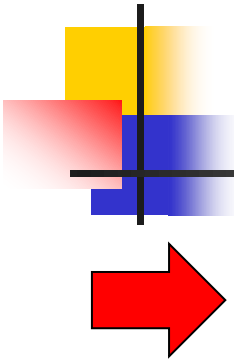
# Knowledge Understanding for Data Analytics

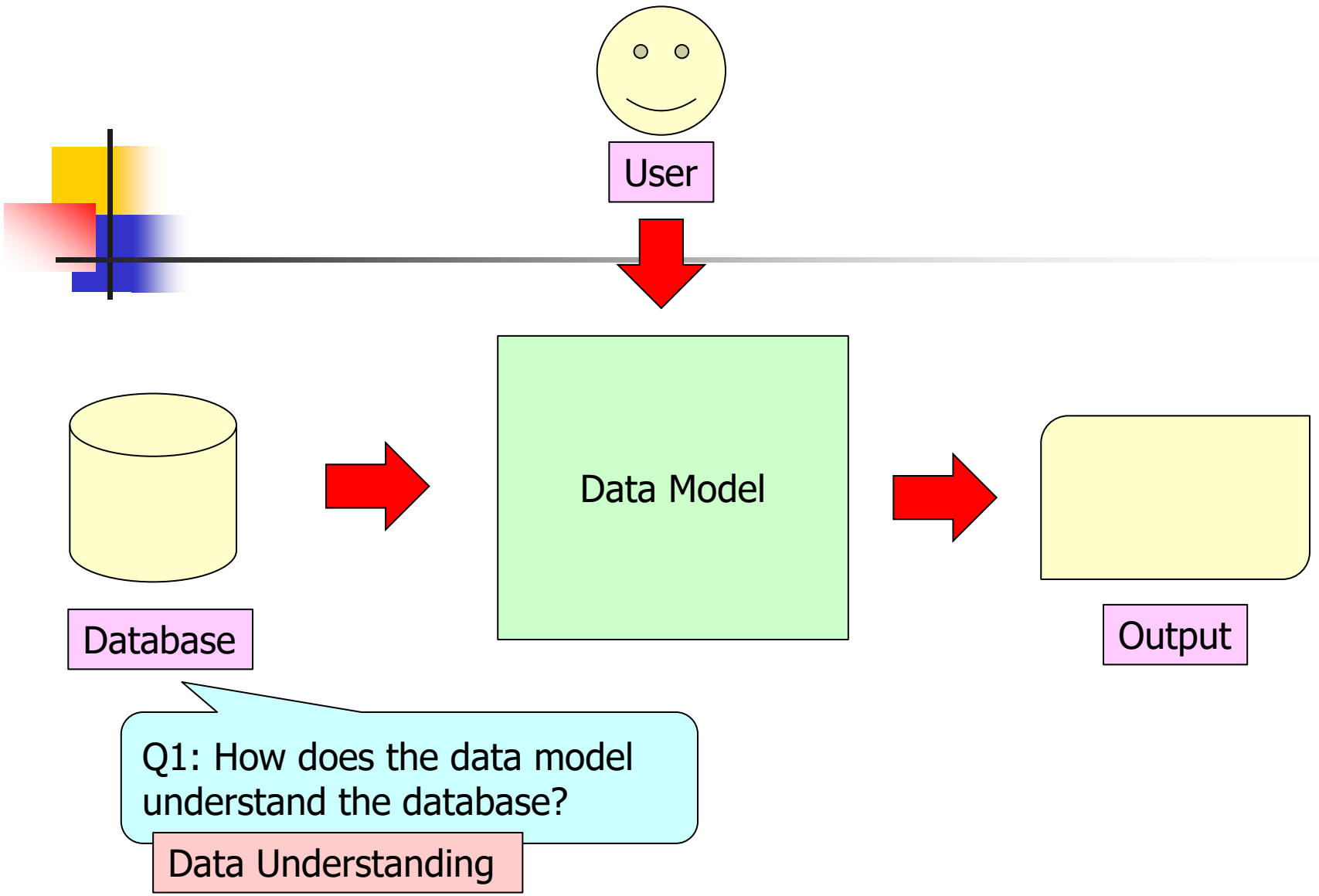
Raymond Chi-Wing Wong

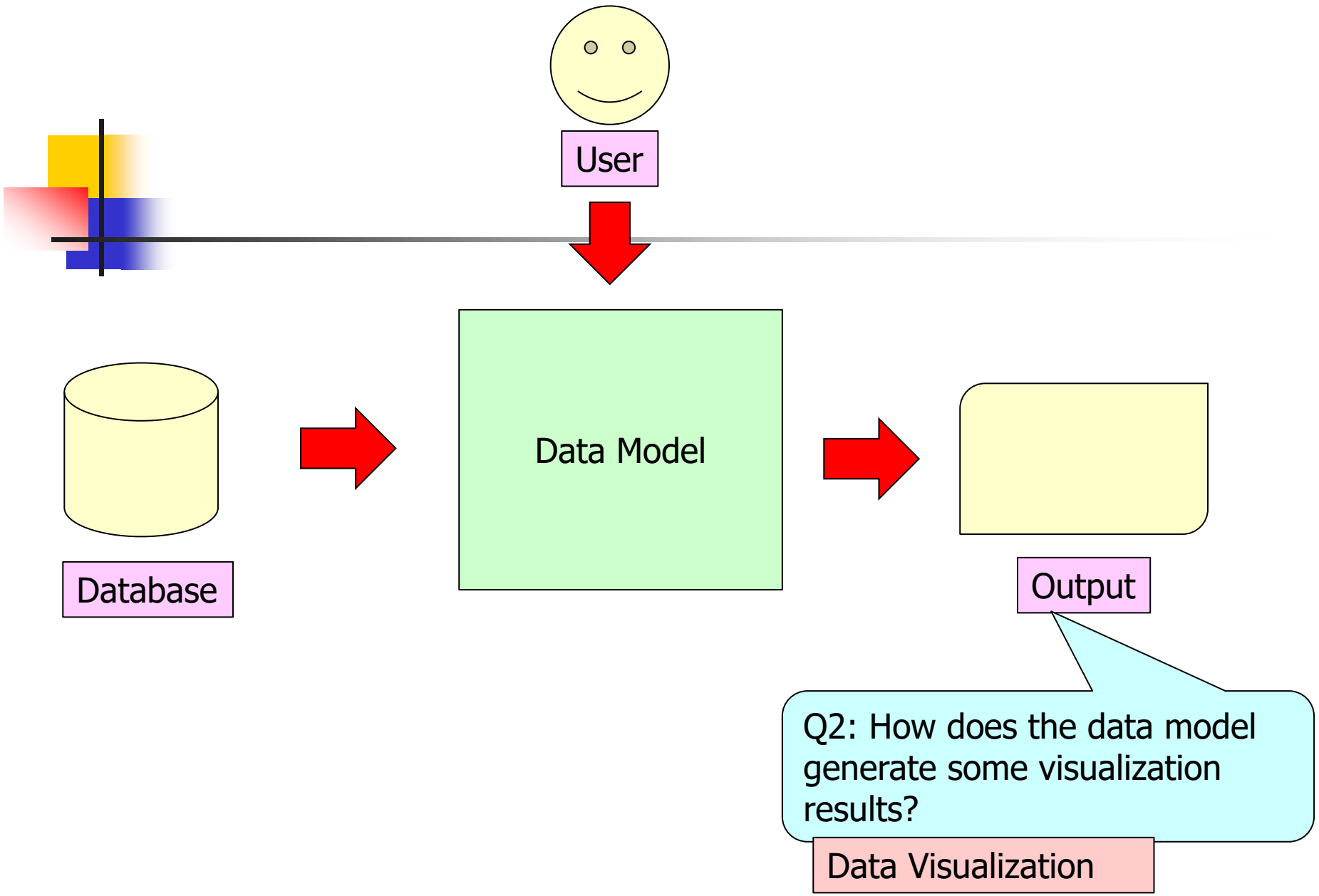
The Hong Kong University of Science and Technology

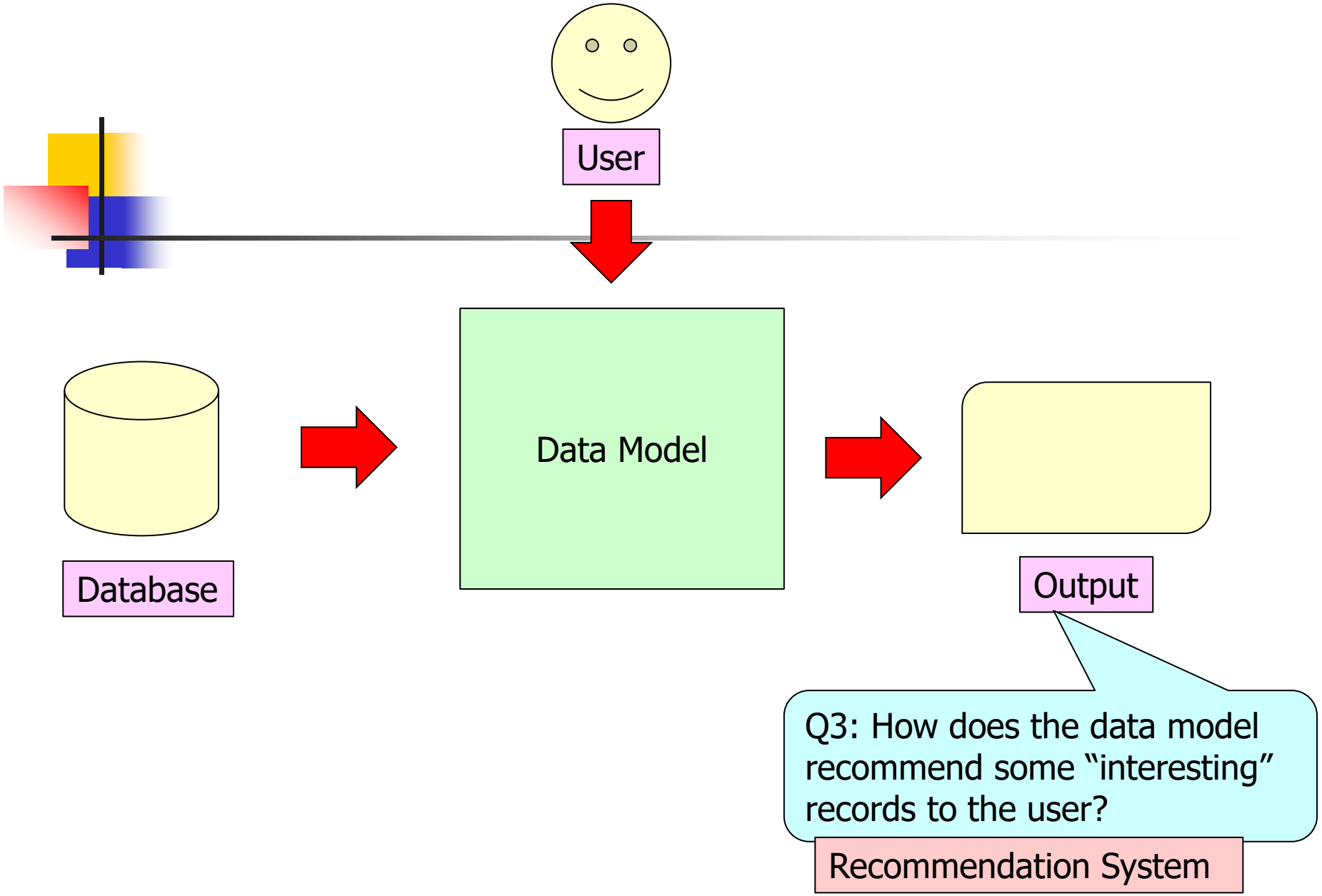


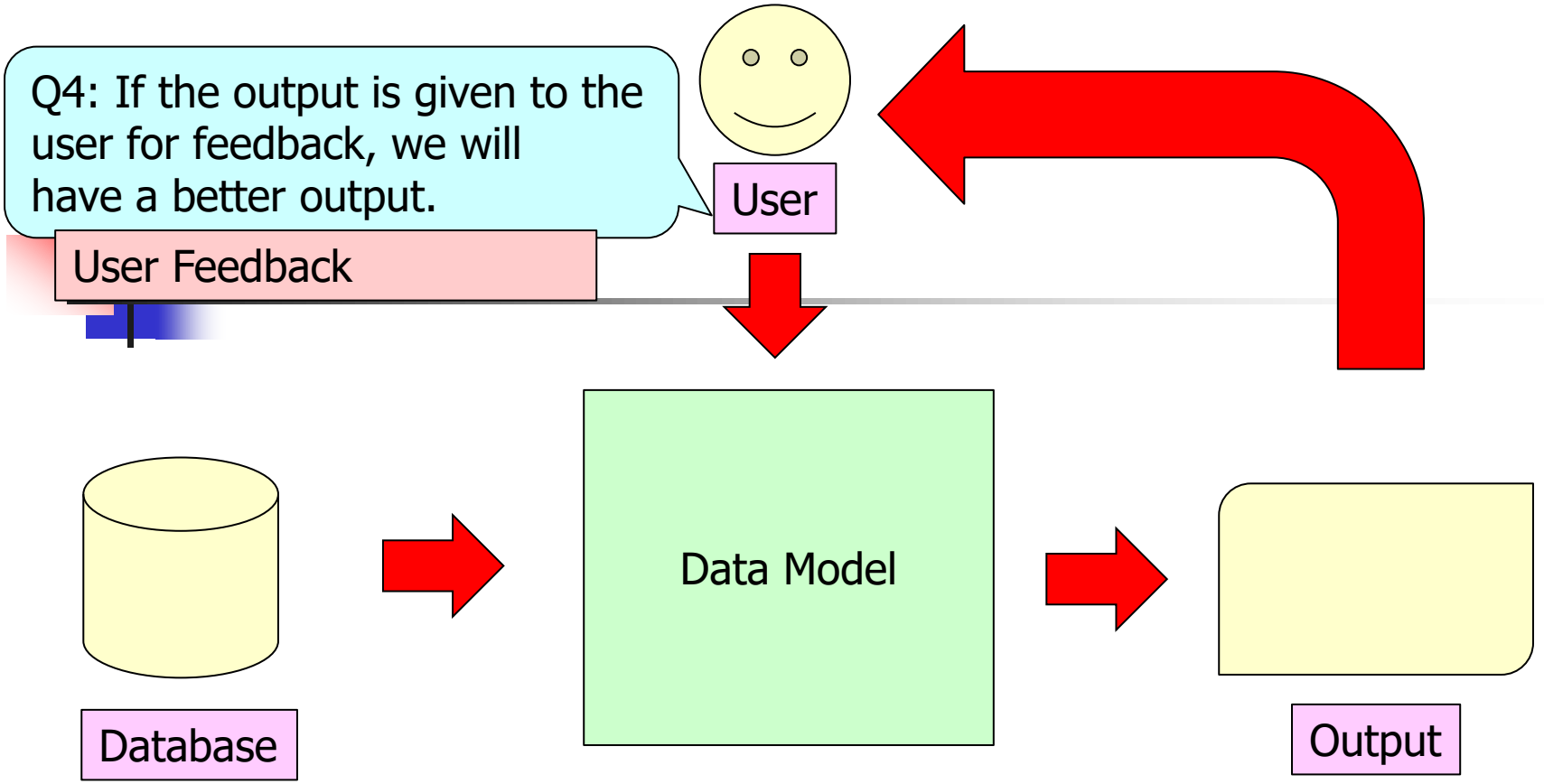
# Outline

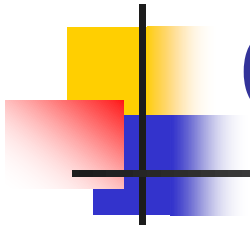
- Topic
  - Data Understanding
  - Data Visualization
  - Recommendation System
  - User Feedback
- Conclusion


User
Data Model
Database
Output
Q1: How does the data model understand the database?
Data Understanding


User
Data Model
Database
Output
Q2: How does the data model generate some visualization results?
Data Visualization

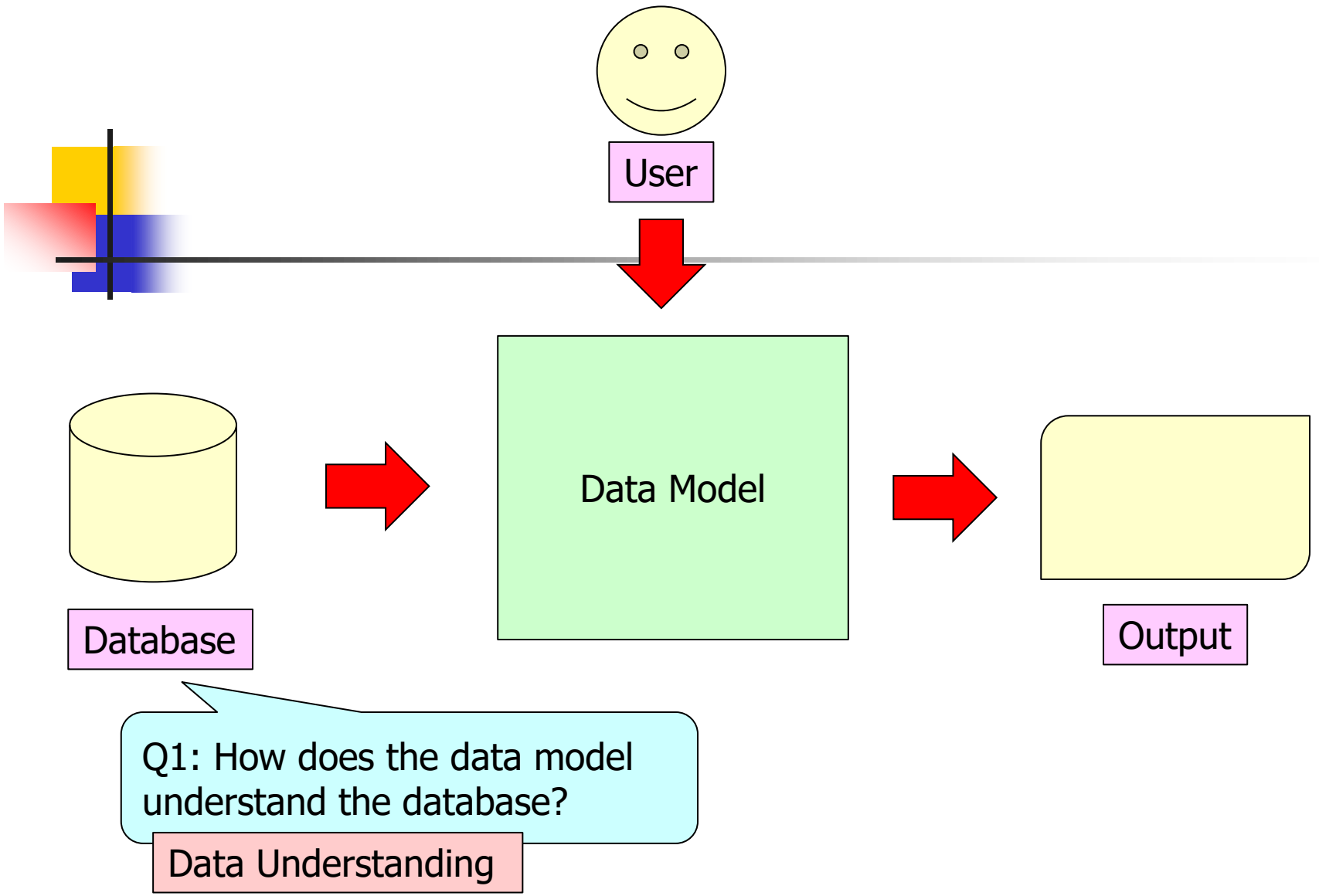

User
Data Model
Database
Output
Q3: How does the data model recommend some "interesting" records to the user?
Recommendation System

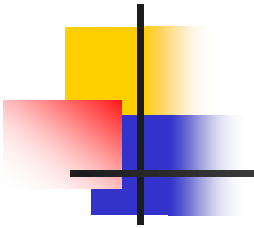

Q4: If the output is given to the user for feedback, we will have a better output.
User
User Feedback
Data Model
Database
Output



# Outline

- Topic
  - Data Understanding
  - Data Visualization
  - Recommendation System
  - User Feedback
- Conclusion


User
Data Model
Database
Output
Q1: How does the data model understand the database?
Data Understanding

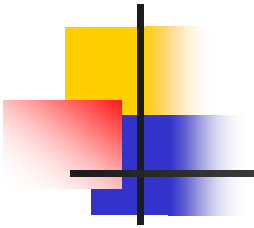


- A user may ask the following question.
  - What is the average age of students who have a pet?
- The model will generate the following SQL statement.

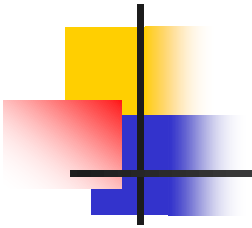
```
select avg(age)
from students
where pet != null
```

- It will execute this statement

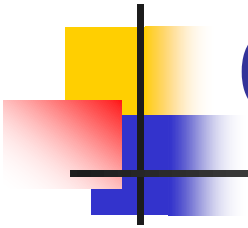
25.4



- We propose a schema encoder to understand the database.
- Then, we propose an encoder-decoder framework model to generate the SQL statement.
- Then, we execute the SQL statement on the database to generate the output.

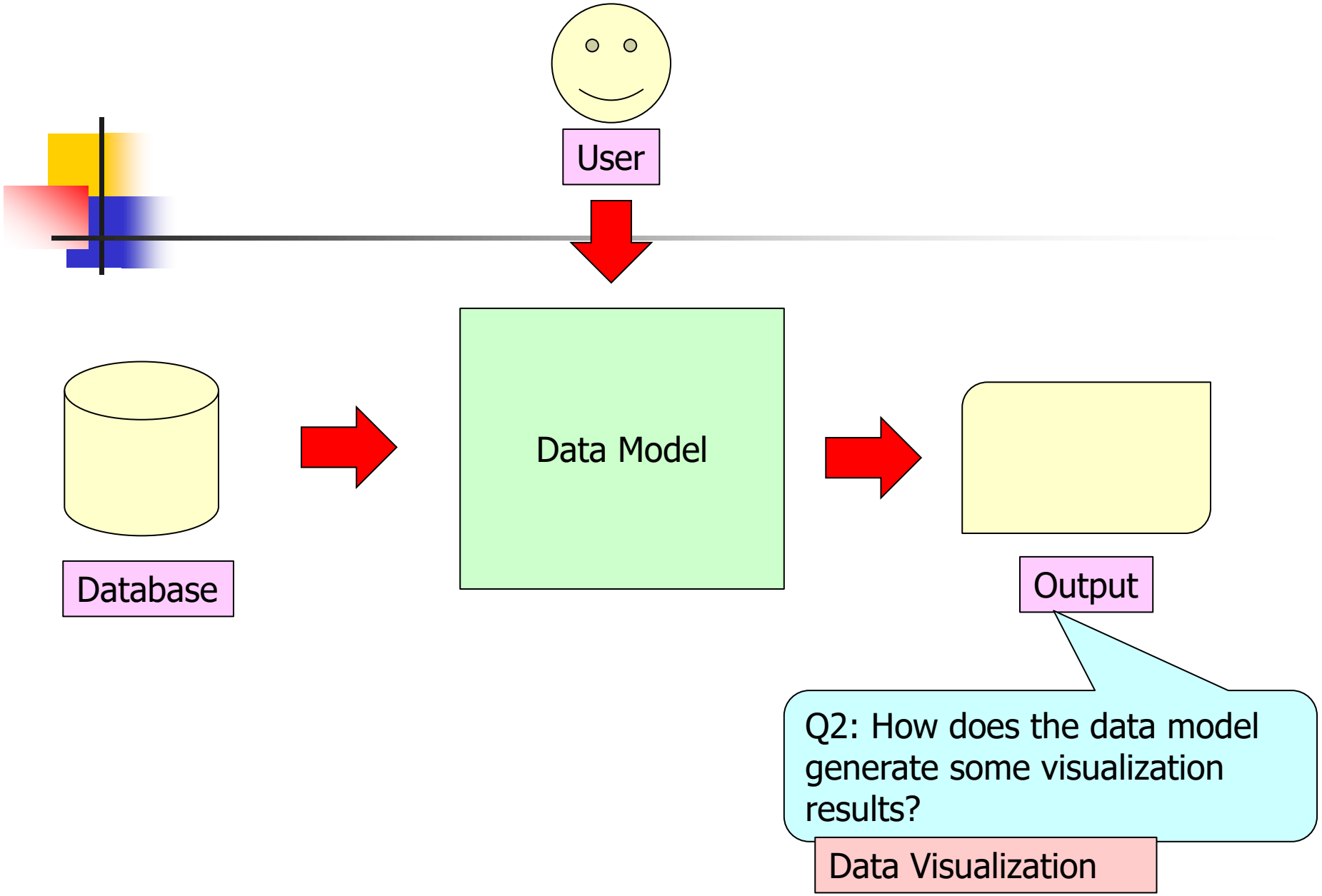


- My previous work
  - Natural Language Interfaces for Tabular Data Querying and Visualization: A Survey (TKDE 2024)
  - Speech-to-SQL: Towards Speech-driven SQL Query Generation From Natural Language Question (VLDBJ 2024)
  - VoiceQuerySystem: A Voice-driven Database Querying System Using Natural Language Questions (Demo) (SIGMOD 2022)

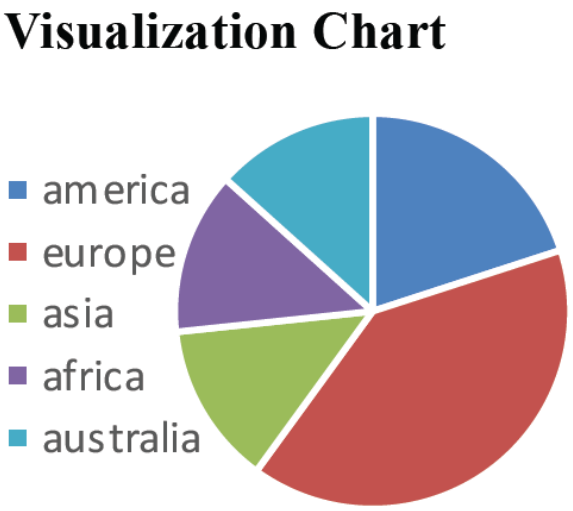


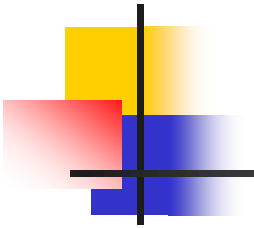
# Outline

- Topic
  - Data Understanding
  - Data Visualization
  - Recommendation System
  - User Feedback
- Conclusion

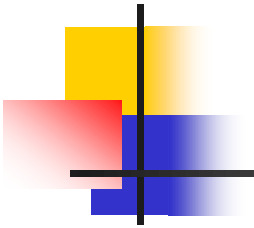

User
Data Model
Database
Output
Q2: How does the data model generate some visualization results?
Data Visualization

- A user may ask the following question.
  - How many countries does each continent have?
  - List the continent name with the percentage of countries in a pie chart.
- The model will generate

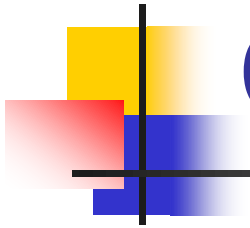




- We propose a novel hybrid retrieval-generation framework.
- We first retrieve the most relevant data visualization query candidate (from the database) (using a neural ranking model).
- We then revise the candidate (using a GNN-based encoder-decoder framework).
- We generate the visualization based on the candidate.

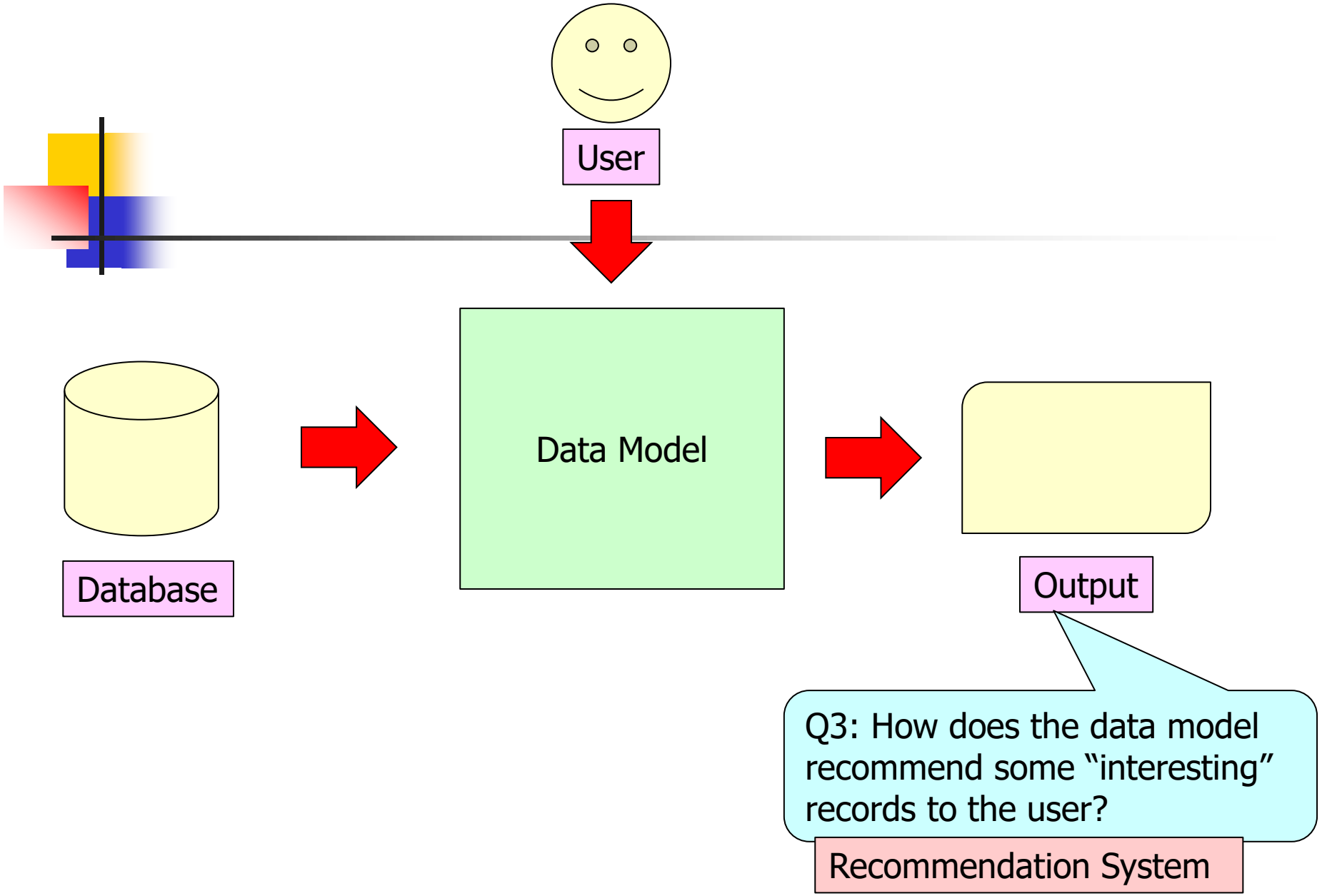


- My previous work
  - FeVisQA: Free-form Question Answering over Data Visualizations (ICDE 2025)
  - Towards Robustness of Text-to-Visualization Translation against Lexical and Phrasal Variability (ICDE 2025)
  - DataVisT5: A Pre-trained Language Model for Jointly Understanding Text and Data Visualization (ICDE 2025)
  - Marrying Dialogue Systems with Data Visualization: Interactive Data Visualization Generation from Natural Language Conversations (KDD 2024)
  - Automatic Data Visualization Generation from Chinese Natural Language Questions (LREC-COLING 2024)
  - Natural Language Generation Meets Data Visualization: Vis-to-Text and its Duality with Text-to-Vis (ICDM 2023)
  - RGVisNet: A Hybrid Retrieval-Generation Neural Framework Towards Automatic Data Visualization Generation (KDD 2022)



# Outline

- Topic
  - Data Understanding
  - Data Visualization
  - Recommendation System
  - User Feedback
- Conclusion


User
Data Model
Database
Output
Q3: How does the data model recommend some "interesting" records to the user?
Recommendation System

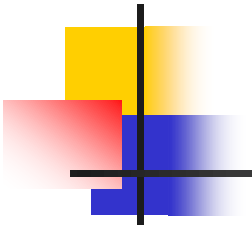




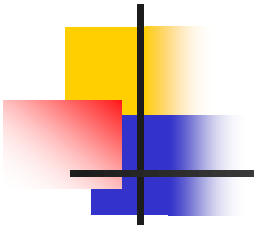





SENNHEISER



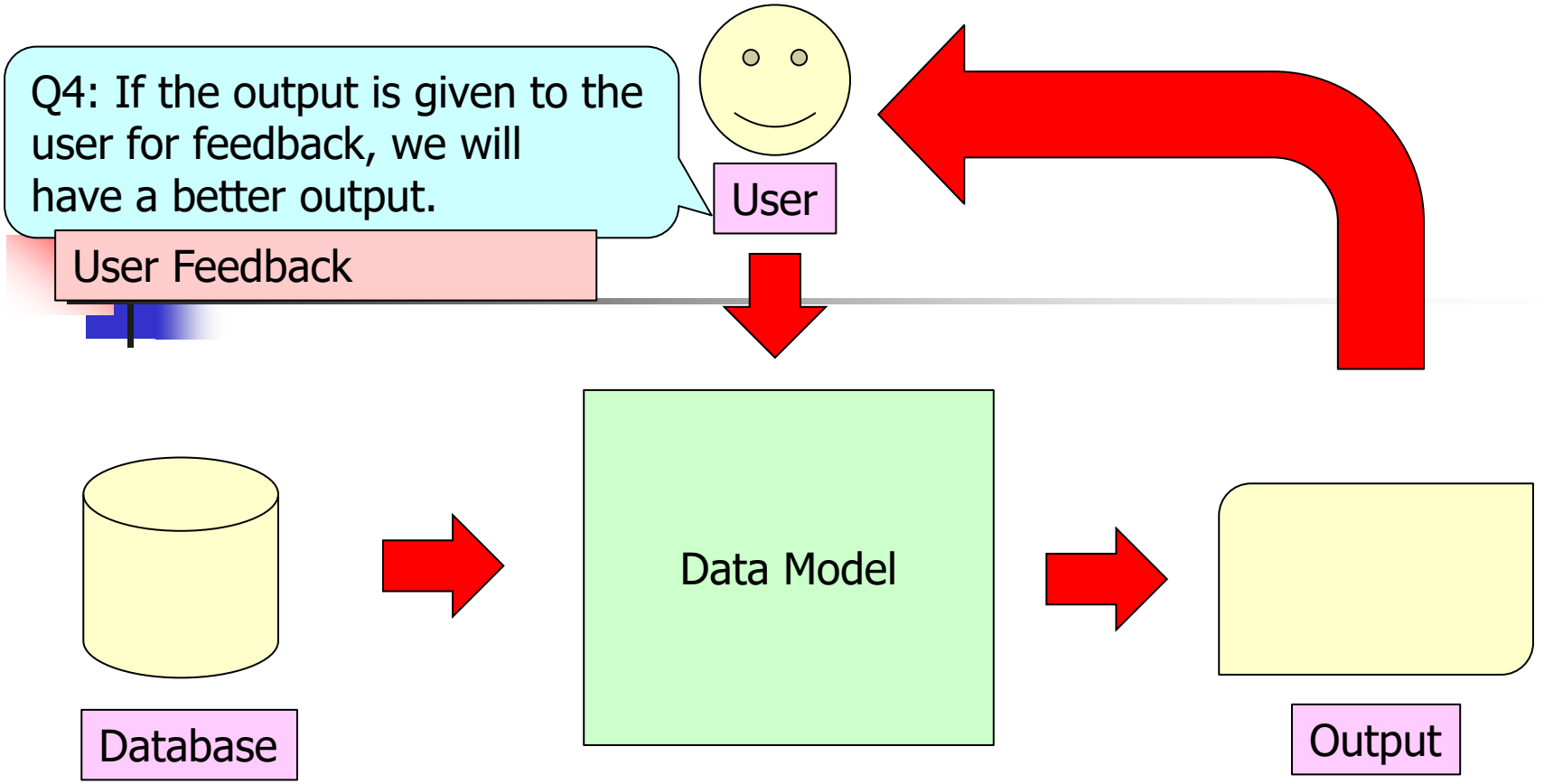
- We propose deep learning models (e.g., graph neural network models) for this recommendation system.



- My previous work
  - SR-PredictAO: Session-based Recommendation with High-Capability Predictor Add-On (ICDM 2024)
  - Improving Representation Learning for Session-based Recommendation (IEEE BigData 2022)
  - An Efficient and Effective Framework for Session-based Social Recommendation (WSDM 2021)
  - Handling Information Loss of Graph Neural Networks for Session-based Recommendation (KDD 2020)
  - Session-based Recommendation with Local Invariance (ICDM 2019)

# Outline

- Topic
  - Data Understanding
  - Data Visualization
  - Recommendation System
  - User Feedback
- Conclusion


Q4: If the output is given to the user for feedback, we will have a better output.
User
User Feedback
Data Model
Database
Output

Which apartment should Raymond buy?

Suppose that user Raymond wants to buy an apartment

If the value is larger, then it is better to a user. One example is the apartment size.

There are 2 popular queries for this problem.

- Top-k queries
- Skyline queries

My recent research focuses on a new type of queries.

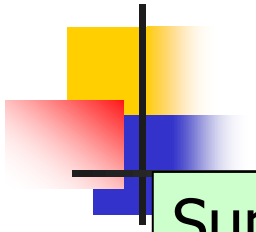
- k-regret queries

Advantage: The output size is "fixed"

D

| Apartment | $X_1$ | $X_2$ |
|---|---|---|
| $p_1$ | 0 | 1 |
| $p_2$ | 0.2 | 1 |
| $p_3$ | 0.6 | 0.9 |
| $p_4$ | 0.9 | 0.6 |
| $p_5$ | 1 | 0.2 |
| $p_6$ | 1 | 0 |
| … | … | … |

Disadvantage: We need to know the "exact" utility function of Raymond



Suppose that user Raymond wants to buy an apartment

Top-k queries

D

| Apartment | $X_1$ | $X_2$ |
|---|---|---|
| $p_1$ | 0 | 1 |
| $p_2$ | 0.2 | 1 |
| $p_3$ | 0.6 | 0.9 |
| $p_4$ | 0.9 | 0.6 |
| $p_5$ | 1 | 0.2 |
| $p_6$ | 1 | 0 |
| … | … | … |

## Top-k queries

Suppose that user Raymond wants to buy an apartment

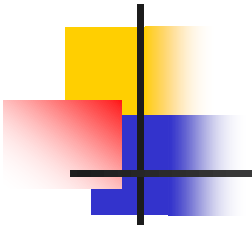
- Assume that Raymond has a "known" utility function.
- Utility function f
  $f(p) = 0.3\ X_1 + 0.7\ X_2$
- Utility vector u = (0.3, 0.7)
- Suppose that we want to find the top-1 apartment.
- **Output**
  **Maximum utility point** of D = $p_3$

D

| Apartment | $X_1$ | $X_2$ | Utility |
|---|---|---|---|
| $p_1$ | 0 | 1 | 0.7 |
| $p_2$ | 0.2 | 1 | 0.76 |
| $p_3$ | 0.6 | 0.9 | 0.81 |
| $p_4$ | 0.9 | 0.6 | 0.69 |
| $p_5$ | 1 | 0.2 | 0.44 |
| $p_6$ | 1 | 0 | 0.3 |
| … | … | … | … |

Advantage: The output size is "fixed"

Disadvantage: We need to know the "exact" utility function of Raymond



- My previous work
  - k-Hit Query: Top-k Query with Probabilistic Utility Function (SIGMOD 2015)

Which apartment should Raymond buy?

Suppose that user Raymond wants to buy an apartment

If the value is larger, then it is better to a user. One example is the apartment size.

There are 2 popular queries for this problem.

- Top-k queries
- Skyline queries

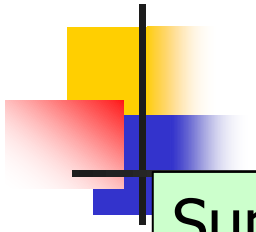
My recent research focuses on a new type of queries.

Advantage: There is no need to specify the utility function of Raymond

Disadvantage: The output size is uncontrollable.

D

| Apartment | $X_1$ | $X_2$ |
|---|---|---|
| $p_1$ | 0 | 1 |
| $p_2$ | 0.2 | 1 |
| $p_3$ | 0.6 | 0.9 |
| $p_4$ | 0.9 | 0.6 |
| $p_5$ | 1 | 0.2 |
| $p_6$ | 1 | 0 |
| … | … | … |



Suppose that user Raymond wants to buy an apartment

Skyline queries

D

| Apartment | $X_1$ | $X_2$ |
|---|---|---|
| $p_1$ | 0 | 1 |
| $p_2$ | 0.2 | 1 |
| $p_3$ | 0.6 | 0.9 |
| $p_4$ | 0.9 | 0.6 |
| $p_5$ | 1 | 0.2 |
| $p_6$ | 1 | 0 |
| … | … | … |

## Suppose that user Raymond wants to buy an apartment

- There is no assumption that we know the "exact" utility function of Raymond
- There is a concept called "dominance"

$p_2$ dominates $p_1$ because
(1) the $X_1$ value of $p_2$ is better than that of $p_1$.
(2) the $X_2$ value of $p_2$ is equal to that of $p_1$.

D

| Apartment | $X_1$ | $X_2$ |
|---|---|---|
| $p_1$ | | |
| $p_2$ | 0.2 | 1 |
| $p_3$ | 0.6 | 0.9 |
| $p_4$ | 0.9 | 0.6 |
| $p_5$ | 1 | 0.2 |
| $p_6$ | 1 | 0 |
| … | … | … |

## Suppose that user Raymond wants to buy an apartment

- There is no assumption that we know the "exact" utility function of Raymond
- There is a concept called "dominance"

D

| Apartment | $X_1$ | $X_2$ |
|---|---|---|
| $p_1$ | | |
| $p_2$ | 0.2 | 1 |
| $p_3$ | 0.6 | 0.9 |
| $p_4$ | 0.9 | 0.6 |
| $p_5$ | 1 | 0.2 |
| $p_6$ | | |
| … | … | … |

$p_5$ dominates $p_6$ because
(1) the $X_1$ value of $p_5$ is equal to that of $p_6$.
(2) the $X_2$ value of $p_5$ is better than that of $p_6$.

## Skyline queries

**Suppose that user Raymond wants to buy an apartment**

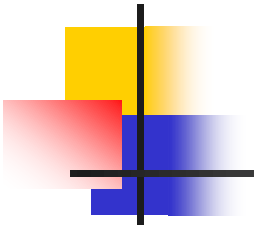
- There is no assumption that we know the "exact" utility function of Raymond
- There is a concept called "dominance"
- Apartments are called **skyline apartments** if they are not dominated by any other apartments
- **Output**
  Skyline apartments = $\{p_2, p_3, p_4, p_5\}$

D

| Apartment | $X_1$ | $X_2$ |
|---|---|---|
| $p_1$ | | |
| $p_2$ | 0.2 | 1 |
| $p_3$ | 0.6 | 0.9 |
| $p_4$ | 0.9 | 0.6 |
| $p_5$ | 1 | 0.2 |
| $p_6$ | | |
| ... | ... | ... |

Advantage: There is no need to specify the utility function of Raymond

Disadvantage: The output size is uncontrollable.



- My previous work
  - Skyline Queries and Pareto Optimality (Encyclopedia of Database Systems, 2016)
  - Finding Competitive Price (SIGSPATIAL GIS 2013)
  - Finding Top-k Preferable Products (TKDE 2012)
  - Finding Top-k Profitable Products (ICDE 2011)
  - Creating Competitive Products (VLDB 2009)
  - Online Skyline Analysis with Dynamic Preferences on Nominal Attributes (TKDE 2009)
  - Finding the Influence Set through Skylines (EDBT 2009)
  - Efficient Skyline Querying with Variable User Preferences on Nominal Attributes (VLDB 2008)
  - Mining Favorable Facets (SIGKDD 2007)

Which apartment should Raymond buy?

Suppose that user Raymond wants to buy an apartment

If the value is larger, then it is better to a user. One example is the apartment size.

There are 2 popular queries for this problem.

- Top-k queries
- Skyline queries

My recent research focuses on a new type of queries.

- k-regret queries

D

| Apartment | $X_1$ | $X_2$ |
|---|---|---|
| $p_1$ | 0 | 1 |
| $p_2$ | 0.2 | 1 |
| $p_3$ | 0.6 | 0.9 |
| $p_4$ | 0.9 | 0.6 |
| $p_5$ | 1 | 0.2 |
| | | 0 |

Advantage: The output size is "fixed"

Advantage: There is no need to specify the utility function of Raymond

Suppose that user Raymond wants to buy an apartment

D

| Apartment | $X_1$ | $X_2$ |
|---|---|---|
| $p_1$ | 0 | 1 |
| $p_2$ | 0.2 | 1 |
| $p_3$ | 0.6 | 0.9 |
| $p_4$ | 0.9 | 0.6 |
| $p_5$ | 1 | 0.2 |
| $p_6$ | 1 | 0 |
| … | … | … |

k-regret queries

## k-regret queries

**Suppose that user Raymond wants to buy an apartment**

- It has **both** the advantage of the top-k queries and the advantage of the skyline queries.

The output size is specified by parameter k (e.g., 2)

Advantage: The output size is "fixed"

Advantage: There is no need to specify the utility function of Raymond

D

| Apartment | $X_1$ | $X_2$ |
|---|---|---|
| $p_1$ | 0 | 1 |
| $p_2$ | 0.2 | 1 |
| $p_3$ | 0.6 | 0.9 |
| $p_4$ | 0.9 | 0.6 |
| $p_5$ | 1 | 0.2 |
| $p_6$ | 1 | 0 |
| … | … | … |

## k-regret queries

Suppose that user Raymond wants to buy an apartment

- Outline
  - K-regret Queries
  - Interactive k-regret Queries

## k-regret queries

### Suppose that user Raymond wants to buy an apartment

- Consider that Raymond has a utility function with the utility vector (0.3, 0.7).
  But, we do not know this utility function.
- Suppose that the whole dataset is seen by user Raymond.
  - We could find his favorite apartment $p_3$ (Raymond's maximum utility point)

0.81

D

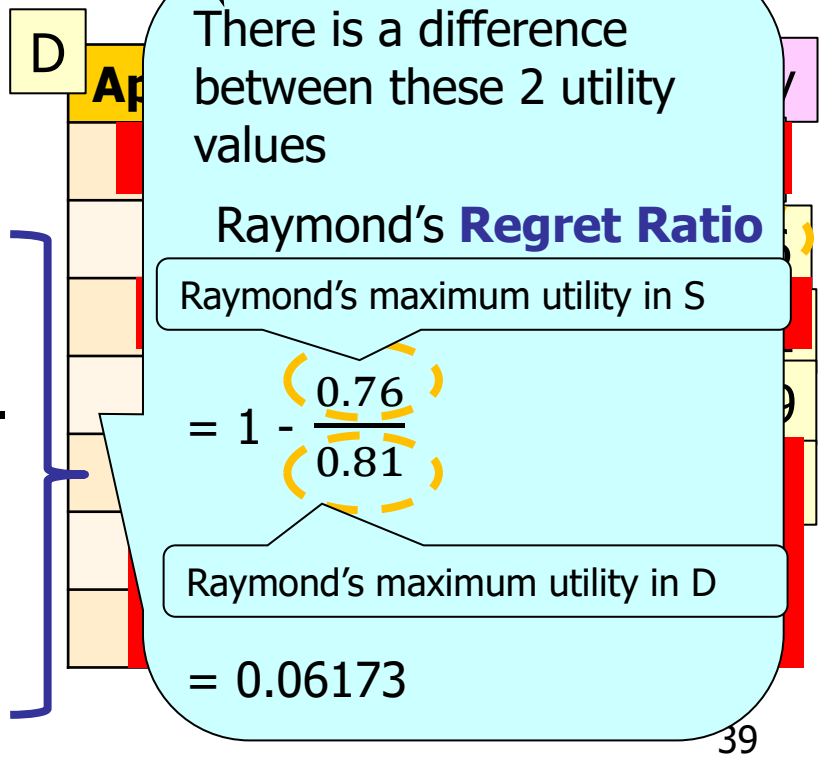
| Apartment | $X_1$ | $X_2$ | Utility |
|---|---|---|---|
| $p_1$ | 0 | 1 | 0.7 |
| $p_2$ | 0.2 | 1 | 0.76 |
| $p_3$ | 0.6 | 0.9 | 0.81 |
| $p_4$ | 0.9 | 0.6 | 0.69 |
| $p_5$ | 1 | 0.2 | 0.44 |
| $p_6$ | 1 | 0 | 0.3 |
| ... | ... | ... | ... |

## Suppose that user Raymond v

- Consider that Raymond has a utility function w… utility vector (0.3, 0.7).
  But, we do not know this utility function.
- Suppose that the whole dataset is seen by user Raymond.
  - We could find his favorite apartment $p_3$ (Raymond's maximum utility point) 0.81
- Consider a set S = {$p_2$, $p_4$} (which could be an output of this query).
- Suppose that set S is seen by user Raymond.
  - We could find his favorite apartment $p_2$ (Raymond's maximum utility point) 0.76

If Raymond's maximum utility in S is equal to Raymond's maximum utility in D, then Raymond's regret ratio is equal to 0

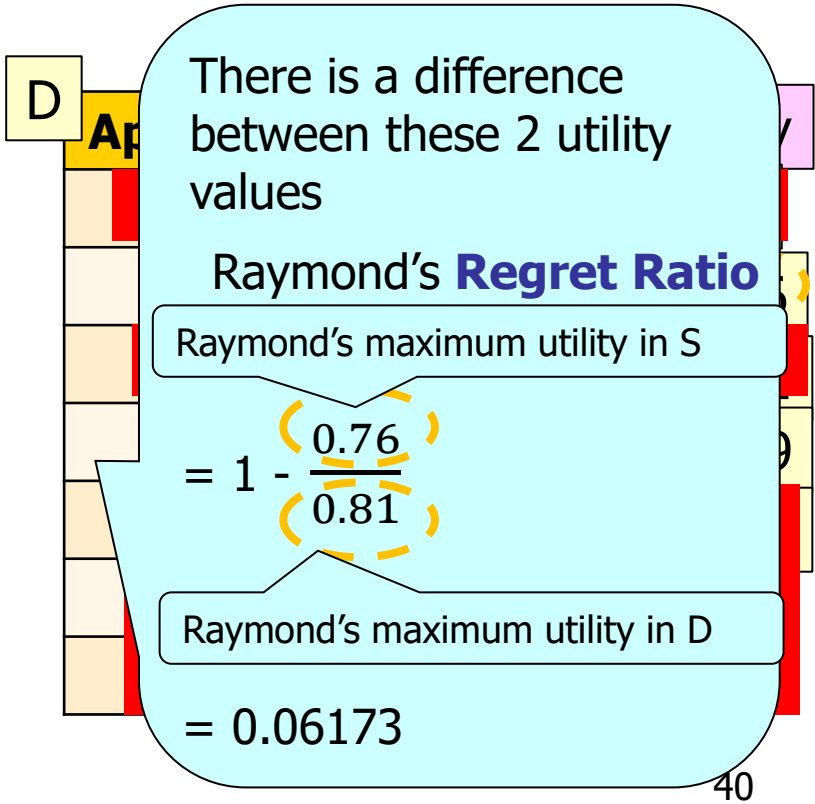
Thus, we would like to have a smaller value for Raymond's regret ratio.



## k-regret queries

### Suppose that user Raymond wants to buy an apartment

- Consider that Raymond has a utility function with the utility vector (0.3, 0.7).
  But, we do not know this utility function.



**Problem (k-regret):** Given a set D, we want to find a set S of k points such that the mrr of S is minimized.

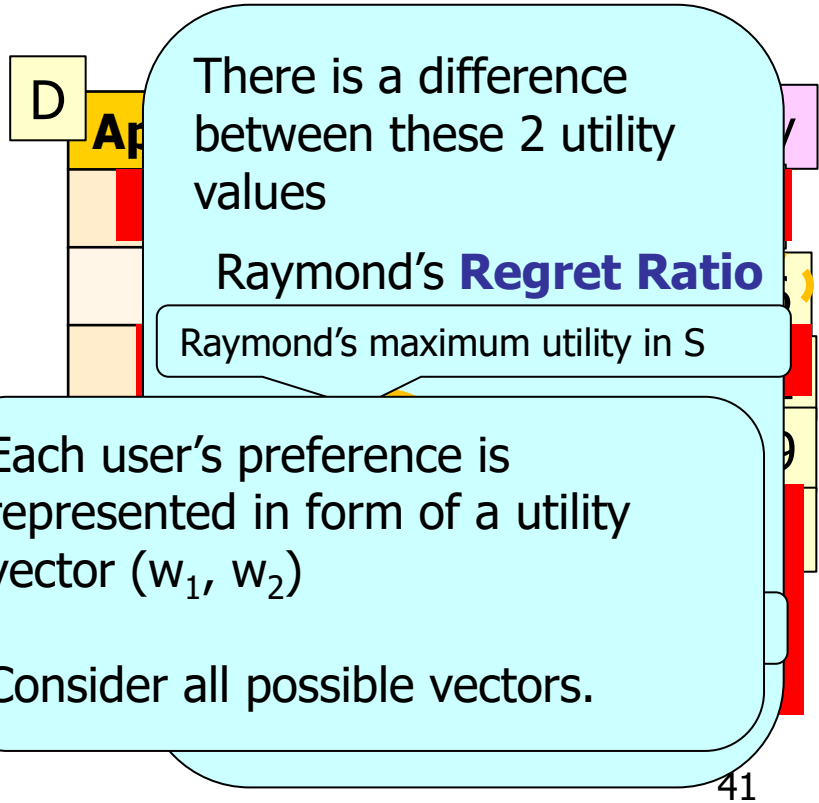
Advantage: The output size is "fixed"

Advantage: There is no need to specify the utility function of Raymond

## k-regret queries

Suppose that user

- Consider that Raymond has a utility function with the utility vector (0.3, 0.7).
  But, we do not know this utility function.
  - Raymond's Regret Ratio = 0.06173
- There are many other users (e.g., Mary and Peter).
- Each of them has different utility functions.
- E.g., Mary's Regret Ratio = 0.05120
  E.g., Peter's Regret Ratio = 0
- **Maximum Regret Ratio (mrr)** = the maximum of all regret ratios (among all users) (e.g., 0.06173)



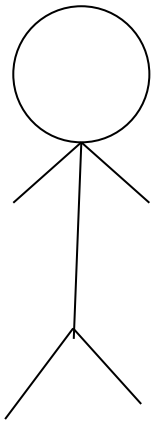
## k-regret queries

Suppose that user Raymond wants to buy an apartment

- Outline
  - K-regret Queries
  - Interactive k-regret Queries

## k-regret queries

Suppose that user Raymond wants to buy an apartment



Raymond

$p_3$

Which one is better?

1. $p_2$
2. $p_3$

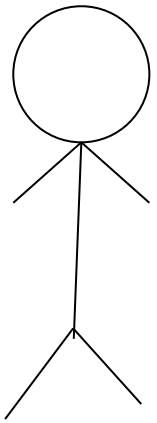
After this round, we understand Raymond's preference better.

D

| Apartment | $X_1$ | $X_2$ |
|---|---|---|
| $p_1$ | 0 | 1 |
| $p_2$ | 0.2 | 1 |
| $p_3$ | 0.6 | 0.9 |
| $p_4$ | 0.9 | 0.6 |
| $p_5$ | 1 | 0.2 |
| $p_6$ | 1 | 0 |
| … | … | … |

## k-regret queries

Suppose that user Raymond wants to buy an apartment



Raymond

$p_4$

Which one is better?

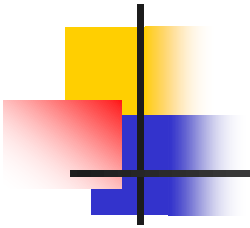
1. $p_4$
2. $p_5$

D

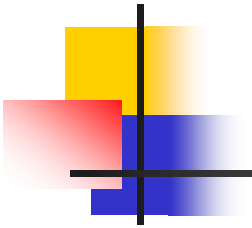
| Apartment | $X_1$ | $X_2$ |
|---|---|---|
| $p_1$ | 0 | 1 |
| $p_2$ | 0.2 | 1 |
| $p_3$ | 0.6 | 0.9 |
| $p_4$ | 0.9 | 0.6 |

After this round, we understand Raymond's preference much better.

With more rounds/questions, we could know Raymond's preference more.



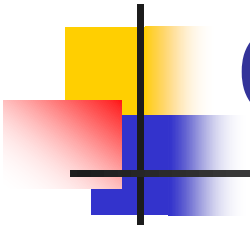
- We propose an optimization model to find the best question (containing two apartments) for each round
- Based on the feedback from the user, we could find the preference of the user.
- Based on this preference found, we could find the "best" apartment to the user.



- My previous work
  - Interactive Learning for Diverse Top-k Set (ICDE 2025)
  - Robust Best Point Selection under Unreliable User Feedback (VLDB 2024)
  - Reverse Regret Query (ICDE 2024) (Best ICDE 2024 Paper)
  - MixedSearch: An Interactive System of Searching for the Best Tuple with Mixed Attributes (ICDE 2024 (demo paper))
  - Finding Best Tuple via Error-prone User Interaction (ICDE 2023)
  - Interactive Mining with Ordered and Unordered Attributes (VLDB 2022)
  - Interactive Search for One of the Top-k (SIGMOD 2021)
  - A Fully Dynamic Algorithm for k-Regret Minimizing Sets (ICDE 2021)
  - Being Happy with the Least: Achieving α-happiness with Minimum Number of Tuple (ICDE 2020)
  - An Experimental Survey of Regret Minimization Query and Variants: Bridging the Best Worlds between Top-k Query and Skyline Query (VLDBJ 2020)
  - Strongly Truthful Interactive Regret Minimization (SIGMOD 2019)
  - FindYourFavorite: An Interactive System for Finding the User's Favorite Tuple in the Database (SIGMOD 2019 (demo paper))
  - Finding Average Regret Ratio Minimizing Set in Database (ICDE 2019)
  - Efficient k-Regret Query Algorithm with Restriction-free Bound for any Dimensionality (SIGMOD 2018)
  - k-Regret Minimizing Set: Efficient Algorithms and Hardness (ICDT 2017)
  - Minimizing Average Regret Ratio in Database (SIGMOD 2016 (Undergraduate Research Competition))
  - Geometry Approach for k-Regret Query (ICDE 2014)

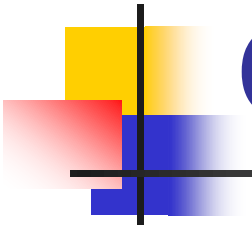
# Outline

- Topic
  - Data Understanding
  - Data Visualization
  - Recommendation System
  - User Feedback
- Conclusion



# Conclusion

- Data Understanding
- Data Visualization
- Recommendation System
- User Feedback



# Q&A

**Problem 1 (Interactive Regret):** Given a set D, we want to ask a number of questions to Raymond and return an apartment such that Raymond's regret ratio is at most $\varepsilon$.

**Problem 2 (Maximum Utility Point Determination):** Given a set D, we want to ask a number of questions to Raymond and return an apartment such that this apartment is Raymond's maximum utility point in D.

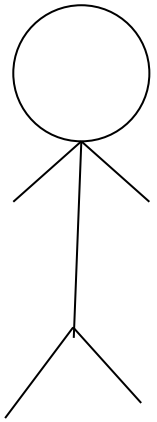
## k-regret queries

### Suppose that use

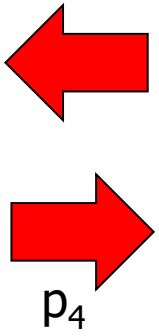
- Consider that Raymond
  (0.3, 0.7).
  But, we do not know th

D

| Apartment | $X_1$ | $X_2$ |
|---|---|---|
| $p_1$ | 0 | 1 |
| $p_2$ | 0.2 | 1 |
| $p_3$ | 0.6 | 0.9 |
| $p_4$ | 0.9 | 0.6 |



Raymond



$p_4$

Which one is better?

1. $p_4$
2. $p_5$

After this round, we understand Raymond's preference much better.

With more rounds/questions, we could know Raymond's preference more.