A Tutorial on Wikimedia Visual Resources and its Application to Neural Visual Recommender Systems

Denis Parra[1], **Antonio Ossa-Guerra**[1], Manuel Cartagena[1], Patricio Cerda-Mardini[2], Felipe del Río[1], Isidora Palma[1], Diego Saez-Trumper[3], and Miriam Redi[3]

1. Pontificia Universidad Católica de Chile
2. MindsDB
3. Wikimedia Foundation

21st IEEE International Conference on Data Mining

ICDM 2021
IEEE International Conference on Data Mining
7 – 10 DECEMBER 2021
AUCKLAND NEW ZEALAND

VisRec Tutorial
Session 2: **Pipeline** + VisRank

Denis Parra[1], **Antonio Ossa-Guerra[1]**, Manuel Cartagena[1], Patricio Cerda-Mardini[2], Felipe del Río[1], Isidora Palma[1], Diego Saez-Trumper[3], and Miriam Redi[3]

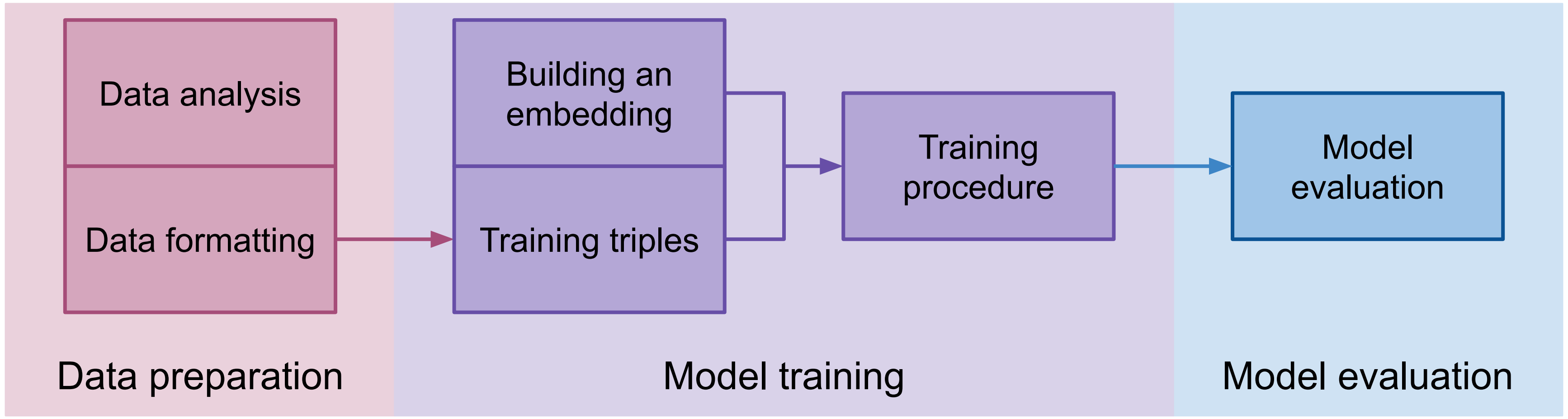
1. Pontificia Universidad Católica de Chile
2. MindsDB
3. Wikimedia Foundation

21st IEEE International Conference on Data Mining

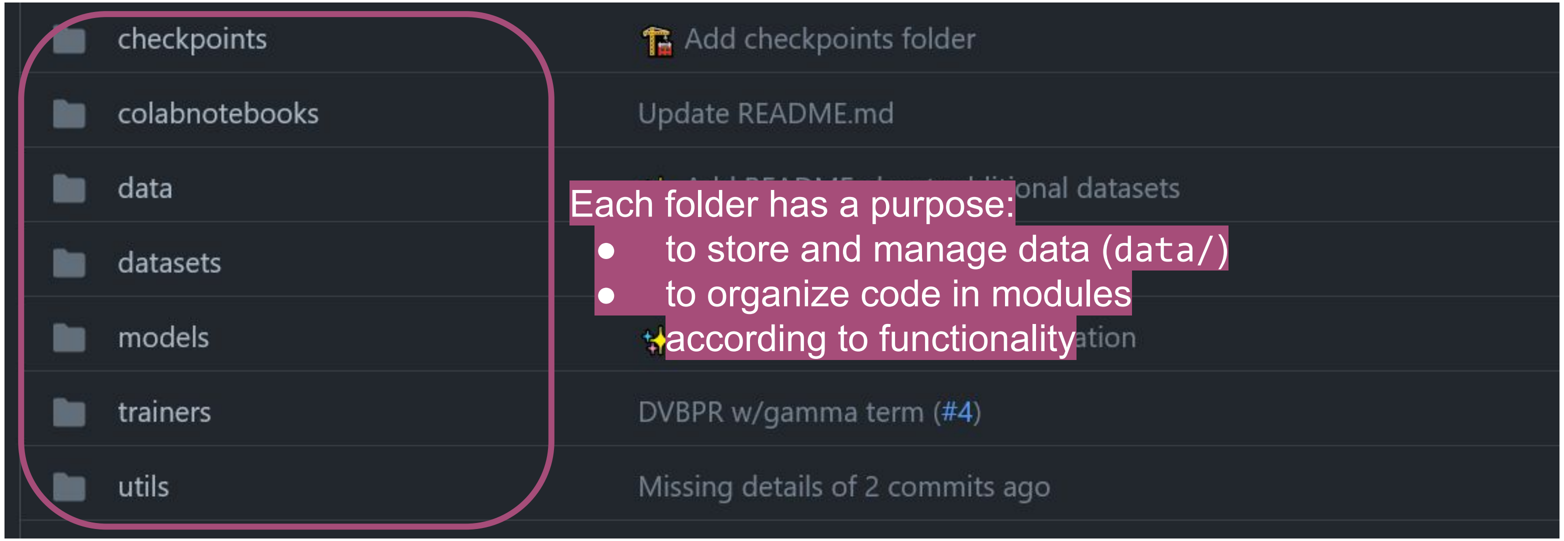
ICDM 2021
IEEE International Conference on Data Mining
7 – 10 DECEMBER 2021
AUCKLAND NEW ZEALAND

Table of Contents

Organization of our pipeline



Our repository

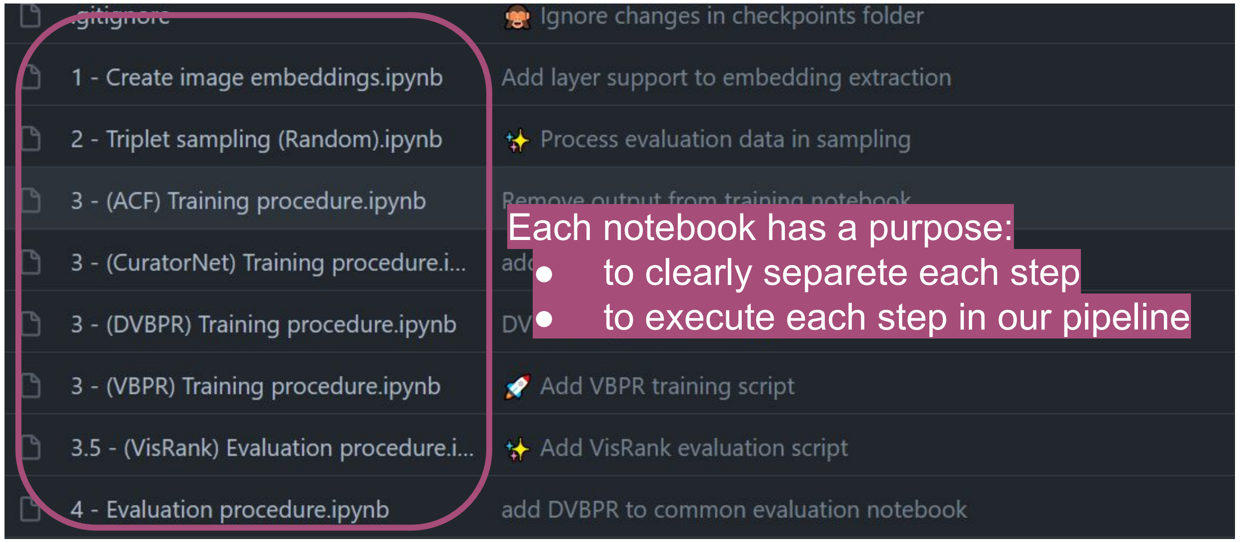


checkpoints	Add checkpoints folder
colabnotebooks	Update README.md
data	onal datasets
datasets	
models	ation
trainers	DVBPR w/gamma term (#4)
utils	Missing details of 2 commits ago

Each folder has a purpose:

- to store and manage data (`data/`)
- to organize code in modules according to functionality

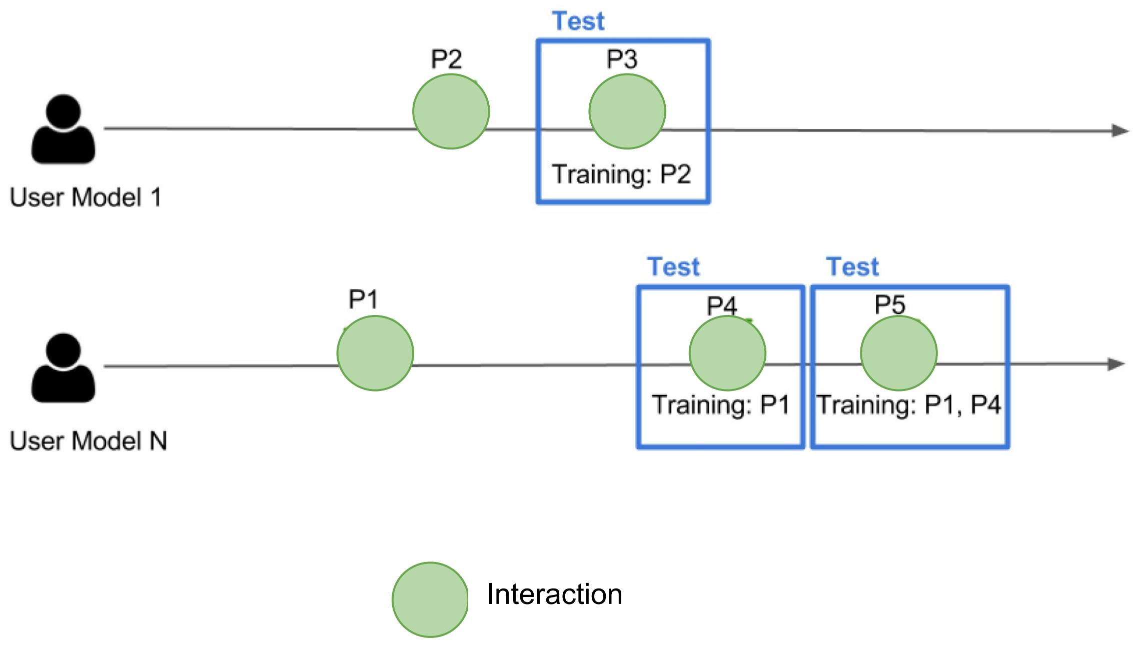
Our repository


.gitignore — 🙊 Ignore changes in checkpoints folder
1 - Create image embeddings.ipynb — Add layer support to embedding extraction
2 - Triplet sampling (Random).ipynb — ✨ Process evaluation data in sampling
3 - (ACF) Training procedure.ipynb — Remove output from training notebook
3 - (CuratorNet) Training procedure.i... — add
3 - (DVBPR) Training procedure.ipynb — DV
3 - (VBPR) Training procedure.ipynb — 🚀 Add VBPR training script
3.5 - (VisRank) Evaluation procedure.i... — ✨ Add VisRank evaluation script
4 - Evaluation procedure.ipynb — add DVBPR to common evaluation notebook

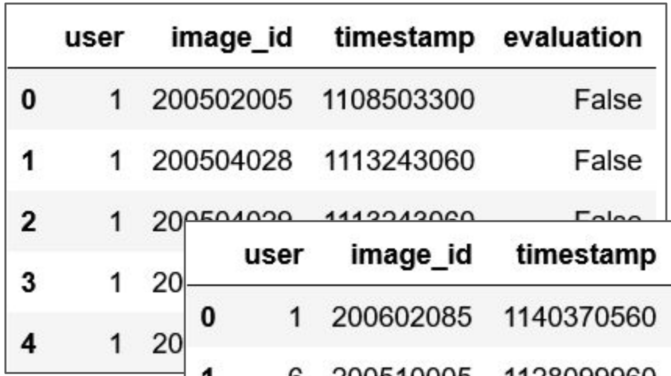
Each notebook has a purpose:

- to clearly separete each step
- to execute each step in our pipeline

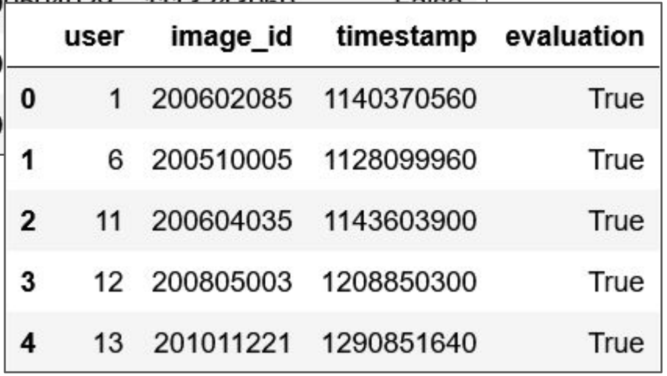
Data preparation: Data analysis



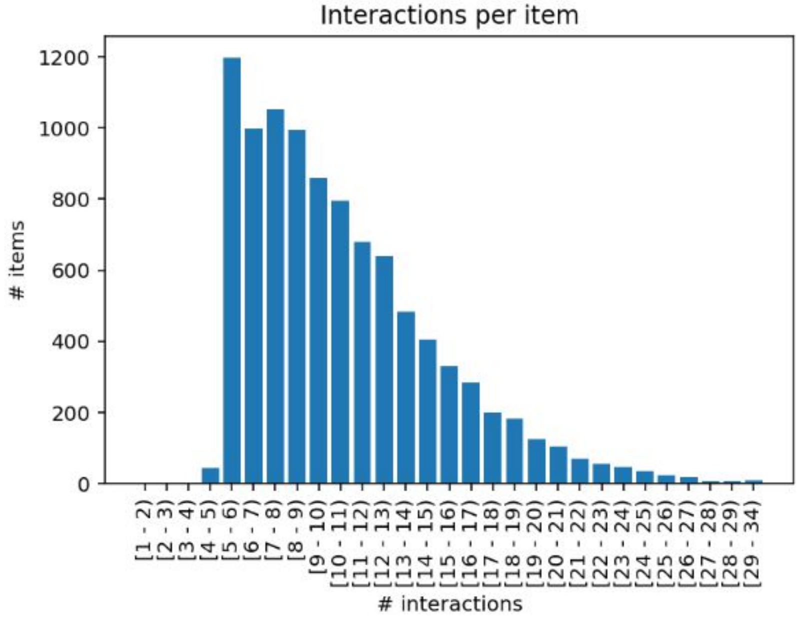
Data preparation: Data analysis



	user	image_id	timestamp	evaluation
0	1	200502005	1108503300	False
1	1	200504028	1113243060	False
2	1	20		
3	1	20		
4	1	20		



	user	image_id	timestamp	evaluation
0	1	200602085	1140370560	True
1	6	200510005	1128099960	True
2	11	200604035	1143603900	True
3	12	200805003	1208850300	True
4	13	201011221	1290851640	True

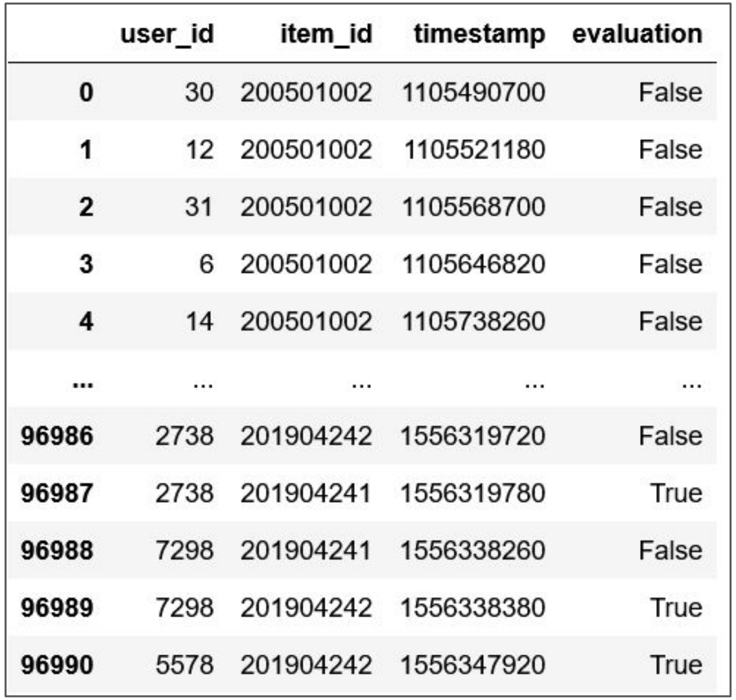


Data preparation: Data formatting

Key properties of format:

- Specific column names
- Interactions sorted by timestamp
- Column to identify evaluation rows

This structure is assumed further into the pipeline, so we must be consistent

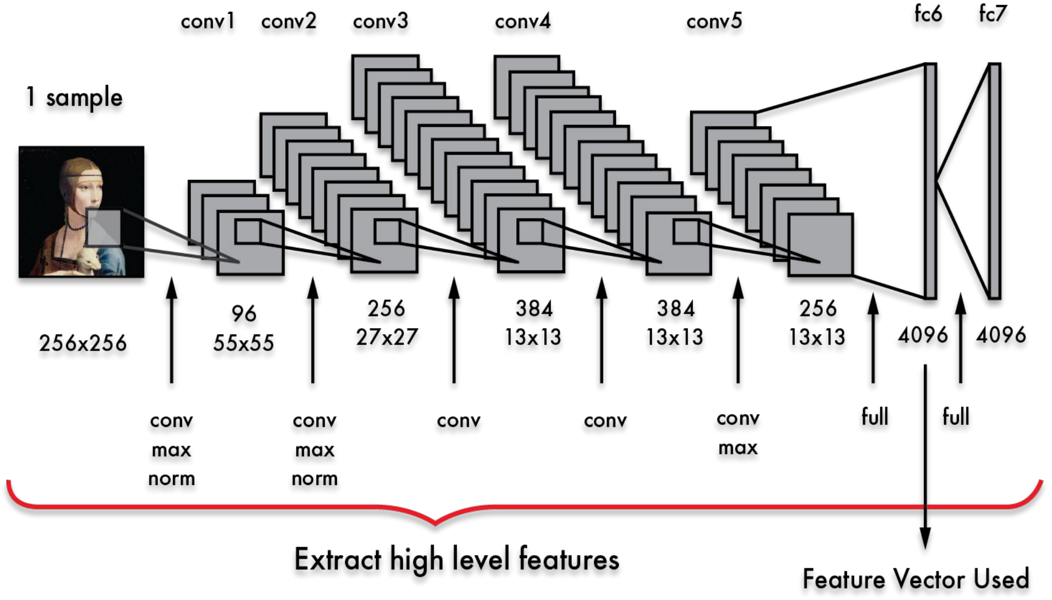


	user_id	item_id	timestamp	evaluation
0	30	200501002	1105490700	False
1	12	200501002	1105521180	False
2	31	200501002	1105568700	False
3	6	200501002	1105646820	False
4	14	200501002	1105738260	False
...	...	...	...	...
96986	2738	201904242	1556319720	False
96987	2738	201904241	1556319780	True
96988	7298	201904241	1556338260	False
96989	7298	201904242	1556338380	True
96990	5578	201904242	1556347920	True

Model training: Building an embedding

- Each image in the dataset is mapped to a latent feature vector
- We store the embeddings in a *.npy file to load it in our models

```
[
    ("item_id1", <vector>),
    ("item_id2", <vector>),
    ...
]
```



Model training: User rating vs BPR

Pointwise approach:

- Looks a **single item** at a time while training, trying to predict how relevant is it fr the current query
- Requires to know how relevant the item really is: **user rating**

$$r_{ui}$$

Pairwise approach:

- Looks a **pair of items** at a time, trying to learn what's the optimal ordering of said pair
- Just needs **implicit feedback** to infere the ordering

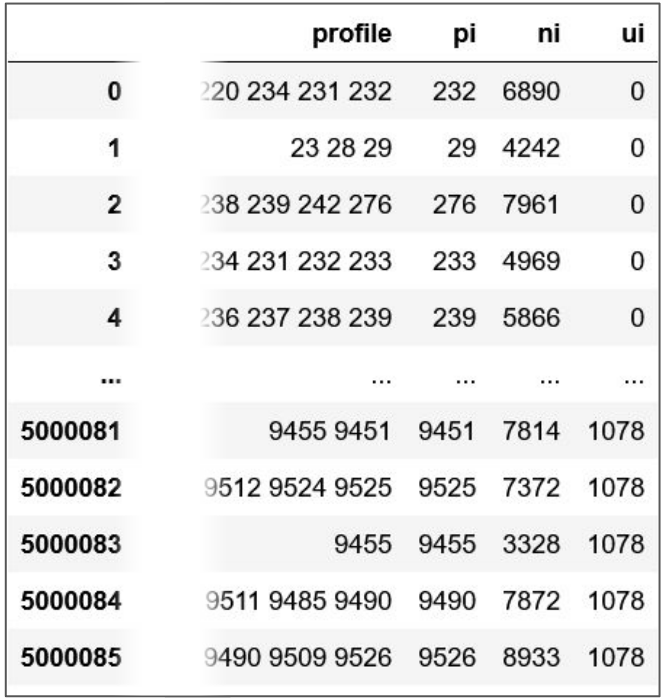
$$x_{uij} = x_{ui} - x_{uj}$$

Model training: Triples for training

The output of this stage contains:

- **pi** and **ni**: positive and negative items (index)
- **ui**: user identifier (index)
- **profile**: index of already consumed items

We generate 5 millions triples for training and 500k for validation



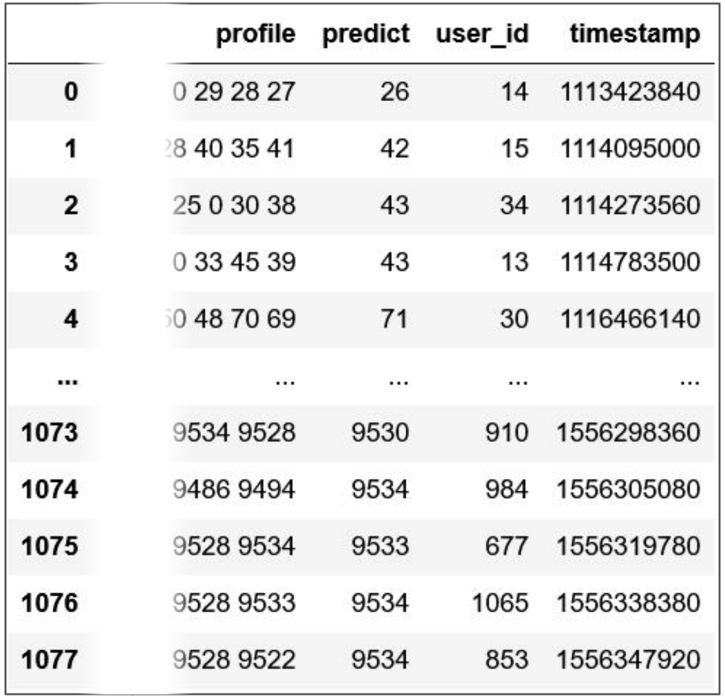
	profile	pi	ni	ui
0	220 234 231 232	232	6890	0
1	23 28 29	29	4242	0
2	238 239 242 276	276	7961	0
3	234 231 232 233	233	4969	0
4	236 237 238 239	239	5866	0
...	...	...	...	...
5000081	9455 9451	9451	7814	1078
5000082	9512 9524 9525	9525	7372	1078
5000083	9455	9455	3328	1078
5000084	9511 9485 9490	9490	7872	1078
5000085	9490 9509 9526	9526	8933	1078

Model training: Data for evaluation

The evaluation data contains:

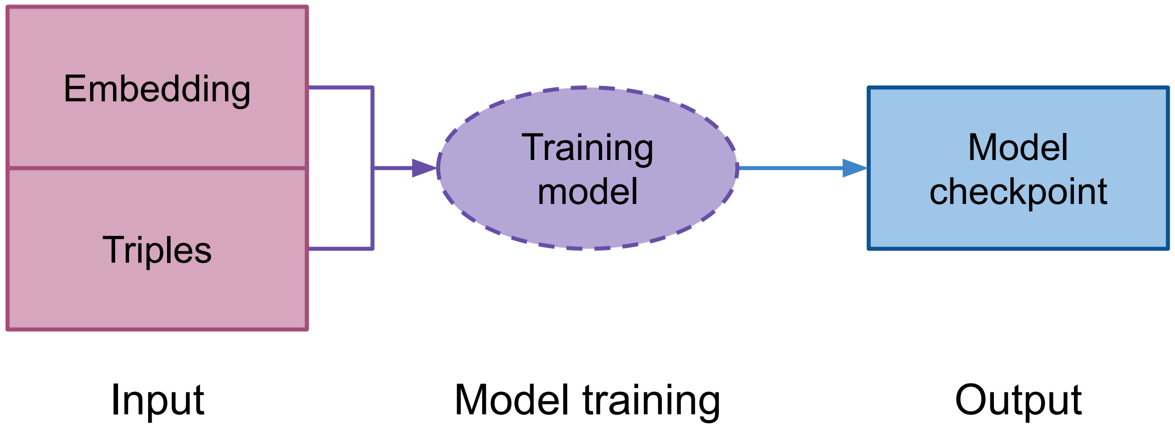
- **profile** and **predict**: already known interactions and ground truth (index)
- **user_id**: user identifier (index)
- **timestamp**: Time of ground truth interactions

We evaluate using the most recent interactions for each user in the dataset



	profile	predict	user_id	timestamp
0	0 29 28 27	26	14	1113423840
1	8 40 35 41	42	15	1114095000
2	25 0 30 38	43	34	1114273560
3	0 33 45 39	43	13	1114783500
4	0 48 70 69	71	30	1116466140
...	...	...	...	...
1073	9534 9528	9530	910	1556298360
1074	9486 9494	9534	984	1556305080
1075	9528 9534	9533	677	1556319780
1076	9528 9533	9534	1065	1556338380
1077	9528 9522	9534	853	1556347920

Model training: Training procedure



Differences in training and inference

Training

$$(user, item_i, item_j)$$

$$x_{uij} = x_{ui} - x_{uj}$$

Inference

$$(user, item)$$

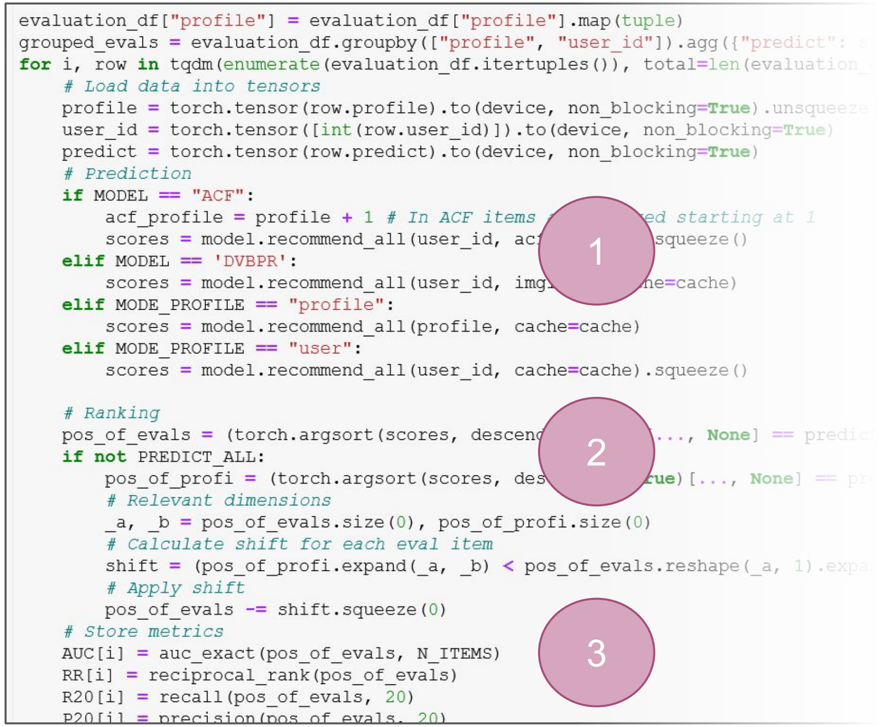
$$x_{ui}$$

Model evaluation

In this stage we only load a trained model checkpoint

Then, for every user:

1. Predict each item score
2. Sort items by score
3. Calculate metrics



```
evaluation_df["profile"] = evaluation_df["profile"].map(tuple)
grouped_evals = evaluation_df.groupby(["profile", "user_id"]).agg({"predict":
for i, row in tqdm(enumerate(evaluation_df.itertuples()), total=len(evaluation_
    # Load data into tensors
    profile = torch.tensor(row.profile).to(device, non_blocking=True).unsqueeze
    user_id = torch.tensor([int(row.user_id)]).to(device, non_blocking=True)
    predict = torch.tensor(row.predict).to(device, non_blocking=True)
    # Prediction
    if MODEL == "ACF":
        acf_profile = profile + 1 # In ACF items [illegible]ed starting at 1
        scores = model.recommend_all(user_id, ac[illegible]squeeze()
    elif MODEL == 'DVBPR':
        scores = model.recommend_all(user_id, img[illegible]he=cache)
    elif MODE_PROFILE == "profile":
        scores = model.recommend_all(profile, cache=cache)
    elif MODE_PROFILE == "user":
        scores = model.recommend_all(user_id, cache=cache).squeeze()

    # Ranking
    pos_of_evals = (torch.argsort(scores, descend[illegible]..., None] == predic
    if not PREDICT_ALL:
        pos_of_profi = (torch.argsort(scores, des[illegible]rue)[..., None] == pr
        # Relevant dimensions
        _a, _b = pos_of_evals.size(0), pos_of_profi.size(0)
        # Calculate shift for each eval item
        shift = (pos_of_profi.expand(_a, _b) < pos_of_evals.reshape(_a, 1).expa
        # Apply shift
        pos_of_evals -= shift.squeeze(0)
    # Store metrics
    AUC[i] = auc_exact(pos_of_evals, N_ITEMS)
    RR[i] = reciprocal_rank(pos_of_evals)
    R20[i] = recall(pos_of_evals, 20)
    P20[i] = precision(pos_of_evals, 20)
```

Tips and recommendations

In case that you want to use our pipeline:

- You’ll need a GPU (Google Colaboratory helps a lot!)
- Make sure to define an explicit criteria to choose evaluation rows
- Be consistent with the data format (ideally, it should not be different)
- Start with simple models to build your baseline
- If you add your own model implementation, follow the current structure
- Please read the README files in our repository

All of this recommendations will be available in the repository

VisRec Tutorial
Session 2: Pipeline + **VisRank**

Denis Parra[1], **Antonio Ossa-Guerra[1]**, Manuel Cartagena[1], Patricio Cerda-Mardini[2], Felipe del Río[1], Isidora Palma[1], Diego Saez-Trumper[3], and Miriam Redi[3]

1. Pontificia Universidad Católica de Chile
2. MindsDB
3. Wikimedia Foundation

21st IEEE International Conference on Data Mining

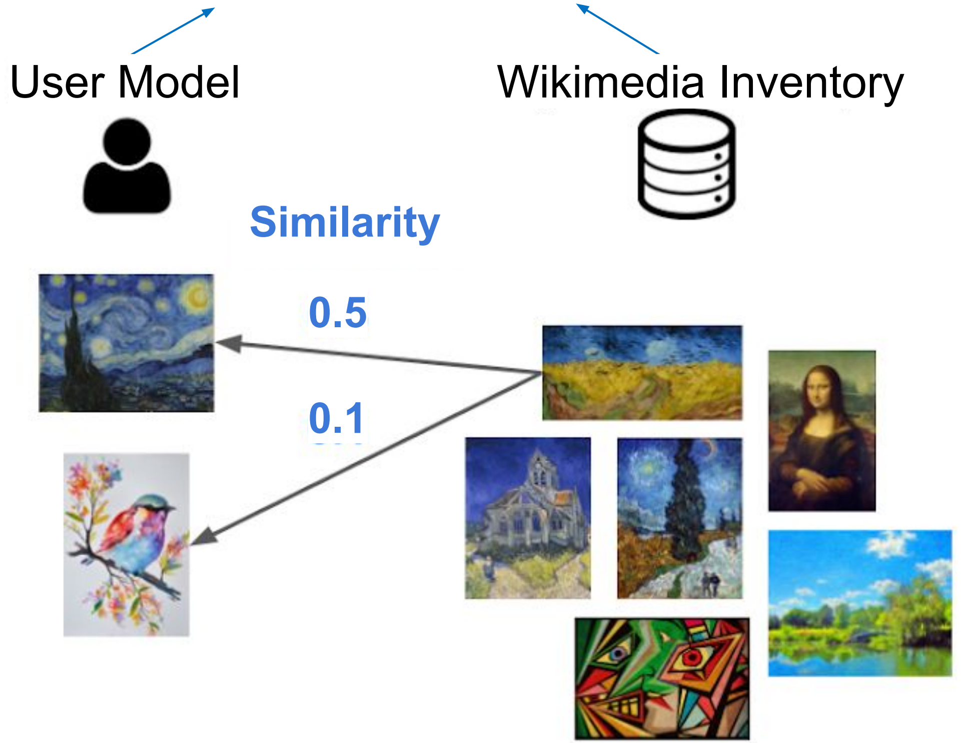
ICDM 2021
IEEE International Conference on Data Mining
7 – 10 DECEMBER 2021
AUCKLAND NEW ZEALAND

Context

- To define a simple and performant baseline model based on similarity
- “Nearest neighbor” style recommendation
- Images are ranked according to their average distance to user items

Content-based recommendation: VisRank (baseline)

s(u,i) = score(ContentBasedProfile(u), Content(i))



Mathematical formulation

Calculates similarity between the items as the cosine similarity between the vector representations

$$sim(V_i, V_j) = cos(V_i, V_j) = \frac{V_i \cdot V_j}{\|V_i\|\|V_j\|}$$

$$score(u,i)_X = \begin{cases} \max\limits_{j \epsilon P_u}\{sim(V_i^X, V_j^X)\} & (maximum) \\ \frac{\sum\limits_{j \in P_u} sim(V_i^X, V_j^X)}{|P_u|} & (average) \\ \frac{\sum\limits_{r=1}^{\min\{K,|P_u|\}} \max\limits_{j \epsilon P_u}{}^{(r)}\{sim(V_i^X, V_j^X)\}}{\min\{K,|P_u|\}} & (average\ top\ K) \end{cases}$$

Metrics for evaluation

AUC

$$AUC = \frac{1}{|\mathcal{U}|} \sum_{u} \frac{1}{|E(u)|} \sum_{(i,j) \in E(u)} \delta(\widehat{x}_{u,i} > \widehat{x}_{u,j})$$

Mean Reciprocal Rank

$$\mathrm{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\mathrm{rank}_i}$$

Metrics for evaluation

Precision@K

$$P@k = \frac{1}{|U_r|} \sum_{u \in U_r} \left(\frac{1}{|T_u|} \sum_{t \in T_u} p@k(t) \right) \qquad p@k(t) = \frac{|r_t^k \cap R_t|}{k}$$

Recall@K

$$R@k = \frac{1}{|U_r|} \sum_{u \in U_r} \left(\frac{1}{|T_u|} \sum_{t \in T_u} r@k(t) \right) \qquad r@k(t) = \frac{|r_t^k \cap R_t|}{|R_t|}$$

nDCG@K

$$nD@k = \frac{1}{|U_r|} \sum_{u \in U_r} \left(\frac{1}{|T_u|} \sum_{t \in T_u} \frac{DCG@k(t)}{iDCG@k(t)} \right) \qquad DCG@k(t) = \sum_{z=1}^{k} \frac{2^{B_t(i_{t,z})} - 1}{log_2(1+z)}$$

Main Results

AUC	RR	R@20	P@20	nDCG@20	R@100	P@100	nDCG@100
0.59216	0.01138	0.01881	0.00111	0.01274	0.03280	0.00039	0.01534

- Reasonable result in AUC, better than random (0.5)

- Good MRR and nDCG values (we'll see other models later)

References

[1] He, R., & McAuley, J. (2016, February). VBPR: visual bayesian personalized ranking from implicit feedback. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 30, No. 1).

[2] Messina, P., Dominguez, V., Parra, D., Trattner, C., & Soto, A. (2019). Content-based artwork recommendation: integrating painting metadata with neural and manually-engineered visual features. User Modeling and User-Adapted Interaction, 29(2), 251-290.

[3] Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618.

VisRec Tutorial
Session 2: Pipeline + VisRank

Denis Parra[1], **Antonio Ossa-Guerra[1]**, Manuel Cartagena[1], Patricio Cerda-Mardini[2], Felipe del Río[1], Isidora Palma[1], Diego Saez-Trumper[3], and Miriam Redi[3]

1. Pontificia Universidad Católica de Chile
2. MindsDB
3. Wikimedia Foundation

21st IEEE International Conference on Data Mining

ICDM 2021
IEEE International Conference on Data Mining
7 – 10 DECEMBER 2021
AUCKLAND NEW ZEALAND

A Tutorial on Wikimedia Visual Resources and its Application to Neural Visual Recommender Systems

Denis Parra[1], **Antonio Ossa-Guerra[1]**, Manuel Cartagena[1], Patricio Cerda-Mardini[2], Felipe del Río[1], Isidora Palma[1], Diego Saez-Trumper[3], and Miriam Redi[3]

1. Pontificia Universidad Católica de Chile
2. MindsDB
3. Wikimedia Foundation

21st IEEE International Conference on Data Mining

ICDM 2021
IEEE International Conference on Data Mining
7 – 10 DECEMBER 2021
AUCKLAND NEW ZEALAND

Thank you!

Antonio Ossa-Guerra [aaossa@uc.cl]

Denis Parra[1], **Antonio Ossa-Guerra[1]**, Manuel Cartagena[1], Patricio Cerda-Mardini[2], Felipe del Río[1], Isidora Palma[1], Diego Saez-Trumper[3], and Miriam Redi[3]

1. Pontificia Universidad Católica de Chile
2. MindsDB
3. Wikimedia Foundation

21st IEEE International Conference on Data Mining

ICDM 2021
IEEE International Conference on Data Mining
7 – 10 DECEMBER 2021
AUCKLAND NEW ZEALAND