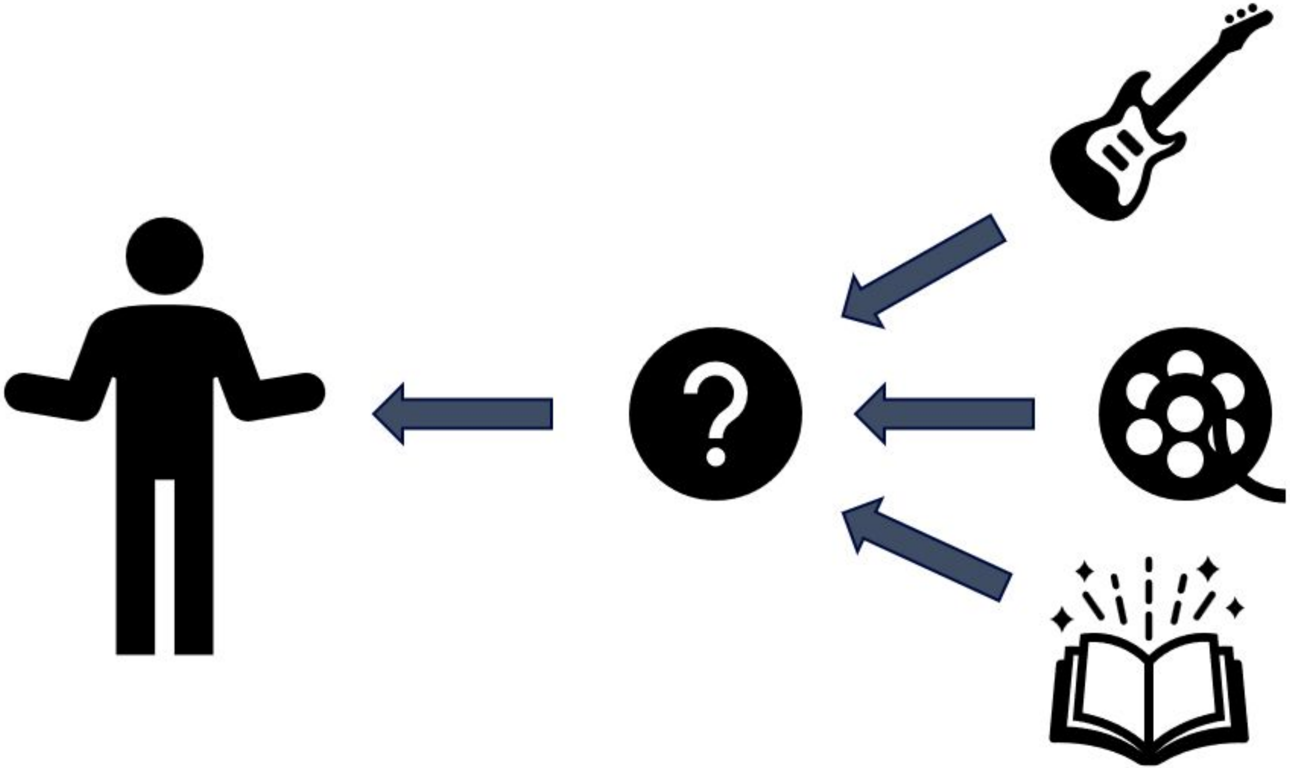
UC San Diego

# Data Mining:

## Item Recommendation Systems

Group 1
Joel Polizzi, Antariksha Ray, Lora Khatib, Jemin Vagadia
12/03/2024

# Introduction



# Overview:

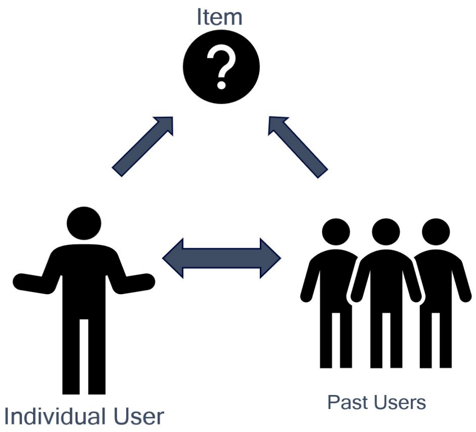
## Evaluating Amazon Data from SVD to BPR and BERT4Rec

- Singular Value Decomposition **(SVD)**
- Bayesian Personalization Ranking **(BPR)**
- Transformers for Recommendation **(Bert4Rec)**

# Overview:

## Foundational: Collaborative Filtering and Singular Value Decomposition (SVD)

- Addresses Collaborative Filtering challenges
- Implements Matrix factorization (SVD)
- Rating System




SVD

$$A = U\Sigma V^T$$

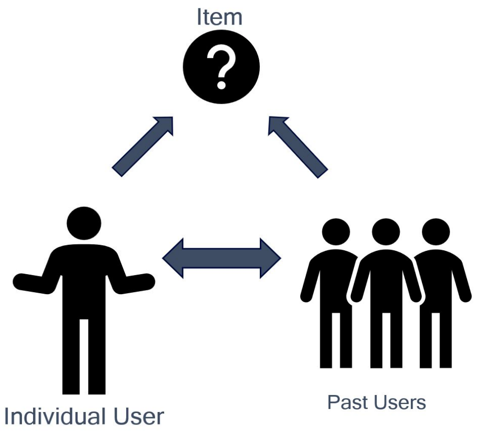
$$\underbrace{\left(\begin{bmatrix}\vdots\\u_1\\\vdots\end{bmatrix}\begin{bmatrix}\vdots\\u_2\\\vdots\end{bmatrix}\cdots\begin{bmatrix}\vdots\\u_i\\\vdots\end{bmatrix}\right)}_{U}\underbrace{\begin{pmatrix}\sigma_1&0&\cdots&0\\0&\sigma_2&\cdots&0\\0&0&\ddots&0\end{pmatrix}}_{\Sigma}\underbrace{\begin{pmatrix}[\cdots & v_1 & \cdots]\\ [\cdots & v_2 & \cdots]\\ & \vdots & \\ [\cdots & v_i & \cdots]\end{pmatrix}}_{V^T}$$

$$\sigma_i = \sqrt{\lambda_i}$$

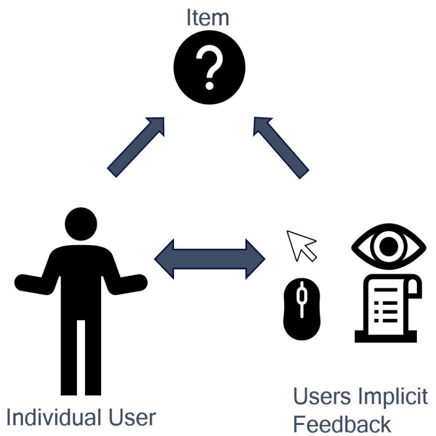
Overview:

# Foundational: Collaborative Filtering and Singular Value Decomposition (SVD)

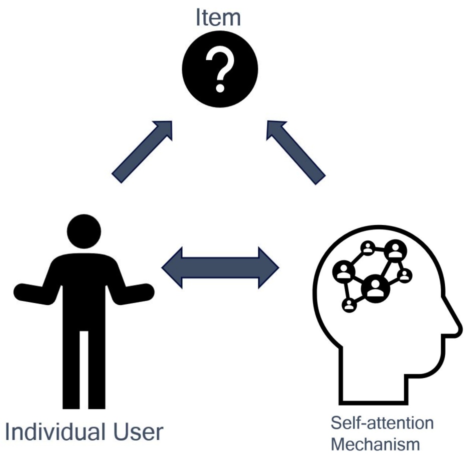
- Addresses Collaborative Filtering challenges
- Implements Matrix factorization (SVD)
- Rating System




SVD




BPR




BERT4Rec

# Importance of the topic

- Implicit feedback drives personalized recommendations
- Increase user experience
- Grow engagement between users and application
- Scalable systems
- Sales impact and ROI

- Streaming and media platforms (Netflix, Prime, Youtube, etc…)
- E-Commerce (Amazon, Netflix, Reverb, etc..)
- Social Media (Meta, X, BlueSky, etc…)
- Finance (Credit Karma, Etrade, etc..)
- Food delivery (DoorDash, GrubHub, etc...)

# Problem formulation

- Recommendation systems have evolved from ratings systems to personalized ranking ones.
- Improve personalization recommendation **(BPR)**
- Examine transformer based approach where user context is applied to the learning system **(BERT4Rec)**

$$User = U,\ Item = I;\ preferred\ item = i,\ non\text{-}preferred\ item = j;$$
$$user\ latent\ factors = p_u,\ item\ latent\ factors = q_i$$

$$\text{BPR with matrix factorization: } \hat{x}_{ui} = p_u^T q_i$$

$$\text{Difference between items: } \hat{x}_{uij} = \hat{x}_{ui} - \hat{x}_{uj}$$

$$\text{Loss Function:} \quad Max \sum_{(u,i,j)\in D} ln\sigma(\hat{x}_{u,i,j}) - \lambda||\theta||^2$$

# **Matrix Factorization:** Relation to SVD

$$R = \underbrace{\begin{bmatrix} 5 & \cdot & \cdot & 2 & 3 \\ \cdot & 4 & 1 & \cdot & \cdot \\ \cdot & 5 & 5 & 3 & \cdot \\ 5 & \cdot & 4 & \cdot & 4 \\ 1 & 1 & \cdot & 4 & 5 \end{bmatrix}}_{\text{items}} \quad C = \underbrace{\begin{bmatrix} 1 & \cdot & \cdot & 1 & 1 \\ \cdot & 1 & 1 & \cdot & \cdot \\ \cdot & 1 & 1 & 1 & \cdot \\ 1 & \cdot & 1 & \cdot & 1 \\ 1 & 1 & \cdot & 1 & 1 \end{bmatrix}}_{\text{items}} \Bigg\} \text{users.}$$

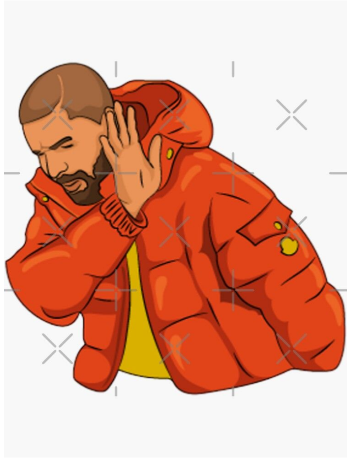
$$\underbrace{\begin{bmatrix} \\ \mathbf{R} \\ \\ \end{bmatrix}}_{|U|\times|I|} = \underbrace{\begin{bmatrix} \\ \gamma_U \\ \\ \end{bmatrix}}_{|U|\times K} \times \underbrace{\begin{bmatrix} \gamma_I^T \end{bmatrix}}_{K\times|I|}.$$

**Best rank-k approximation**

# Matrix Factorization: SVD in practice 1/2

- SVD is defined for fully observed matrices, which is often not the case for large user-item interaction data

- Even with data imputation, solving SVD on large matrices with millions of rows and columns is not recommended



# Matrix Factorization: SVD in practice 2/2

- Solving SVD on large matrices with million of rows and columns ?

It’s better to use gradient based approach

- Alternating Least Squares (ALS)
- Stochastic Gradient Descent (SGD)

# Matrix Factorization: SVD with SGD

- Solving SVD on large matrices with millions of rows and columns ❌

  It's better to use gradient based approach ✅
- Alternating Least Squares
- **Stochastic Gradient Descent**

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

$$\sum_{r_{ui} \in R_{train}} \left(r_{ui} - \hat{r}_{ui}\right)^2 + \lambda\left(b_i^2 + b_u^2 + ||q_i||^2 + ||p_u||^2\right)$$

Minimize the regularized squared error

$$\begin{aligned} b_u &\leftarrow b_u &+ \gamma(e_{ui} - \lambda b_u) \\ b_i &\leftarrow b_i &+ \gamma(e_{ui} - \lambda b_i) \\ p_u &\leftarrow p_u &+ \gamma(e_{ui} \cdot q_i - \lambda p_u) \\ q_i &\leftarrow q_i &+ \gamma(e_{ui} \cdot p_u - \lambda q_i) \end{aligned}$$

Gradient update equations

where $e_{ui} = r_{ui} - \hat{r}_{ui}$.

# Implicit feedback and ranking models (step up)

- A lot of the times we deal with binary outcomes
(e.g. click prediction on unseen articles, add to cart a new item, like or dislike a new video)

- Re-use regression value through a sigmoid function?
prob(+ve interaction) high, pro(-ve interaction) low ?

But these -ve interactions are what we want to recommend!

# **Bayesian Personalized Ranking** (Rendle et. al 2009)

- Generate ranked list of items for user u such that

$$x_{u,i,j} > 0 \rightarrow u \text{ prefers } i$$
$$x_{u,i,j} \leq 0 \rightarrow u \text{ prefers } j.$$

$$x_{u,i,j} = x_{u,i} - x_{u,j} = \gamma_u \cdot \gamma_i - \gamma_u \cdot \gamma_j.$$

$$p(i >_u j|\Theta) := \sigma(\hat{x}_{uij}(\Theta)), \textit{Where } \sigma(x) := \frac{1}{1+e^{-x}}$$

$$\boxed{Loss = -\sum_{(u,i,j)\in D} ln(p(i >_u j|\Theta))p(\Theta) = -(\sum_{(u,i,j)\in D} ln(\sigma(\hat{x}_{uij}) - \lambda_\Theta||\Theta||^2)}$$

Bayes Theorem

$$\boxed{\begin{gathered} P(A|B) = \frac{P(A)P(B|A)}{P(B)} \\ p(\Theta| >_u) \propto p(>_u |\Theta)p(\Theta) \end{gathered}}$$

Maximize

Theta = parameter vector (MF)
>u = latent structure for user u

# **Bayesian Personalized Ranking** (continued)

- The loss function is analogous to the AUC score

$$\text{AUC}(u) := \frac{1}{|I_u^+|\,|I \setminus I_u^+|} \sum_{i \in I_u^+} \sum_{j \in |I \setminus I_u^+|} \delta(\hat{x}_{uij} > 0)$$ (AUC per user)

$$\delta(x > 0) = H(x) := \begin{cases} 1, & x > 0 \\ 0, & \text{else} \end{cases}$$

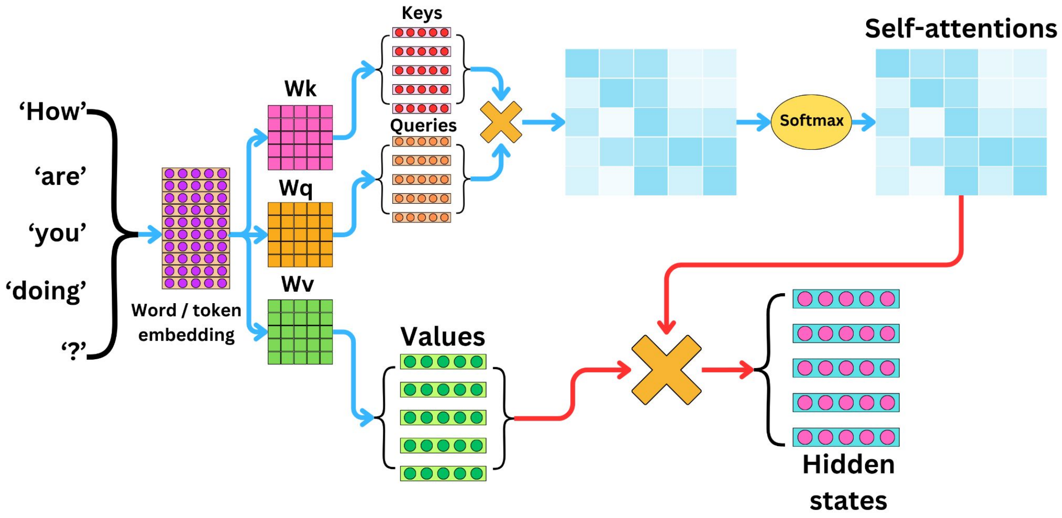
$$\text{AUC} := \frac{1}{|U|} \sum_{u \in U} AUC(u) \quad = \sum_{(u,i,j) \in D_S} z_u \, \delta(\hat{x}_{uij} > 0)$$ (Average AUC)

$$z_u = \frac{1}{|U|\,|I_u^+|\,|I \setminus I_u^+|}$$

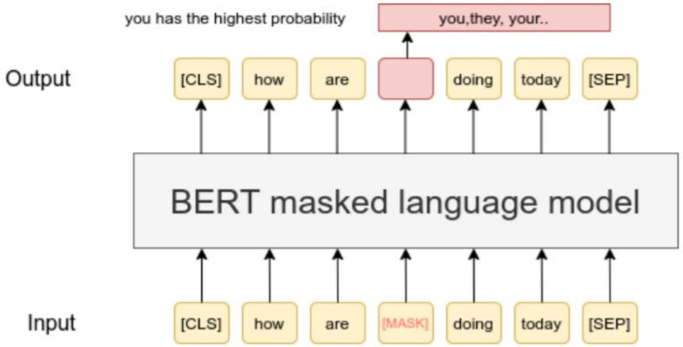
$$\boxed{Loss = -\sum_{(u,i,j) \in D} ln(p(i >_u j|\Theta))p(\Theta) = -(\sum_{(u,i,j) \in D} ln(\sigma(\hat{x}_{uij}) - \lambda_\Theta ||\Theta||^2)}$$

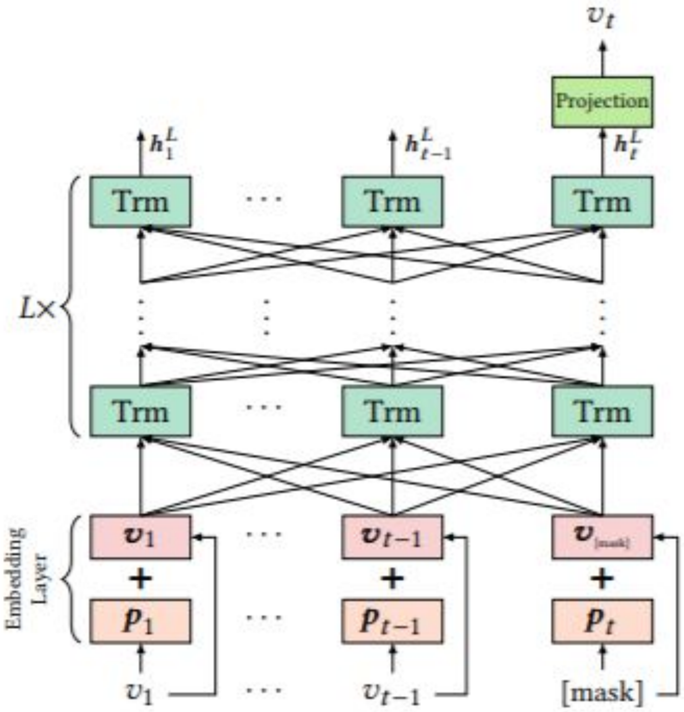
# Transformers Architecture

- Key innovation of the transformer architecture is the self-attention mechanism



# Bidirectional Encoder Representations from Transformers (BERT4Rec)




(b) BERT4Rec model architecture.

# Comparison of Approaches

| Aspect | SVD | BPR | BERT4Rec |
|---|---|---|---|
| **Model Type** | Matrix Factorization | Pairwise Learning-to-Rank | Transformer-based model |
| **Objective** | Minimize error in approximating user-item ratings (e.g., RMSE). | Maximize ranking of observed interactions over unobserved ones. | Predict items in a sequence. |
| **Training Data** | User-item interaction matrix | Implicit feedback | Sequential interaction data |
| **Strengths** | -simple<br>-efficient for dense data | -works well for implicit feedback<br>-scalable for large datasets | -Captures global dependencies<br>-Robust to noise |
| **Weaknesses** | -struggles with sparsity<br>-ignores sequential patterns | -ignores sequential patterns | -computationally expensive<br>-requires large training data |

# Experiment Setup

## Dataset

- We use the Amazon Reviews 2023 Dataset (link)
- It has ~572M reviews, **User reviews** (ratings, text, timestamp,...), **Item metadata** (description, price, imageurls,...)
- Interactions range from May 1996 to Sep 2023
- From several categories, we picked ***Musical_Instruments*** **(1.8M users, 213.6K items, 3.0M ratings)**

## Programming

python

jupyter

RecBole

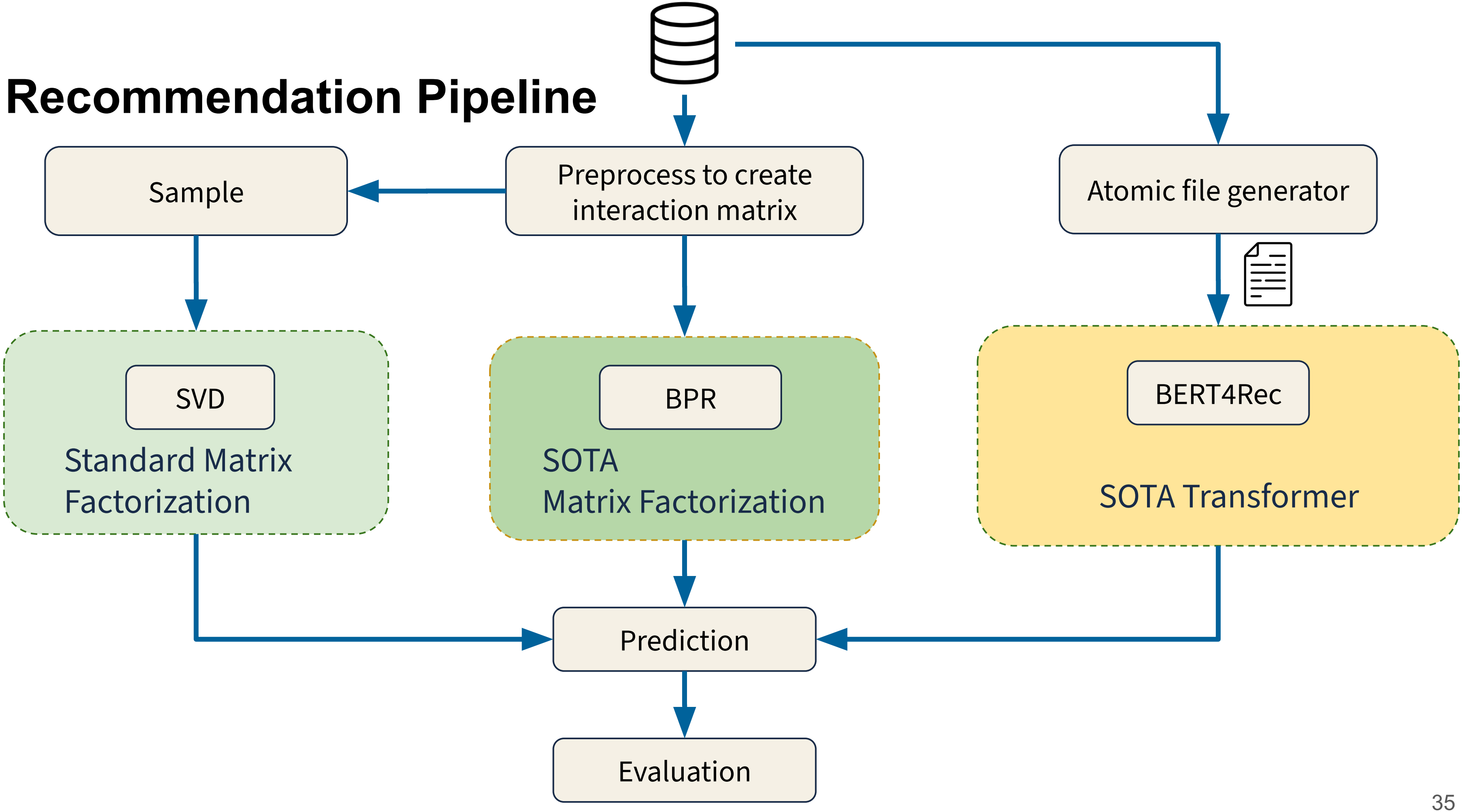
matplotlib

## Environment & Version Control

kaggle

colab

# Recommendation Pipeline





35

# Evaluation Metrics

- Precision@k: Measures the proportion of relevant items in the top-k recommendations.

- NDCG@K: Measures the overall reward at all positions (till K) that hold a relevant item. The reward is an inverse log of the position (i.e. higher ranks for relevant items would lead to better reward, as desired)

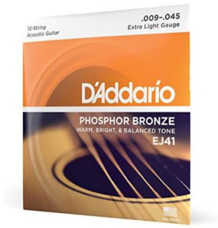
# Results and Interpretation

- Results on the proposed evaluation metrics

| Model | Precision@10 | NDCG@10 |
|---|---|---|
| BPR | 0.524 | 0.377 |
| BERT4Rec | 0.678 | 0.712 |

# Recommendations Visualized

User items


BPR

BERT4Rec


# Conclusions

- Ratings to rankings to sessions
- SVD forms the basis of our linear algebra approach
- Transformer models enable Deep Learning and Self-Attention
- Future: Generative Recommenders

# References


- Rendle, Steffen, et al. "BPR: Bayesian personalized ranking from implicit feedback." *arXiv preprint arXiv:1205.2618* (2012).

- Sun, Fei, et al. "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer." *Proceedings of the 28th ACM international conference on information and knowledge management*. 2019.

- de Souza Pereira Moreira, Gabriel, et al. "Transformers4rec: Bridging the gap between nlp and sequential/session-based recommendation." *Proceedings of the 15th ACM conference on recommender systems*. 2021.