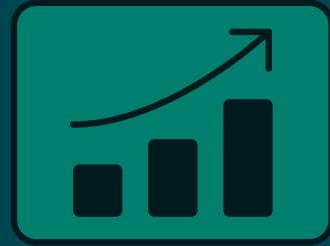
Bu hiyerarşik yapılar, kullanıcıların aşamalı veri keşfini desteklemek için kullanılır.

1. Aralığa Dayalı Kümeleme (Range-Based Binning) – imMens

- Farklı çözünürlüklerde görselleştirmeler sunmak için kutu boyutlarını (bin sizes) değiştirerek çalışır.
- Çok boyutlu verileri, "veri küpleri" (data cubes) olarak böler ve bu küpleri farklı seviyelerde "karolar" (tiles) halinde organize eder.
- Kullanıcılar yakınlaştırma (zoom-in) veya uzaklaştırma (zoom-out) yaparak farklı çözünürlüklerde veri keşfi yapabilir.

2. Aralık ve İçeriğe Dayalı Kümeleme (Range and Content-Based Binning) – HETree

- HETree-R (Range-Based HETree) ve HETree-C (Content-Based HETree) olmak üzere iki farklı ağaç yapısı sunar.
- HETree-R, imMens'e benzer şekilde çalışır ve eşit genişlikte veri aralıkları oluşturur.
- HETree-C, tüm yaprak düğümlerinde (leaf nodes) eşit sayıda veri noktası bulundurur.
- Örneğin, sınav notları analizinde öğretmenler için daha uygundur.
- Öğrencileri eşit grplara ayırarak sıralama yapabilir (örneğin, en iyi %10, %10-20, %20-30 gibi).



4- Görselleştirme Öneri Sistemleri

Uygulamadaki sorunlardan biri de birkaç değişiklik yapılmak istense bile her adımda yer alma zorunluluğudur.

Bunu aşmak için öneri sistemleri kullanılabilir.



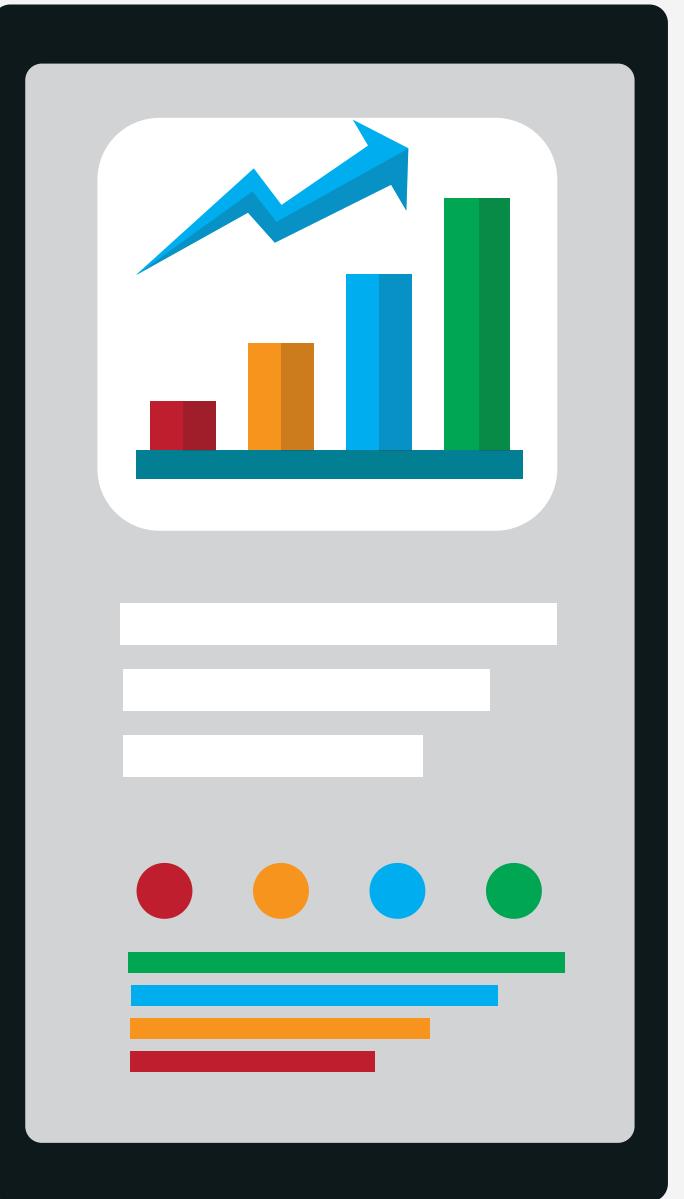
[Gündem Sayfasına Geri Dön](#)



4- Görselleştirme Öneri Sistemleri

Bazı metodları elemek için geçmiş tecrübelерden faydalabilir.

- **Kullanıcı tarafından belirlenen kısıtlamalar** - kolon, veri kayıtları gibi
- **Uzman tarafından belirlenen kısıtlamalar** - örneğin pasta grafiğinde “boy” göstermemek



4- Görselleştirme Öneri Sistemleri



Yöntem	Bölme Kriteri	Yaş Aralığı	Nokta Sayısı
HETree-R	Eşit yaş aralıkları	Sabit	Değişken
HETree-C	Eşit nokta sayısı	Değişken	Sabit

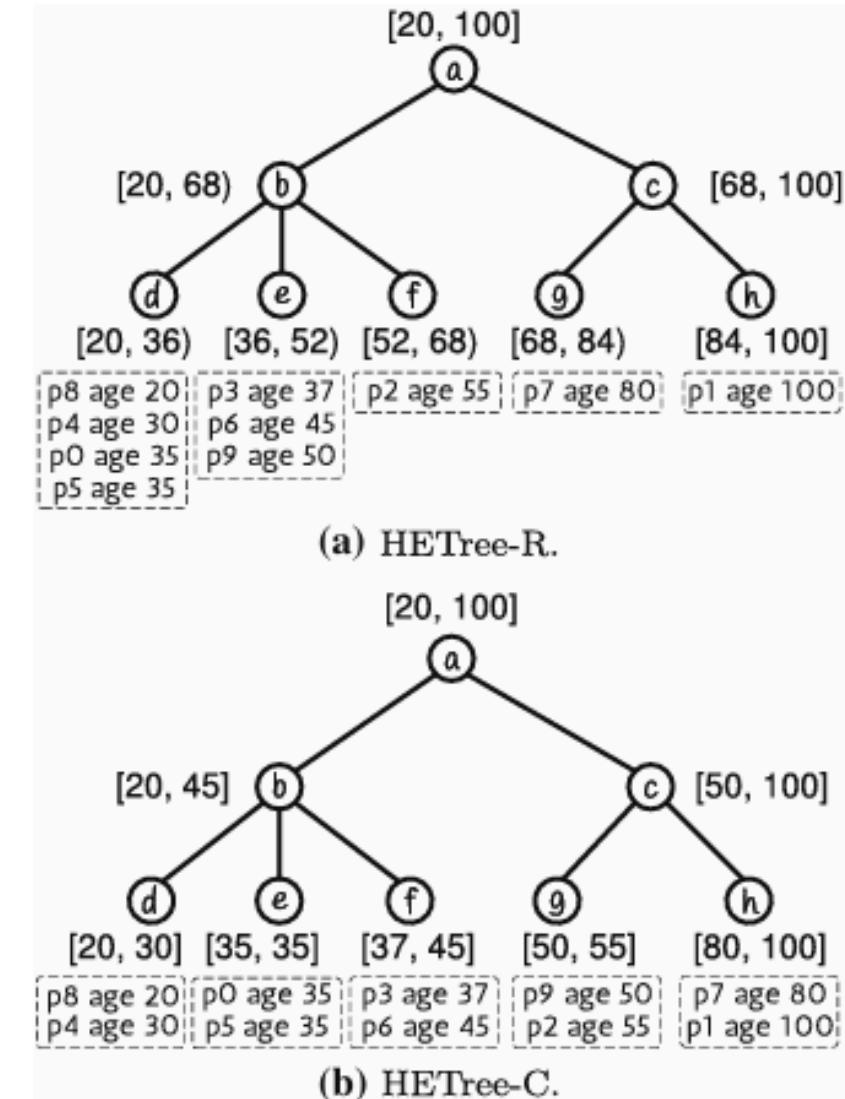
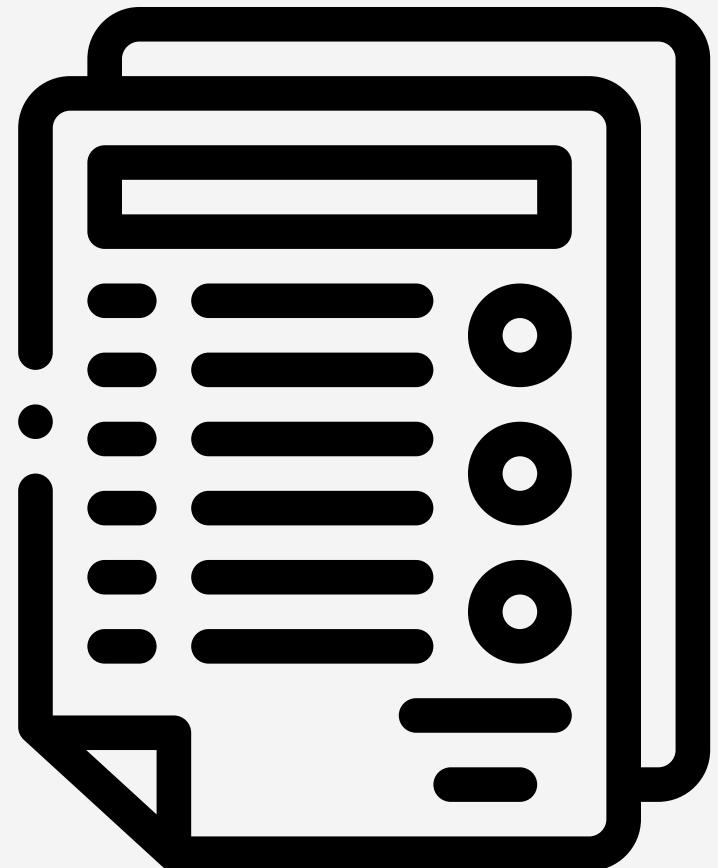


Fig. 9 A binning example of HETree-R and HETree-C. Ten points (p0–p9) are binned by attribute age. HETree-R bins data by equal age ranges: each leaf is a bin with size 16, and HETree-C bins data by equal number of points: each leaf is a bin with 2 points

4-1 Spesifikasyon Tabanlı Öneriler

- Tamamlanmamış spesifikasyon
- Referansa dayalı spesifikasyon



4-1-1 Tamamlanmamış Spesifikasyon

- Kullanıcı, görselleştirme öğelerinin tamamını veya bir kısmını belirtmeden öneri almak isteyebilir.
- Örneğin APT sistemi, kullanıcının veri görüntüleme hedeflerini belirlemesini ister.
- Voyager, ilk önce kullanıcının ilgilendiği bir kolonu seçmesini ister.

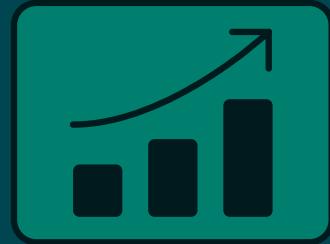


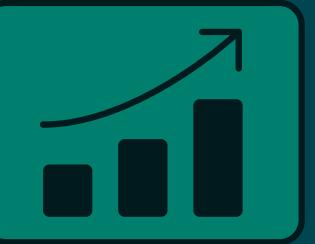
Table 6 Ranking rules in Voyager

$T(X)$	$T(Y)$	Mark Type
Categorical	Categorical	point > text
Categorical	Numerical	bar > point > text
Temporal	Numerical	line > bar > point > text
Numerical	Numerical	point > text

$T(X)$ and $T(Y)$ denote the data type of X- and Y- axes, respectively, and *Mark Type* denotes the permitted ranked mark types for this data type correspondingly

4-1-1 Kural Bazlı Sıralama

- İnsan algısına dayalı ölçütlerle çalışır.
- Bu sistemler; verinin türü, istatistiksel bilgileri ve görsel tercihlere göre aday görselleştirmeleri belirli kurallar çerçevesinde sıralar.



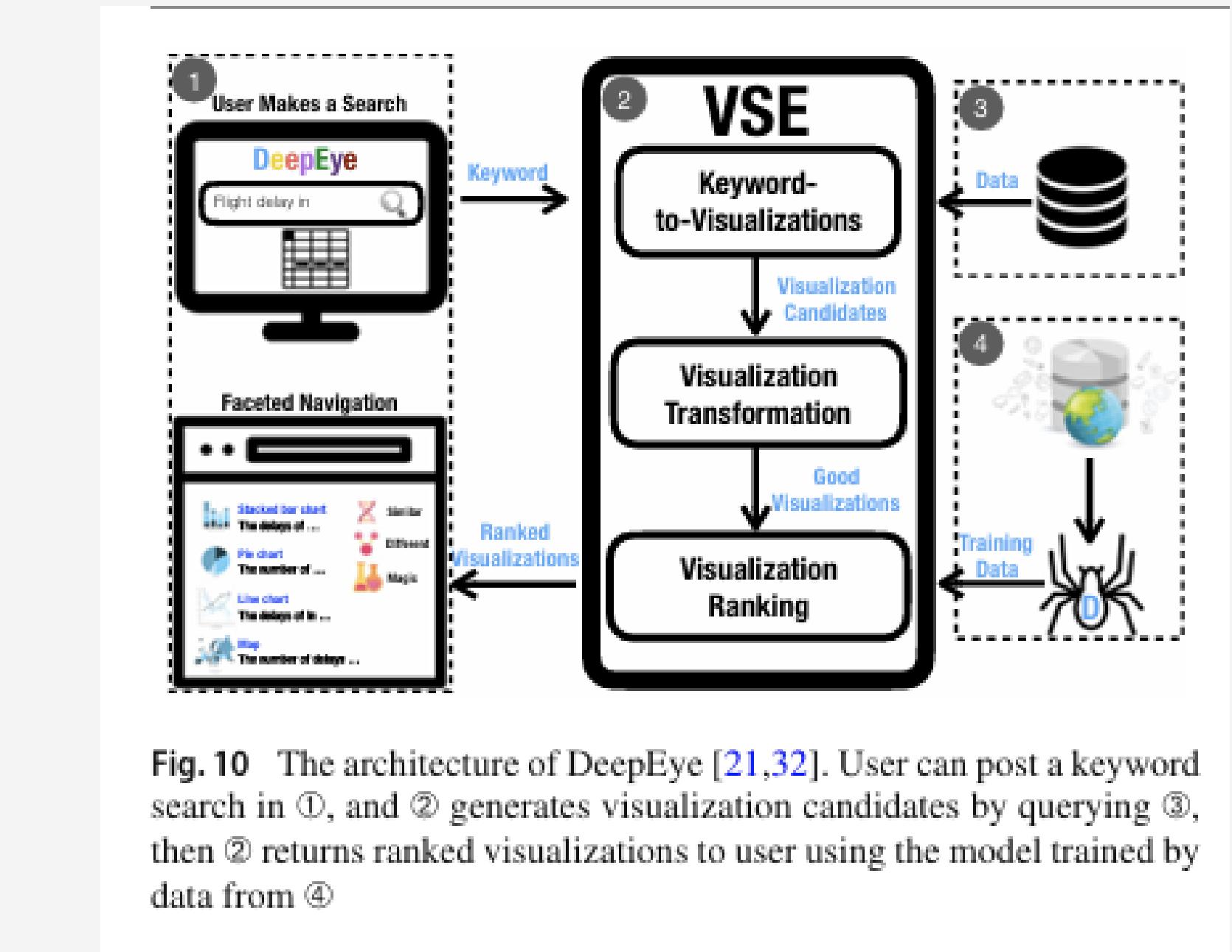
4-1-1 Kural Bazlı Sıralama

- **İstatistiksel Kurallar:** histogramlar, kutu grafikleri ve dağılım grafikleri
- **Algısal Kurallar:** Veri türü, kardinalite, dönüşüm kalitesi gibi faktörleri dikkate alarak görselleştirmeleri sıralar. Mesela Voyager ağırlıklı bir skor sistemi kullanırken, DeepEye kısmi sıralama yöntemiyle daha detaylı değerlendirme yapar.



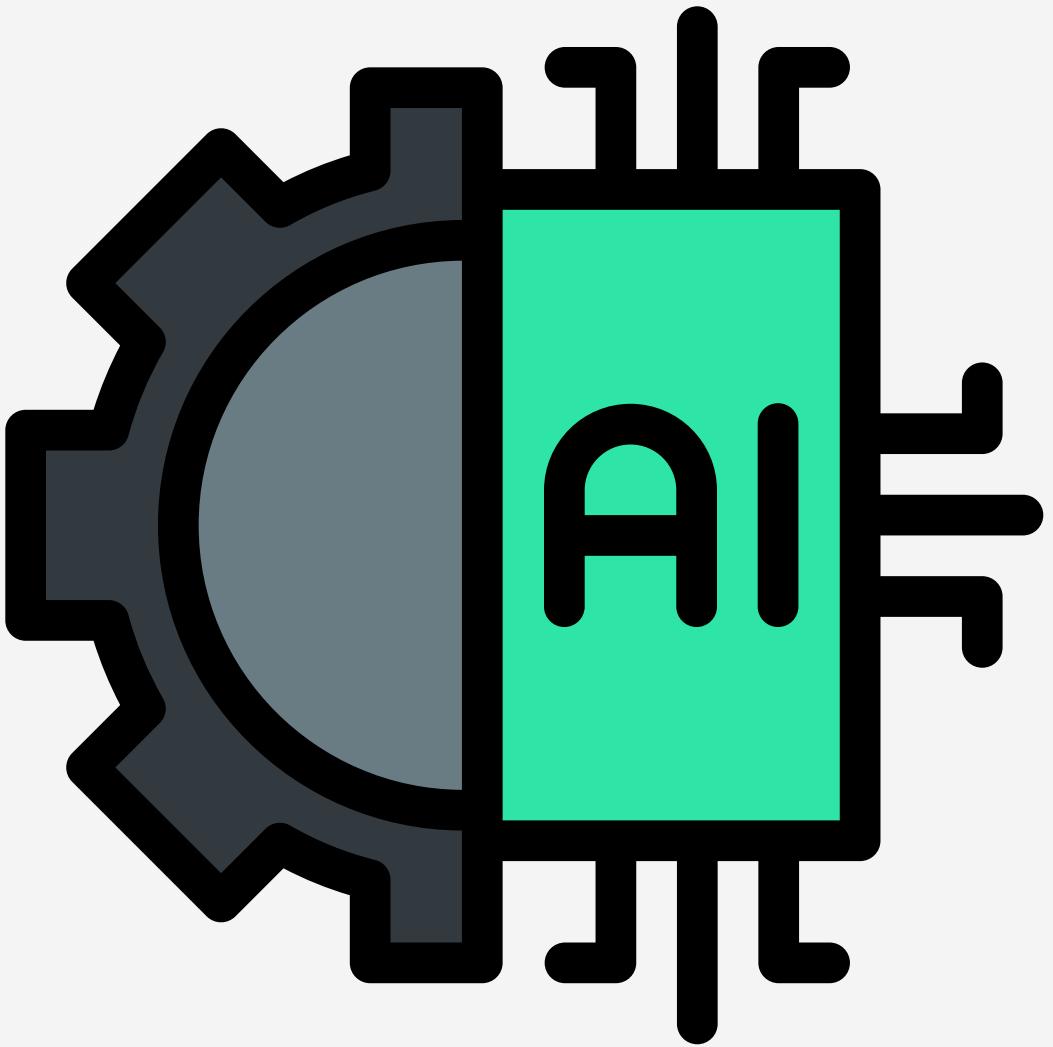
4-1-1 Makine Öğrenmesi Bazlı Sıralama

Bu sistemler, kullanıcı tercihlerini öğrenerek en iyi görselleştirmeyi belirlemeye çalışır.



4-1-1 Makine Öğrenmesi Bazlı Sıralama

- **Draco:** Yumuşak kısıtlar görselleştirme tercihlerine dayalı bir maliyet fonksiyonu oluşturur. Kısıtları ihlal eden görselleştirmeler cezalandırılır ve RankSVM modeli ile en iyi seçenekler belirlenir.
- **DeepEye:** Örneklerden öğrenerek otomatik görselleştirme önerir. Önce "Visualization Recognition" (iyi/kötü ayrımı) yapar, ardından "Visualization Ranking" ile en iyi görselleştirmeleri sıralar. Karar ağaçları ve LambdaMART algoritması kullanılarak model eğitilir.
- **Data2Vis:** RNN kullanarak JSON formatındaki verileri Vega-Lite görselleştirme kodlarına çevirir. Model, görselleştirme tasarımını bir sıralı çeviri problemi olarak ele alır.



4-1-1 Makine Öğrenmesi Bazlı Sıralama

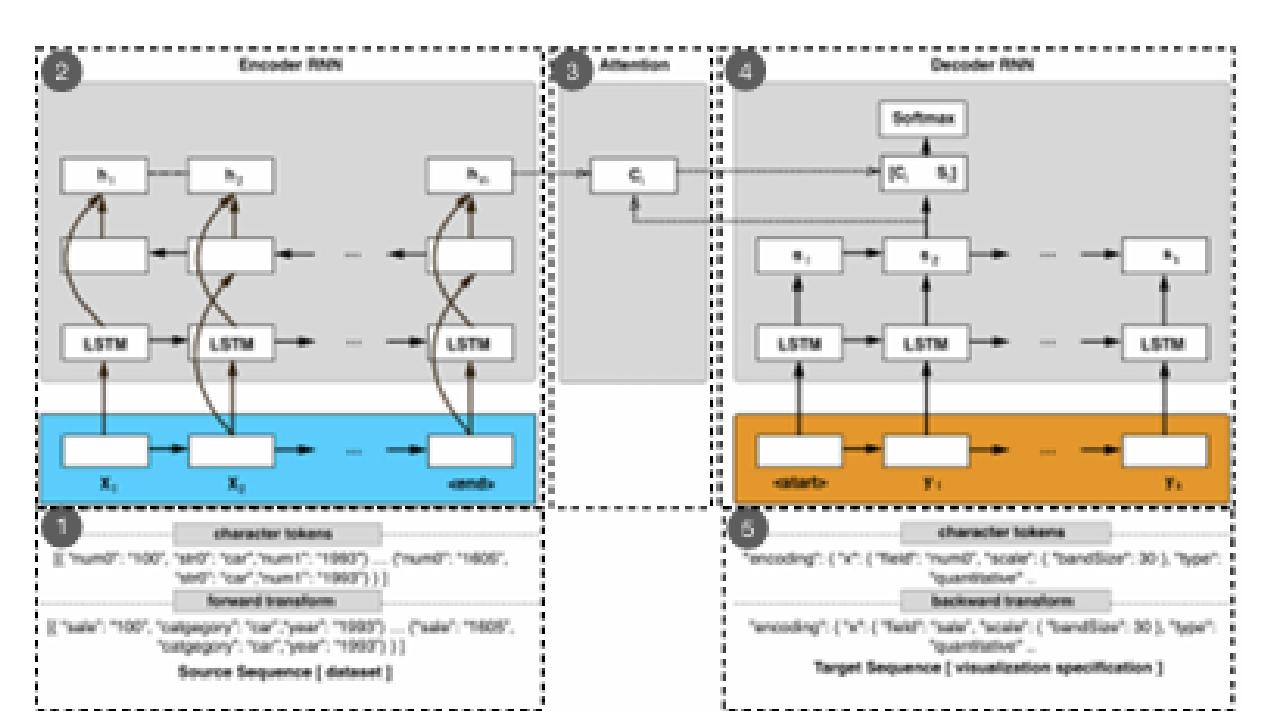


Fig. 11 The architecture of Data2Vis [120]. Data2Vis is built based on sequence to sequence model with the encoder–decoder architecture (② and ④) and the attention mechanism (③). It takes original datasets (①) as input and automatically recommends visualizations (⑤) by given datasets (①)

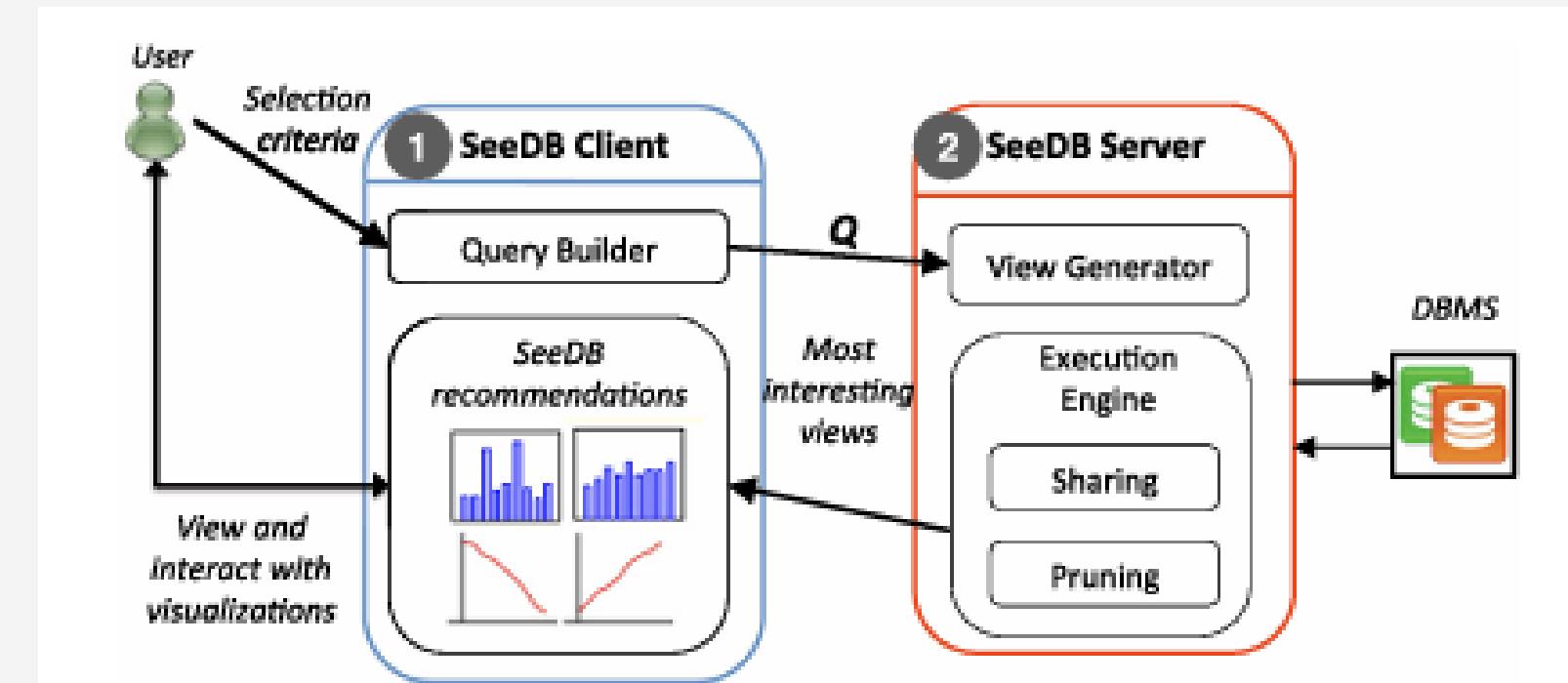


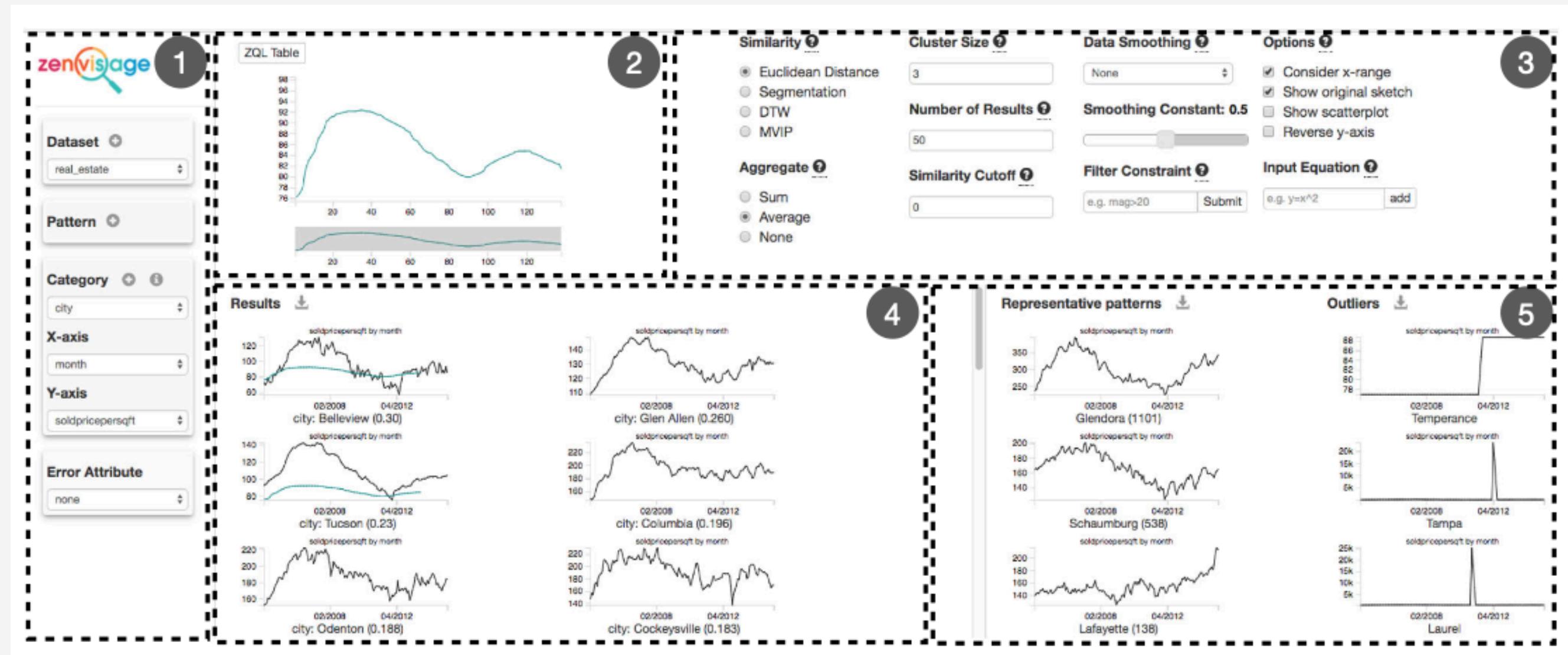
Fig. 12 The architecture of SeeDB [22]. Client (①) accepts users' input, constructs visualization queries and shows recommended visualizations to users. Server (②) generates visualization candidates (view generator) and recommends visualizations to users (execution engine)

4-1-2 Referansa Dayalı Spesifikasyon

Kullanıcının belirlediği bir görselleştirmeye benzer veya farklı özelliklere sahip alternatif görseller önerilir.

Zenvisage bu sistemlerden biri olarak örnek verilebilir.

4-1-2 Referansa Dayalı Spesifikasyon



[Gündem Sayfasına Geri Dön](#)

4-1-2 Referansa Dayalı Spesifikasyon

- **Veri yükleme & seçim:** Kullanıcı veri setini yükler, X-Y eksenlerini ve kategoriyi belirler.
- **Referans trend:** Kullanıcı bir referans trend seçer.
- **Parametre ayarları:** Benzerlik ve toplama fonksiyonları gibi sistem parametreleri ayarlanır.
- **Benzer görseller:** Referans trende benzer görseller gösterilir.
- **Öneriler:** Tipik ve sıra dışı trendler otomatik olarak önerilir.

4-1-2 Sapma Tabanlı Öneriler



SeeDB:

- Kullanıcı, ilgilendiği veri sorgusunu (DQ) ve karşılaştırma için bir referans veri kümesini (DR) belirler.
- SeeDB, bu iki veri kümesi arasında farklı kombinasyonlarla oluşturulan görselleştirmeleri karşılaştırır.
- En büyük sapmaya sahip olan top-k görselleştirmeleri önerir.

4-1-2 Anomali Tabanlı Öneriler

- Anomaly detection methods represent the visualization process by classifying normal and abnormal points.
- Selecting the best distinguishing visualizations for anomalies uses the independence measure (mutual information and entropy).



4-1-2 Benzerlik/Uzaklık Tabanlı Öneriler

- Kullanıcı, aradığı trend veya deseni görsel olarak çizer.
- Sistem, öklid mesafesi veya Dinamik Zaman Eğme (DTW) yöntemi ile en benzer veya en farklı görselleştirmeleri önerir.



4-2 Davranış Tabanlı Öneriler

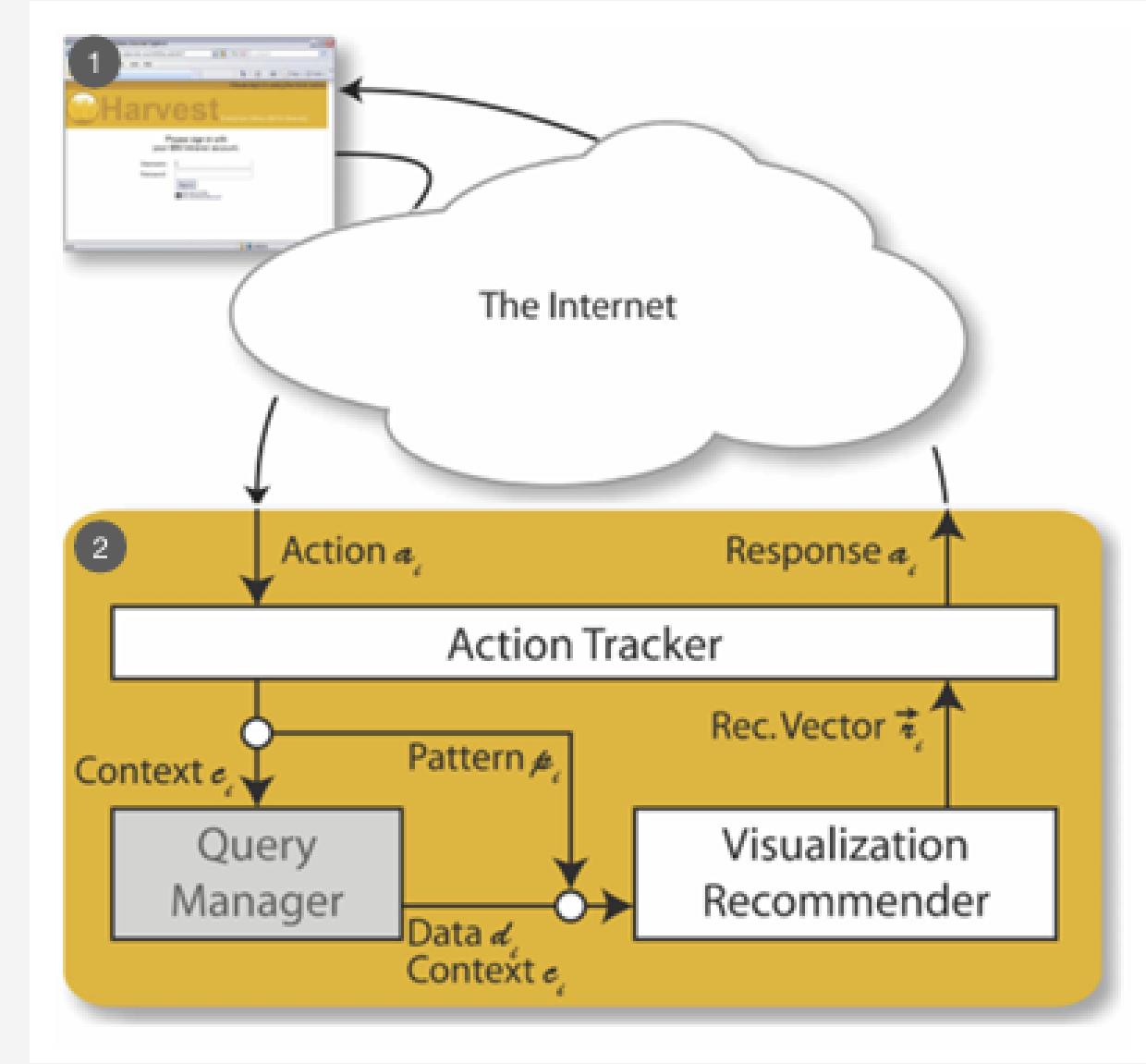
Kullanıcıların mevcut davranışlarını girdi olarak alır, ardından amaçları tahmin etmeye çalışarak uygun görselleştirmeleri önerir.



HARVEST bu sistemelerden biri olarak örnek verilebilir.

4-2 Davranış Tabanlı Öneriler

HARVEST, kullanıcıların davranışlarını analiz ederek görevlerini tahmin eden ve buna göre görselleştirme öneren bir sistemdir. Kullanıcıların eylemleri izlenir ve bu davranışlara göre görselleştirmeler önerilir.



4-3 Kişiiselleştirilmiş Öneriler

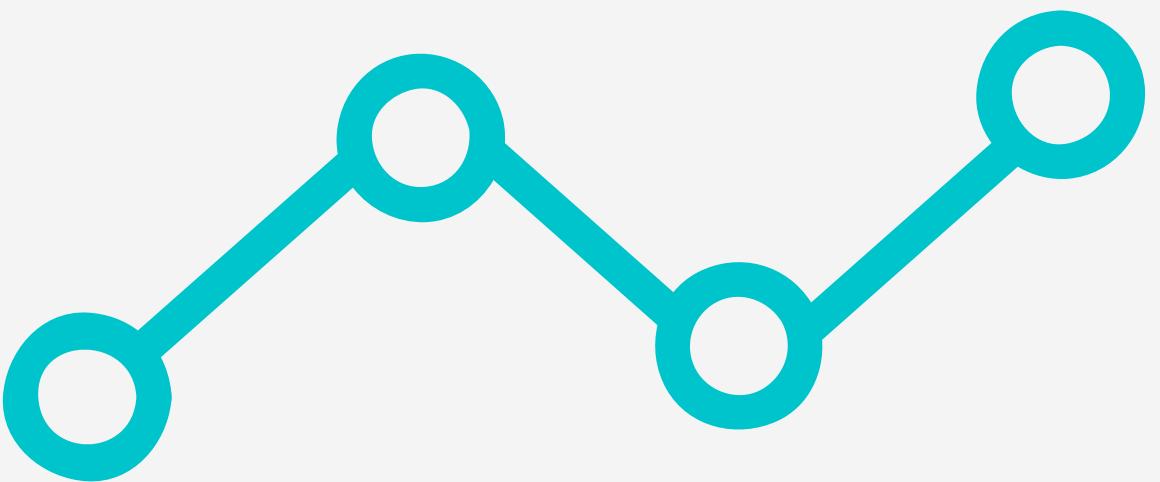
Kullanıcıların geçmiş davranışlarını kullanarak ilgi çekici görselleştirmeleri önerir.



4-3-1 Doğrusal Model

VizDeck:

- Kullanıcıların geçmiş davranışlarını kullanarak her kullanıcı için doğrusal bir model eğitir.
- Kullanıcılar, önerilen görselleştirmeleri **vizlet** adı verilen küçük elemanlar olarak bir izgarada görüntüler ve bu elemanları sıralayarak etkileşimli bir panoda gösterebilir.



4-3-2 İşbirlikçi Filtreleme

VizRec:

- **İşbirlikçi filtreleme:** Kullanıcılar arasındaki benzerlikleri kullanarak öneriler yapar.
- **İçerik tabanlı filtreleme:** Yeni kullanıcılar için öneriler yaparken, görselleştirmelerin özelliklerine dayalı içerik profilleri oluşturur ve bunları kullanarak önerilerde bulunur.
- **Hibrit filtreleme:** İçerik tabanlı ve işbirlikçi filtrellemeyi birleştirerek daha dinamik ve adaptif bir öneri sistemi oluşturur.



5- Tartışmalar

- Görselleştirme için veri hazırlığı
- Veri görselleştirme ölçütleri
- Veri tabanı bağlantılı uygulamalar için veri görselleştirme



[Gündem Sayfasına Geri Dön](#)



5-1 Görselleştirme İçin Veri Hazırlığı

- Günümüzde veri genelde hatalar içerir ve bu bizleri yanıltabilir.
- Aykırı ve/veya eksik değerlerin veri setine etkisini inceleyen birkaç çalışmaya makalede değinilmiştir.
- Yanlı görselleştirmelerin tespiti ve görev odaklı veri temizleme konuları araştırmaya değer olarak ele alınmıştır.

5-2 Veri Görselleştirme Ölçütleri



- Performans ve tavsiyeler için ölçüt geliştirmek önemli bir husustur.
- Görselleştirmede kategorizasyon ve veri eğitimi konuları iyi bir araştırma konusu olabilir.

5-3 Veri Tabanı Bağlantılı Uygulamalar İçin Veri Görselleştirme

- Excel gibi veri tabanı *tabanlı* uygulamalar önemli bir rol almaktadır.
- Veri keşfi ve hata ayıklamak için veri görselleştirme araştırılabilecek konular olarak öne çıkmaktadır.

[Gündem Sayfasına Geri Dön](#)



6- Sonuç

- Veri görselleştirme hızlı gelişen bir alan olup birçok yeni çalışma ve sistem yayımlanmıştır.
- Bu makale veri görselleştirmeyi veri yönetimi açısından ele alarak güncel durumu bizlere göstermiştir.

TEŞEKKÜRLER !

