Stanford CS246: Relational Deep Learning

CS246: Mining Massive Datasets

Jure Leskovec, Stanford University

http://cs246.stanford.edu

Announcements

- **Homework 3 and Colab 6 due today (2/20)**
- **Colab 7 is released (due 2/27)**
- **Homework 4 will be released today (due 3/6)**
- Homework 2, Colab 3 &4 grades will be released today (2/20)

Stanford CS46: Relational Databases

CS246: Mining Massive Datasets

Jure Leskovec, Stanford University

http://cs246.stanford.edu

Deep Learning Revolution

Sequences

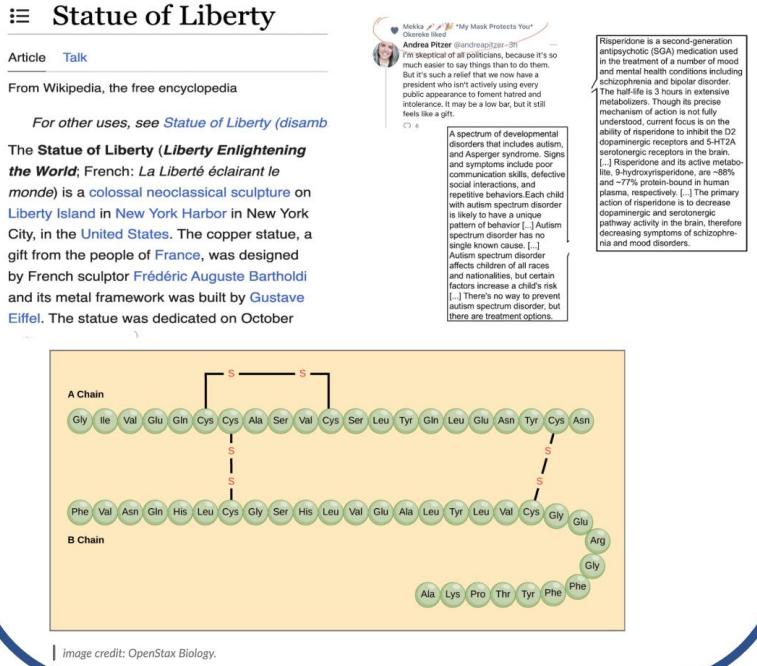
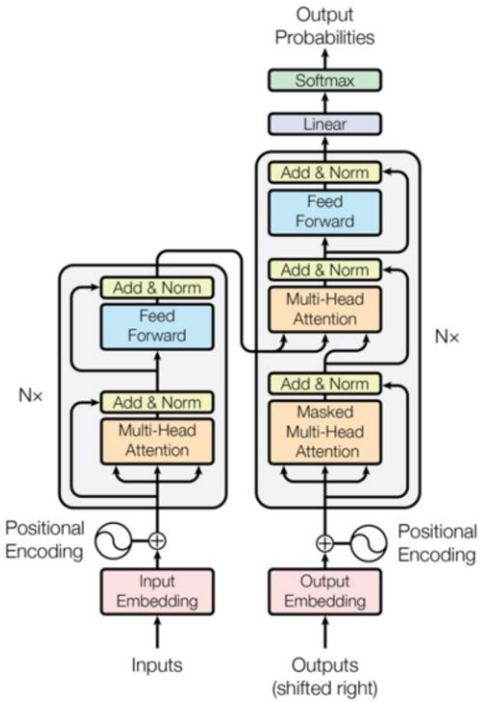


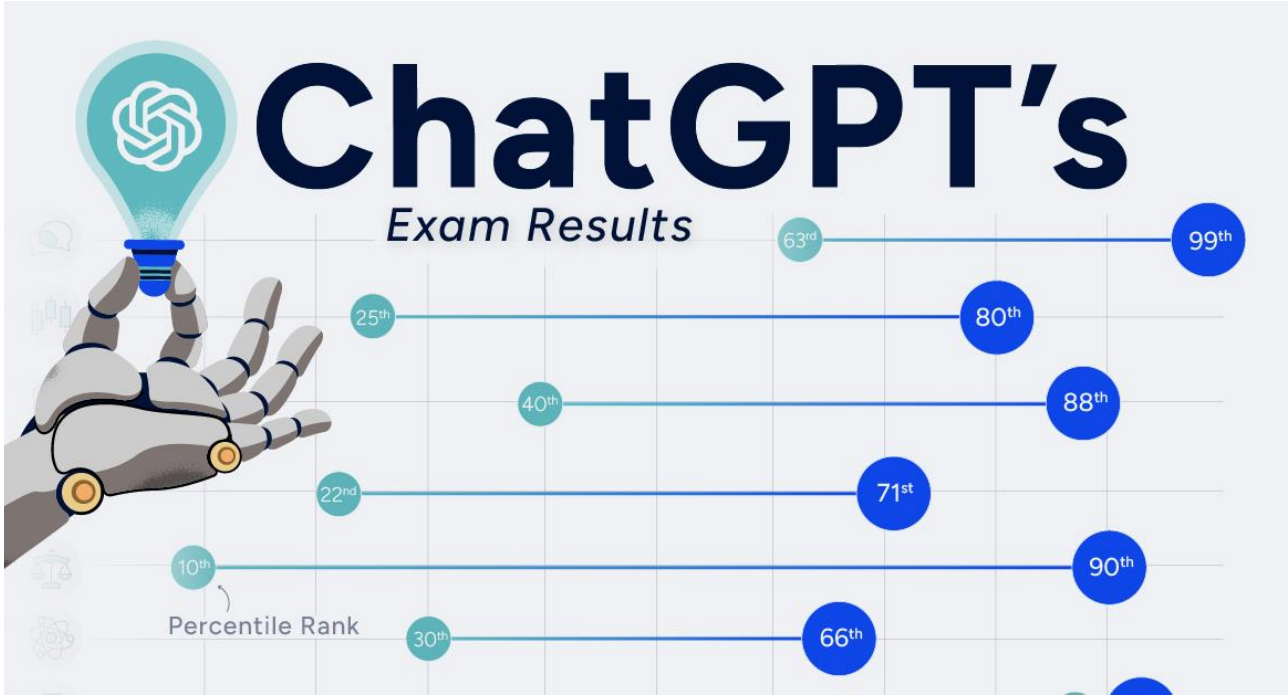
image credit: OpenStax Biology.

Images

Deep Learning Revolution

These breakthroughs are fueled by Transformers

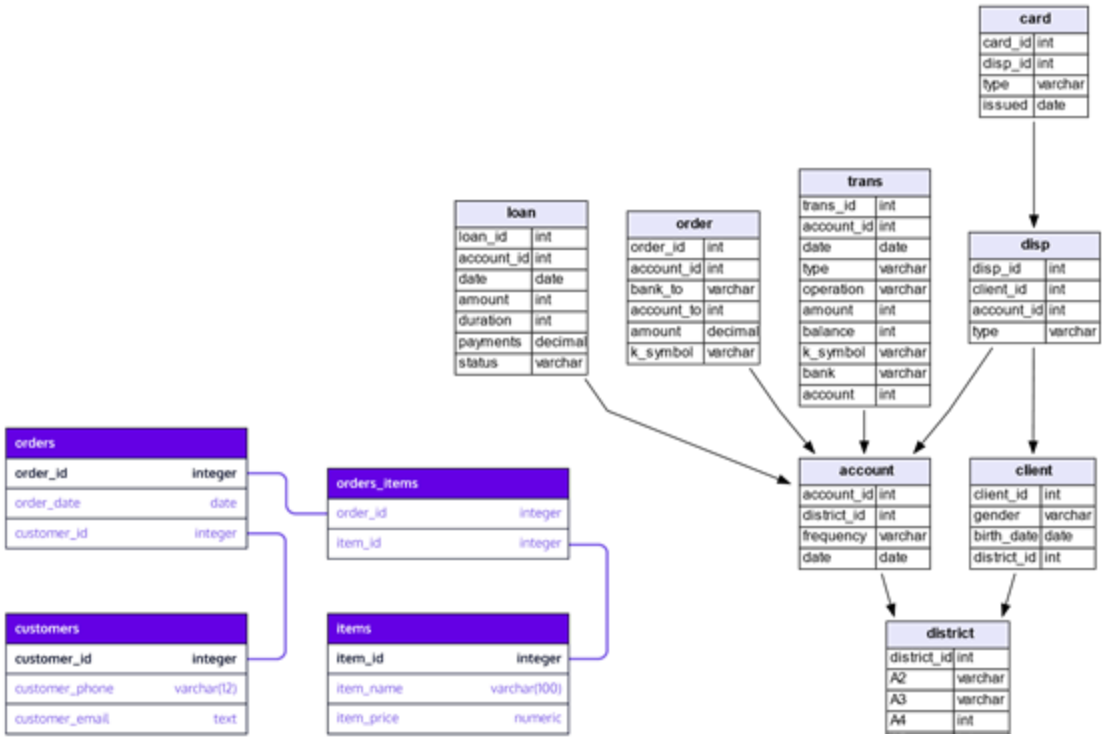


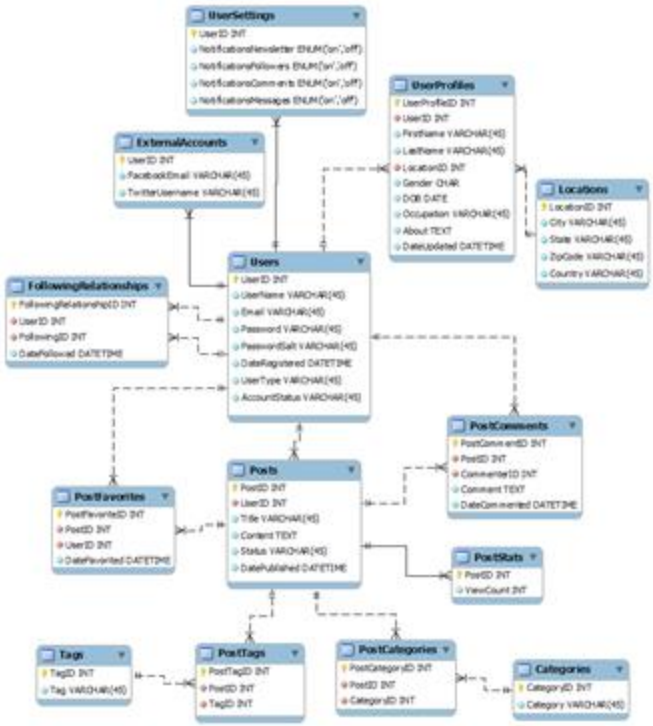


Deep Learning Revolution

These breakthroughs are fueled by Data

Data stored in Relational Databases



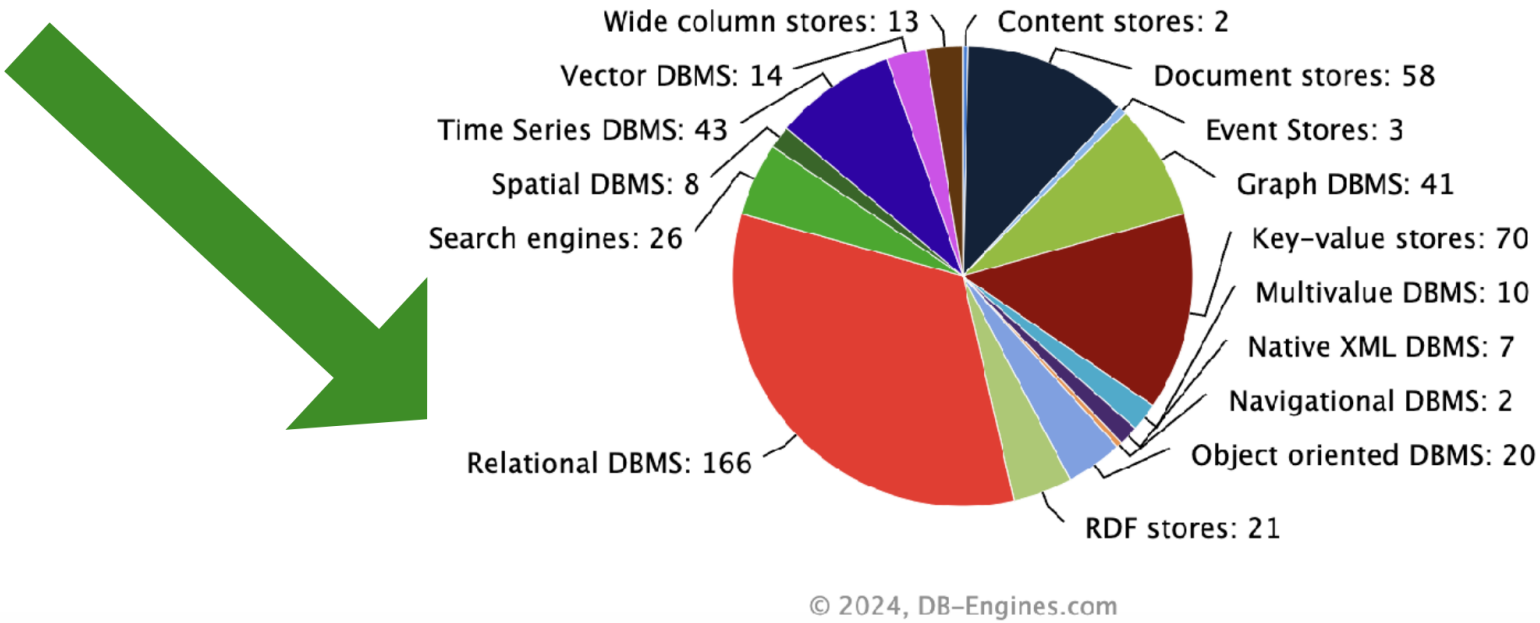


Commerce Finance Social Media

Database Management Systems

DBMS popularity broken down by database model

Number of systems per category, April 2024

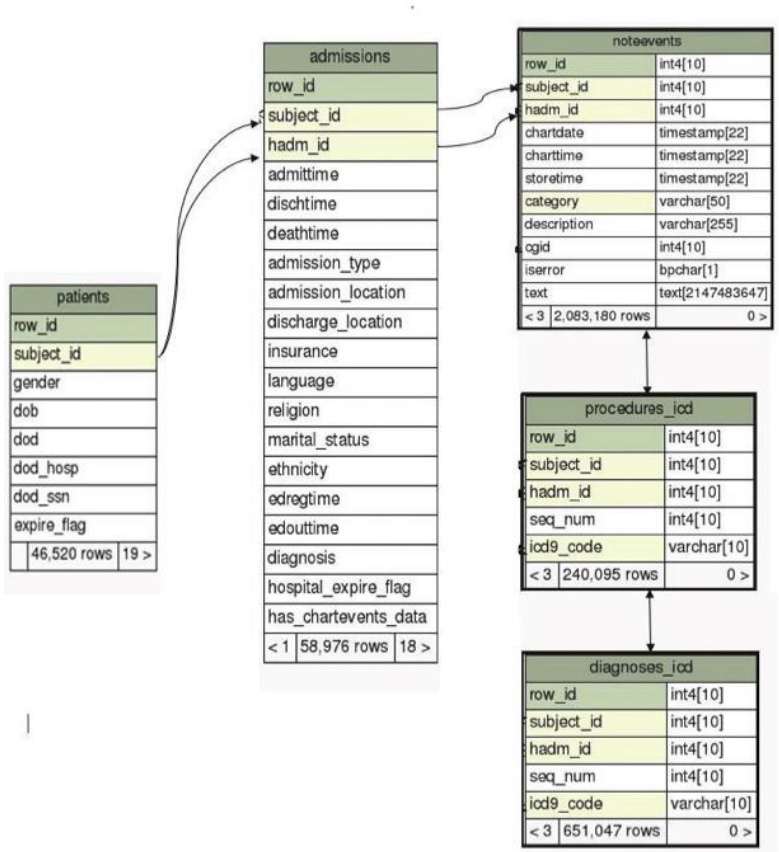


Predictions on Relational Data

- Which products will a user purchase in the next 7 days?
- Will an active user churn in the next 90 days?
- What will be the total sales for each product in the next 30 days?

Predictions on Relational Data

- Will a patient return if discharged from the hospital?
- Which hospital admissions have greatest risk to life in the next 24 hours

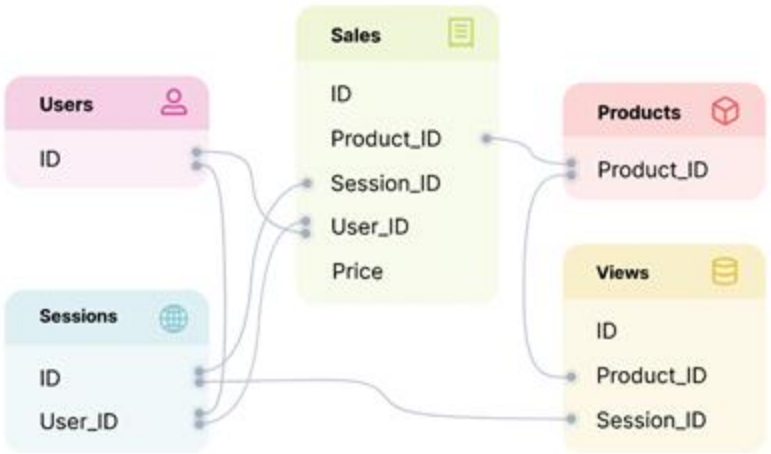


Doing AI is slow & complex

6-12 months of time for a team of data scientists, data engineers and product engineers

Impedance Mismatch

Goal: Learn a user churn model based on their sales, purchased products and browsing behavior



Examples

#monthly sales	sum weekly item price	#orders relative to Sunday	**Label**
…	…	…	…
…	…	…	…
…	…	…	…

Features

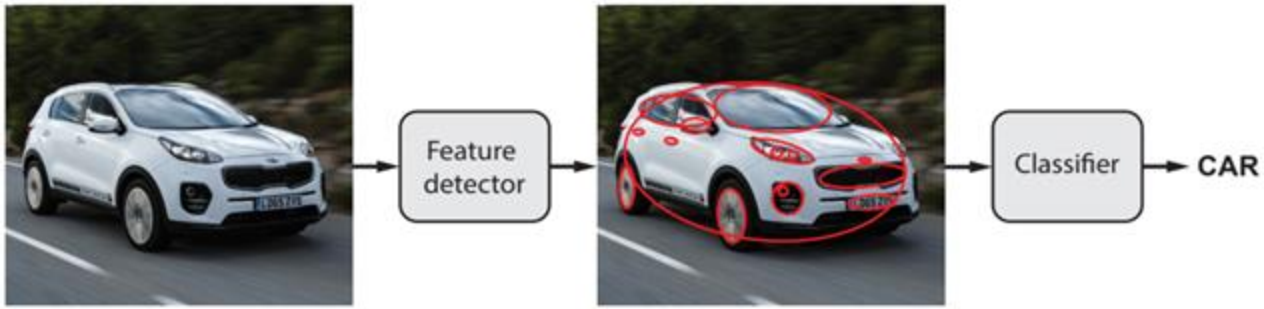
- Features are chosen arbitrarily (*e.g.*, aggregations, time windows)
- Only a limited set of data is used
- Issues with point-in-time correctness/information leakage

Key Question

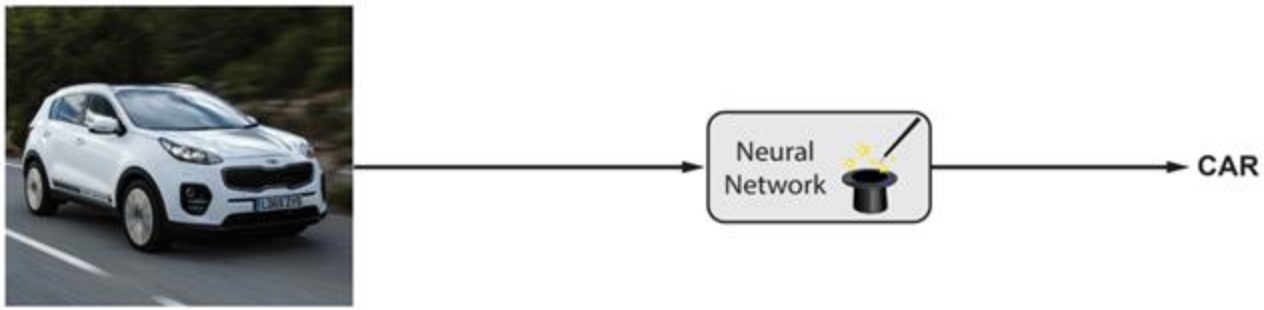
How to leverage the data without going through the longest duration tasks (extraction, transformation, loading)?

We want ML algorithms that can process data in its natural form!

Example from Computer Vision



Classical computer vision: hand-crafted features (e.g. SIFT)
+ simple classifier (e.g. SVM)



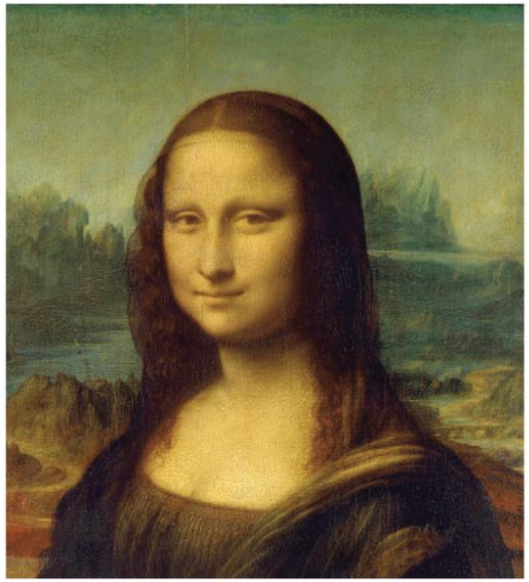
Modern computer vision: data-driven end-to-end systems

Why is ML stuck in the past?

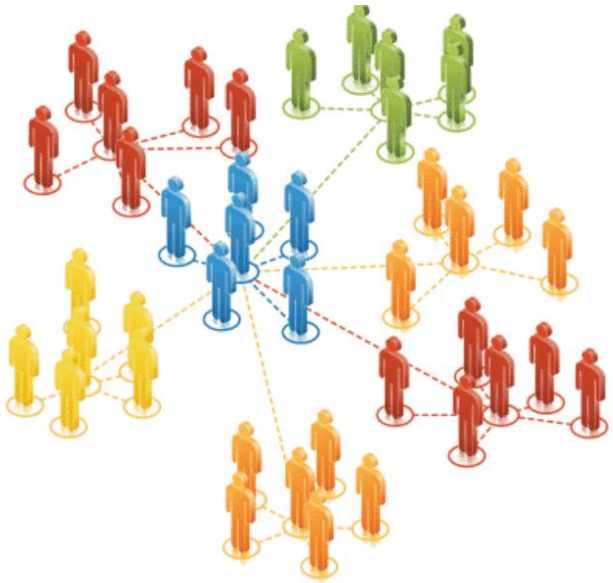
- Today we want to design deep learning models that operate on relational tables
- But modern deep learning toolbox is designed for **different type of inputs**

Doubt thou the stars are fire,
Doubt that the sun doth move,
Doubt truth to be a liar,
But never doubt I love...

Text



Images



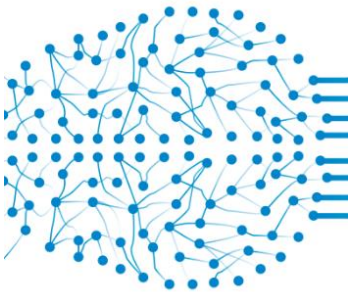
Stanford CS246: Generalizing Deep Learning to Databases

CS246: Mining Massive Datasets

Jure Leskovec, Stanford University

http://cs246.stanford.edu

Deep Learning on Relational Tables



Relational Deep Learning

Prediction

What is RDL?

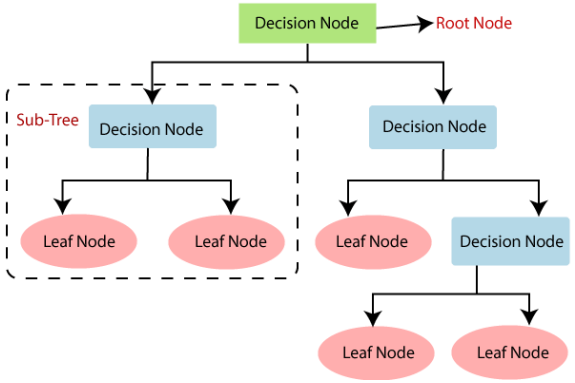
- End-to-end deep learning on relational tables (*i.e.*, databases)
- Works directly on relational tables, no transformations, no feature engineering
- Casts predictive tasks as **graph representation learning** problems

Impact and Consequences

- More accurate models
  (no feature engineering)
- More robust models
  (model-learned features automatically "update" over time)
- Shorter time to models
  (no mundane ETL work)
- Simpler infrastructure
  (no pipelines, no feature stores, etc.)

Related Work: Tabular ML

- Great for building models on **one table**
  - But most data is not in a single table!



- Deep learning is not dominant:
  - Hypothesis: Because single table already contains features engineered from other tables (loss of information)
- RDL accounts for relations between multiple tables, unlike tabular ML!

Stanford CS246: Relational Deep Learning

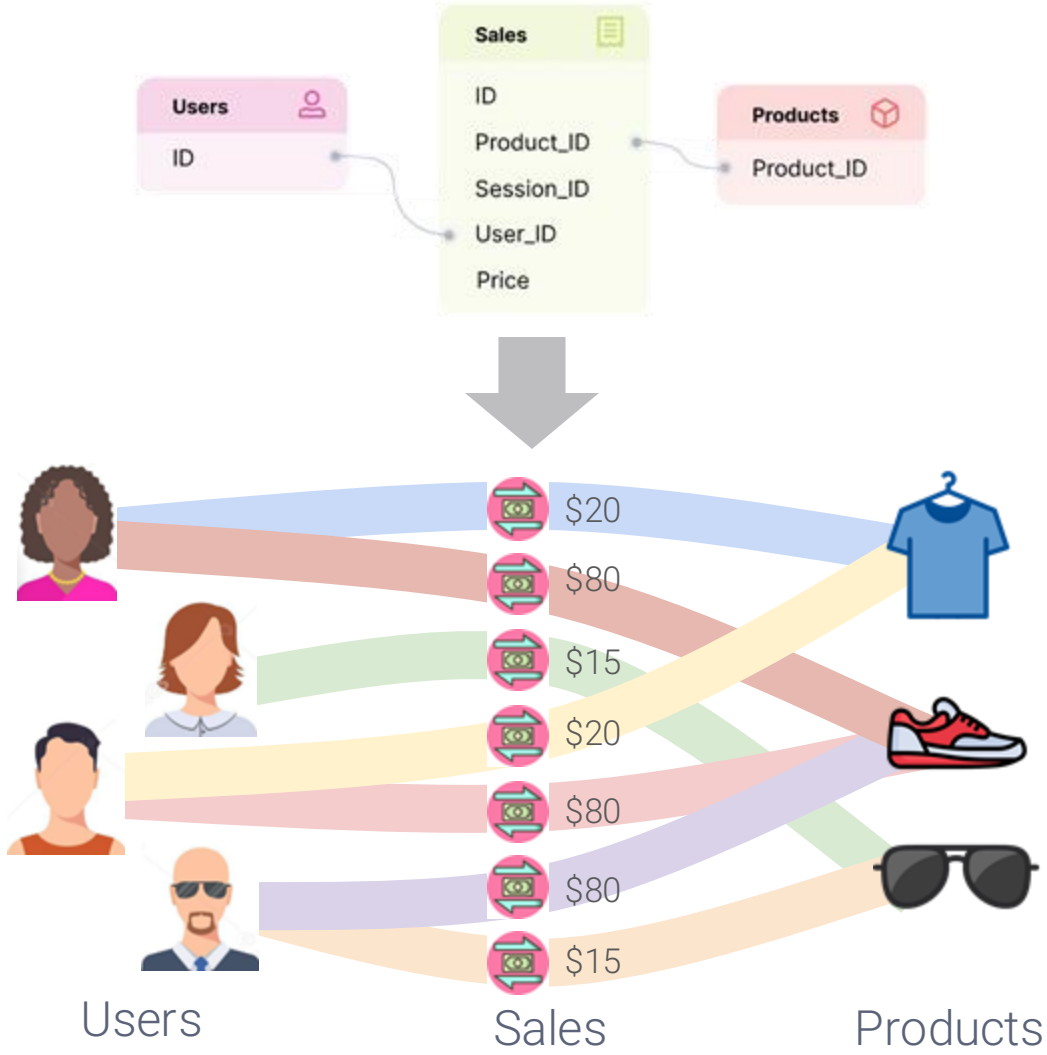
CS246: Mining Massive Datasets

Jure Leskovec, Stanford University

http://cs46.stanford.edu

Insight: A Data is a graph!

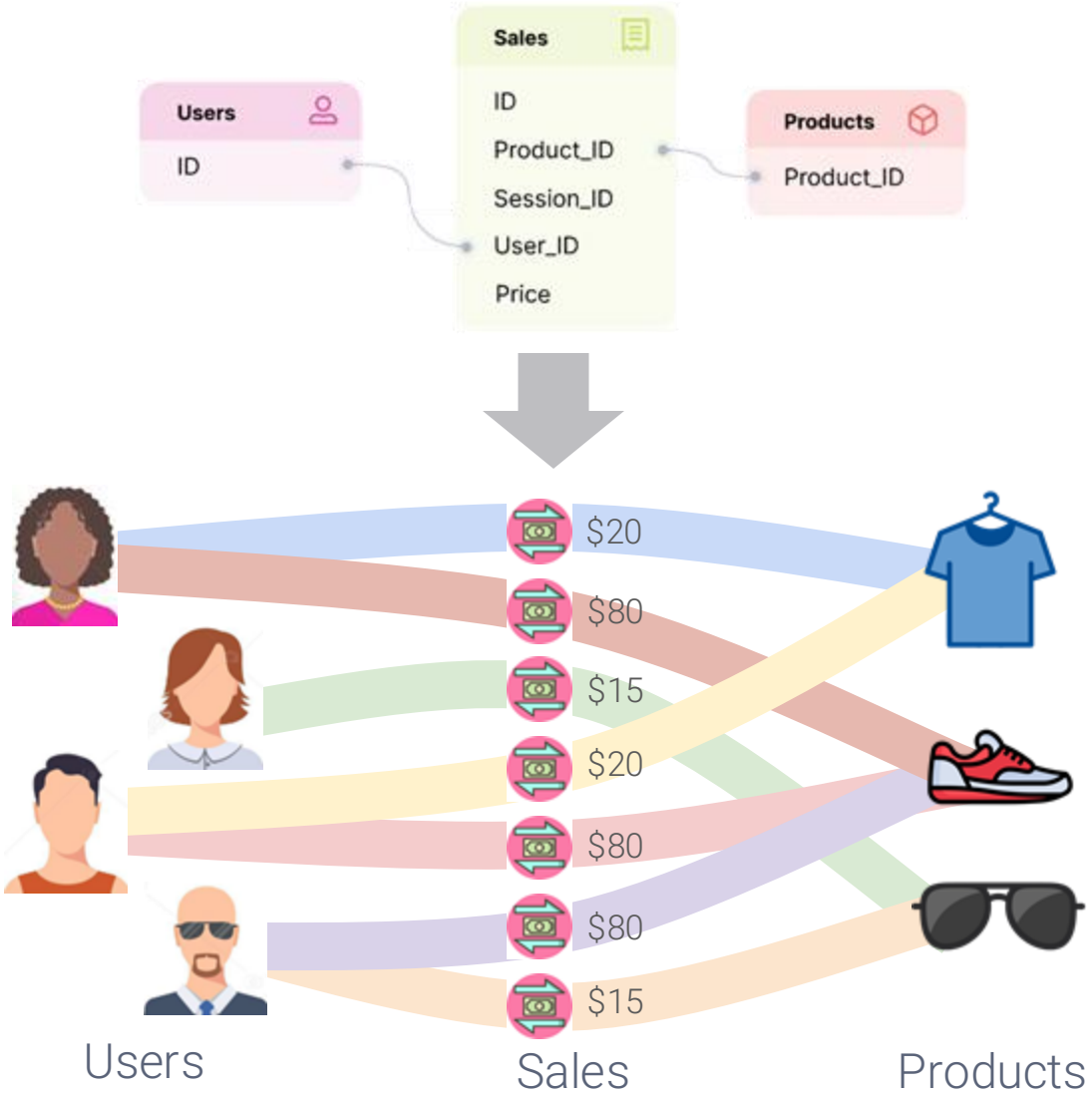
A Database is a graph!



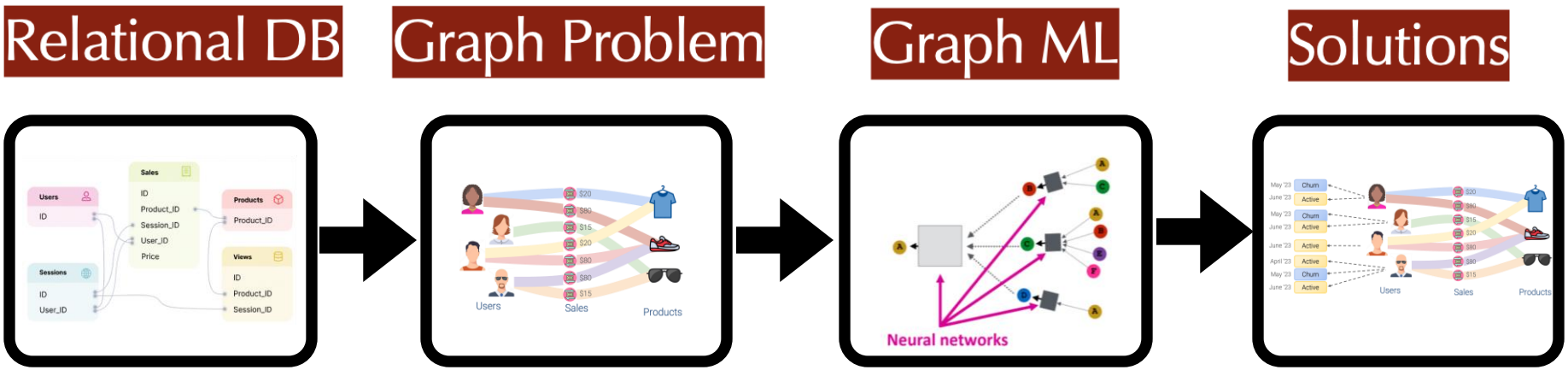
Just do ML on a Graph!

ML in the language of graphs:

- Node-level:
  - Churn
  - Life-time value
  - Next best action
- Link-level:
  - Product affinity
  - Recommendations
- Graph-level:
  - Fraud, money laundering



Graph ML Problem Solving Pipeline



Stanford CS246: Relational Database Graph

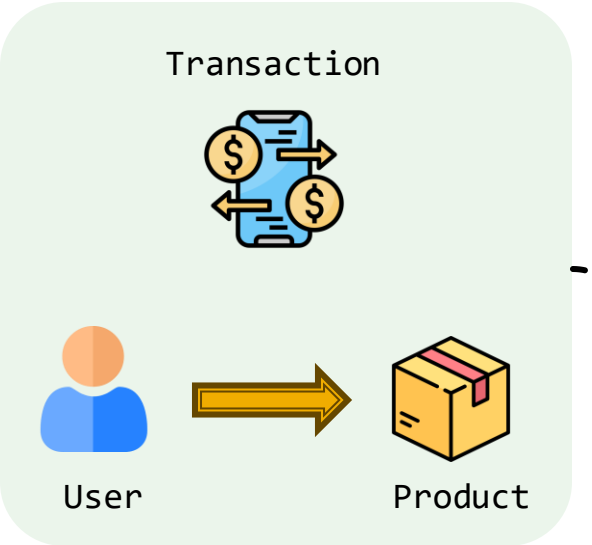
CS246: Mining Massive Datasets

Jure Leskovec, Stanford University

http://cs246.stanford.edu

Databases

- Real-world data are stored in databases

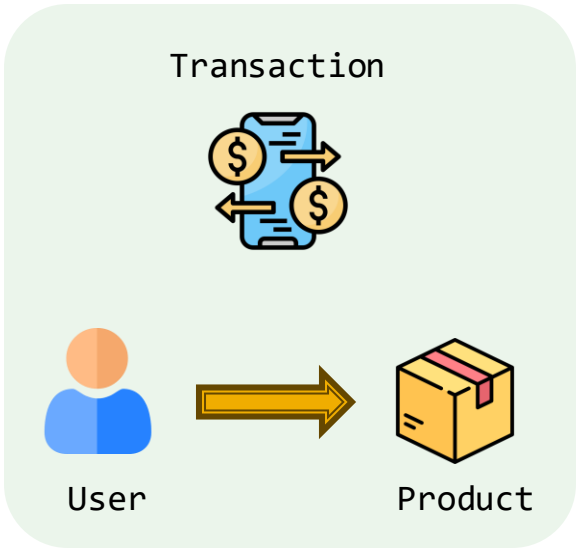


Transaction Table

Trans ID	User ID	Product ID	Value	Timestamp
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…
100	**51**	**345**	**$1000**	**1710281001**
…	…	…	…	…

Relational Databases

- Databases are often relational



Transaction Table

Trans ID	User ID	Product ID	Value	Timestamp
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…
100	51	345	$1000	1710281001
…	…	…	…	…

User Table

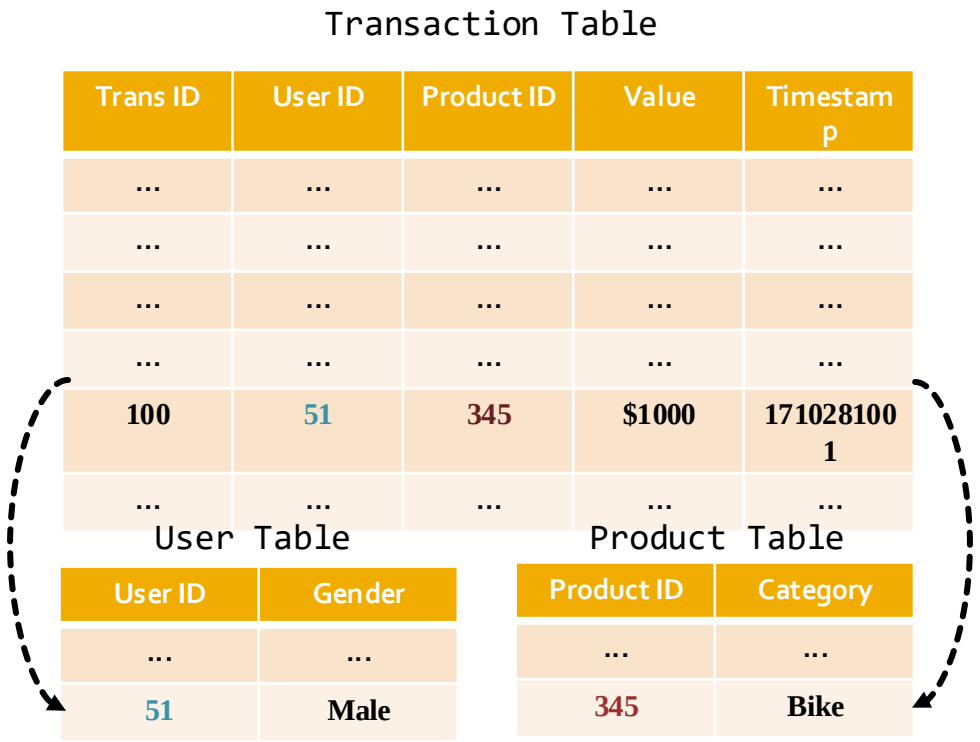
User ID	Gender
…	…
51	Male

Product Table

Product ID	Category
…	…
345	Bike

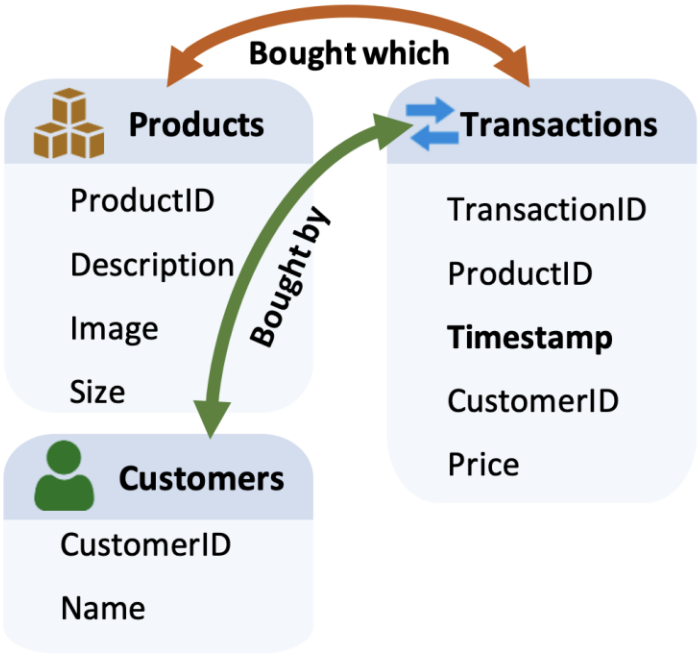
Relational databases as heterogeneous graphs

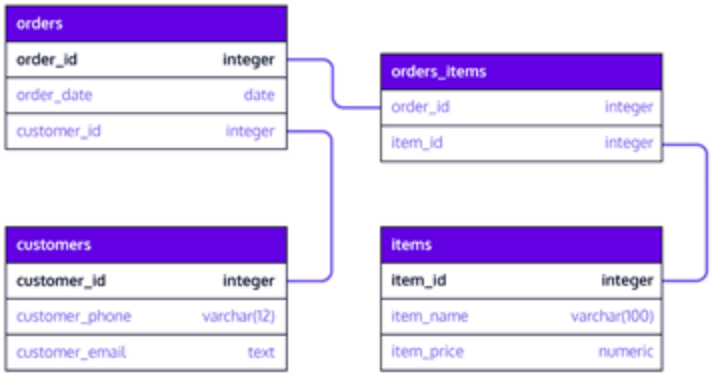
Mathematically…



- A database is a set of tables $\mathcal{T} = \{T_1, \ldots, T_n\}$ and
- Links between tables $\mathcal{L} \subseteq \mathcal{T} \times \mathcal{T}$

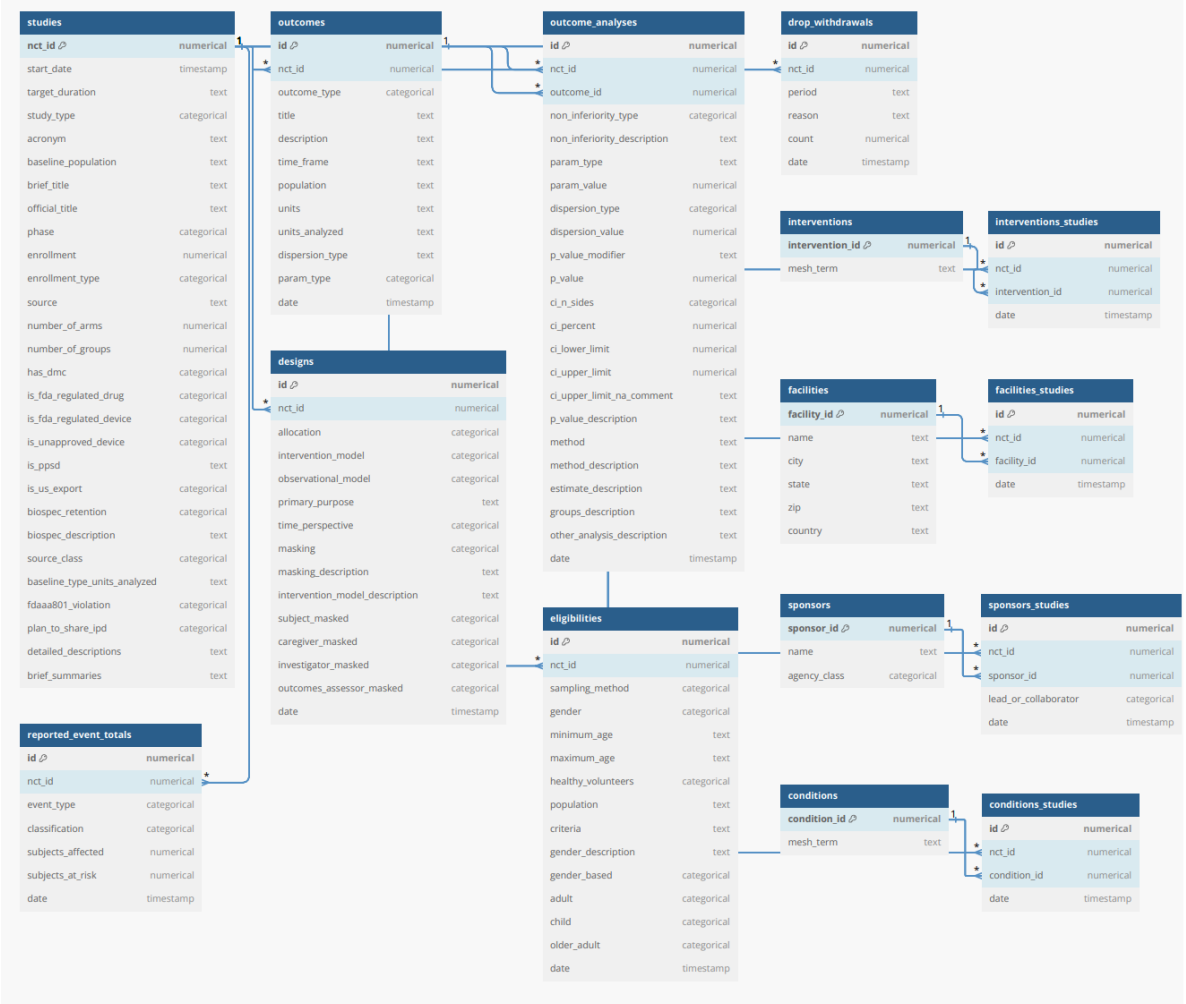
Schema Graph





- The schema graph represents the high-level structure of the heterogeneous graph

Rel-trial Schema Graph

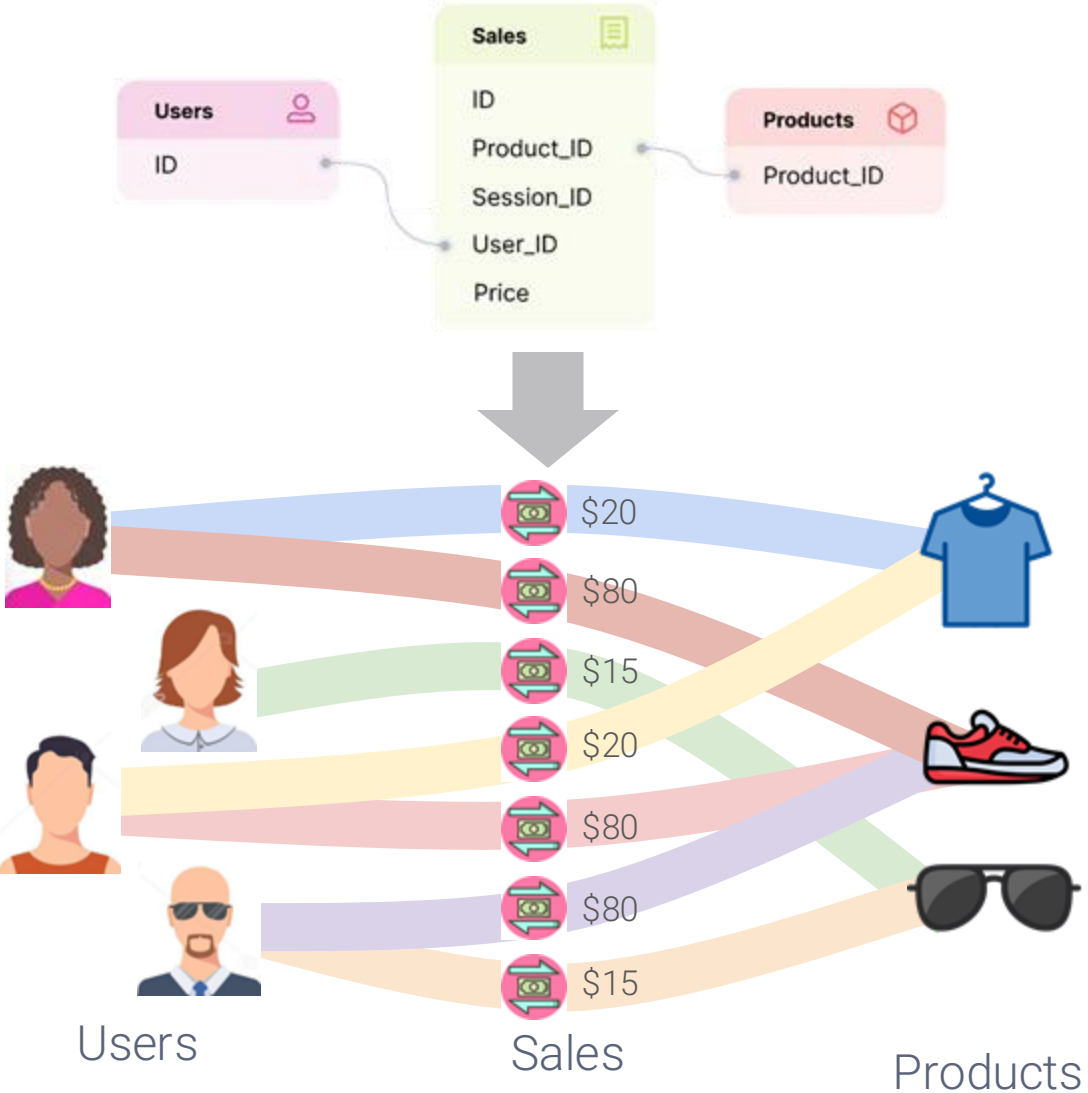


- The schema graphs can be more complex

Relational Entity Graph

Relational Entity Graph:

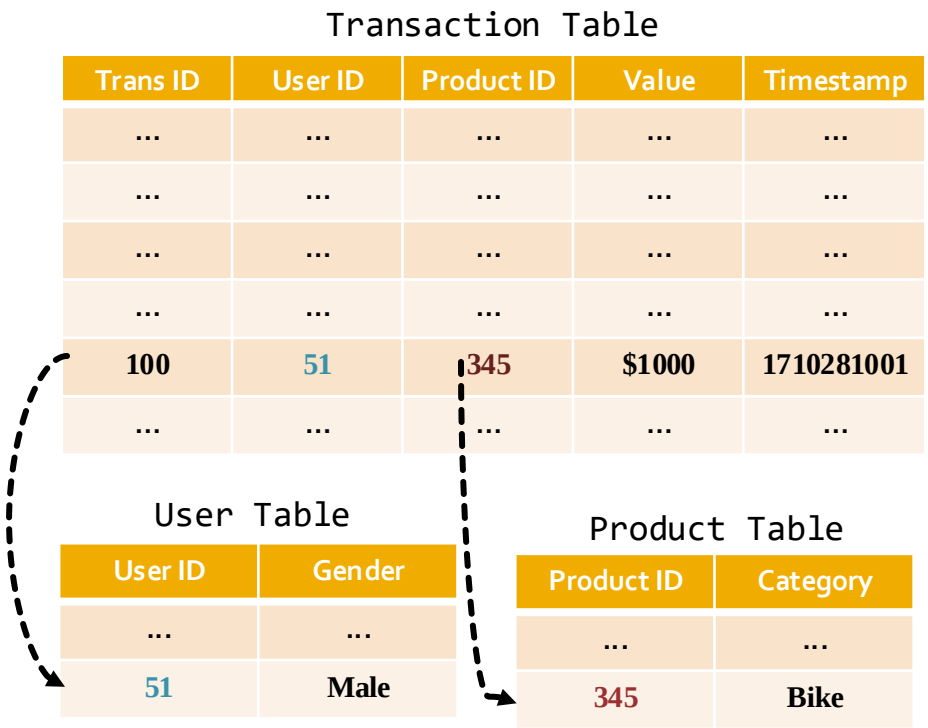
Create connections via primary-foreign keys



Mathematically…

- Given database of tables $\mathcal{T} = \{T_1, \ldots, T_n\}$ and relations $\mathcal{L} \subseteq \mathcal{T} \times \mathcal{T}$
- Each table is a set of entities $v \in T_i$ possessing a primary key and optional foreign keys
- The **relational entity graph** is such that
  - The set of nodes is defined by all rows in all tables $\mathcal{V} = \bigcup_{T \in \mathcal{T}} T$
  - The set of edges is defined by connecting two entities v_1, v_2 whose primary and foreign keys match

Entities in Relational Entity Graph



- Entities also have features (different than KGs)

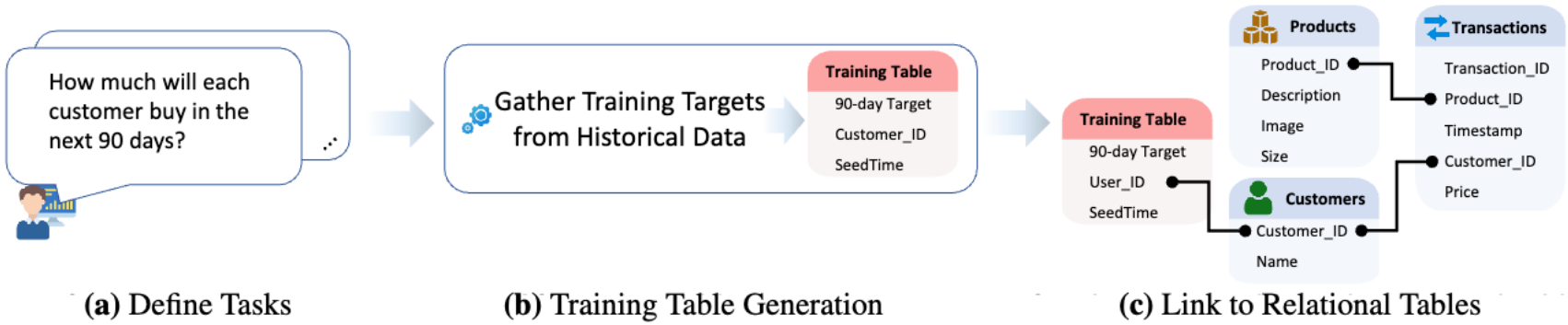
Stanford CS246: Predictive Tasks in Relational Databases

CS246: Mining Massive Datasets

Jure Leskovec, Stanford University

http://cs246.stanford.edu

Overview of the Approach



(a) Define Tasks **(b)** Training Table Generation **(c)** Link to Relational Tables

- Next:
  - Define task(s)
  - Relational Deep Learning

Defining a Task

- **Example:** Predict whether a user is going to churn in the next 30 days?
- Most tasks are **temporal**:
  - Customer's label changes over time
  - Database changes over time
    - For every time, a feature needs to be recomputed
- To define a task, we need:
  - Entity
  - Label
  - Time

Temporal tasks as especially challenging because features are time dependent:

- For every time, a feature needs to be recomputed
- Entity's label can change between time steps

Defining a Task: Training Table

- Training Table: A special table containing training labels
  - `(Entity ID, Time, Labels)`
    - Classification, Regression, Multi-class
  - Time column is essential for temporal prediction tasks
    - An entity may have different labels at different times
    - Only use information up to the time of label

Defining a Task: Training Table

- Training Table: A special table containing training labels
  - `(Entity ID, Time, Labels)`
    - Classification, Regression, Multi-class

`Training Table`

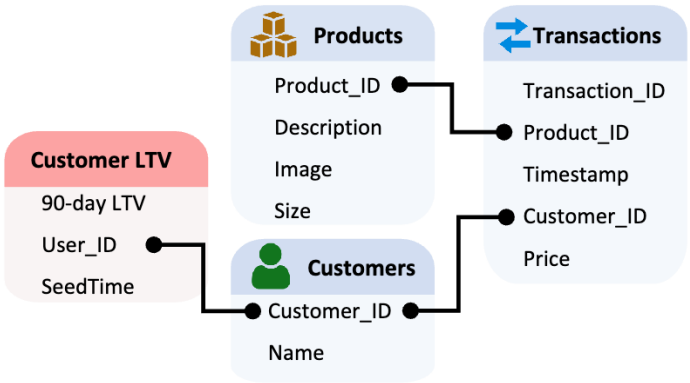
Entity ID	Timestamp	Label
99	10172024	1
99	10182024	1
…	…	…
100	10172024	1
100	10182024	0
…	…	…

Example: Churn

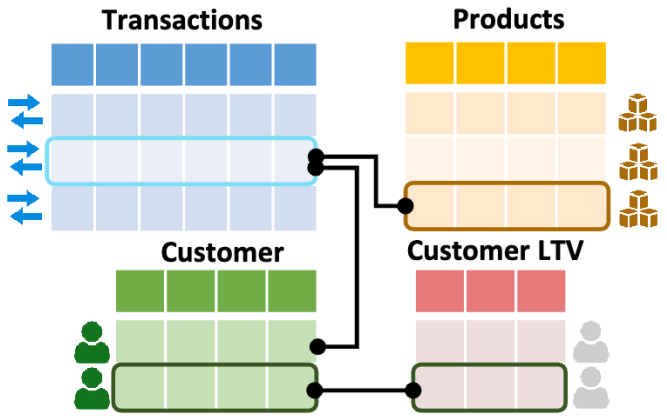
- **Schema:**

- Example prediction task:
  - Predict whether a user is going to churn
    - Zero sales in the next 30 days.
- Training table:
  - `(User, Time, Churn_label)`

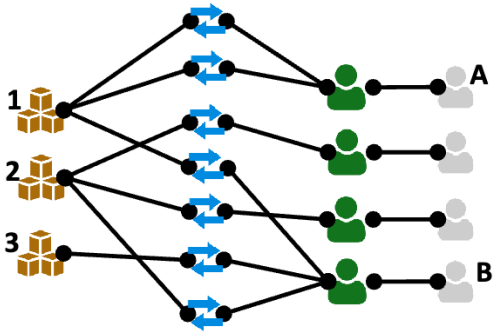
Relational Deep Learning



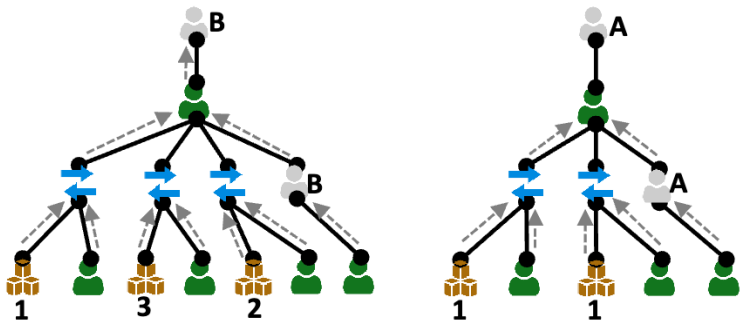
(a) Rel. Tables with Training Table



(b) Entities Linked by Foreign Keys



(c) Relational Entity Graph

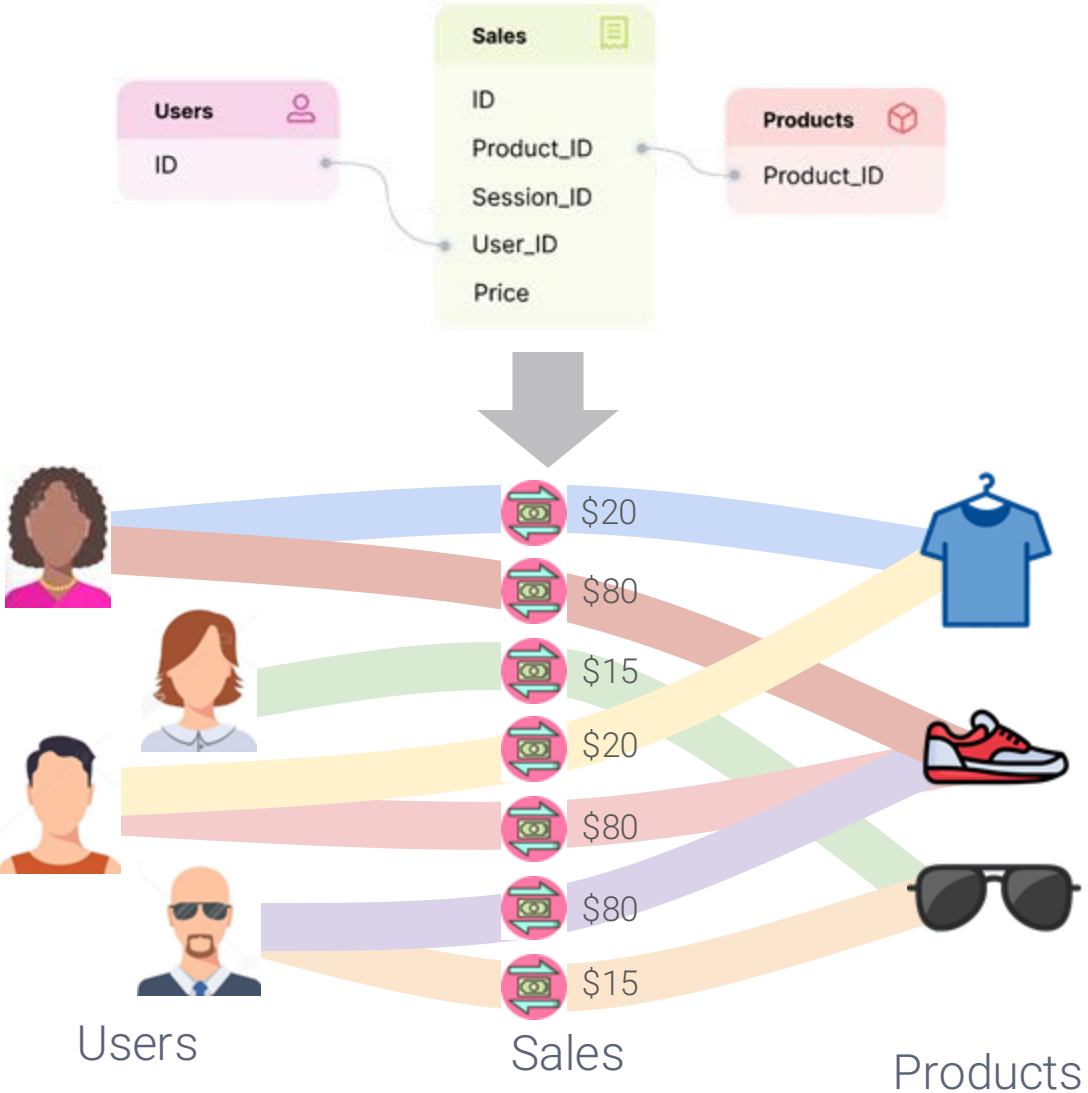


(d) Graph Neural Network

Relational Entity Graph

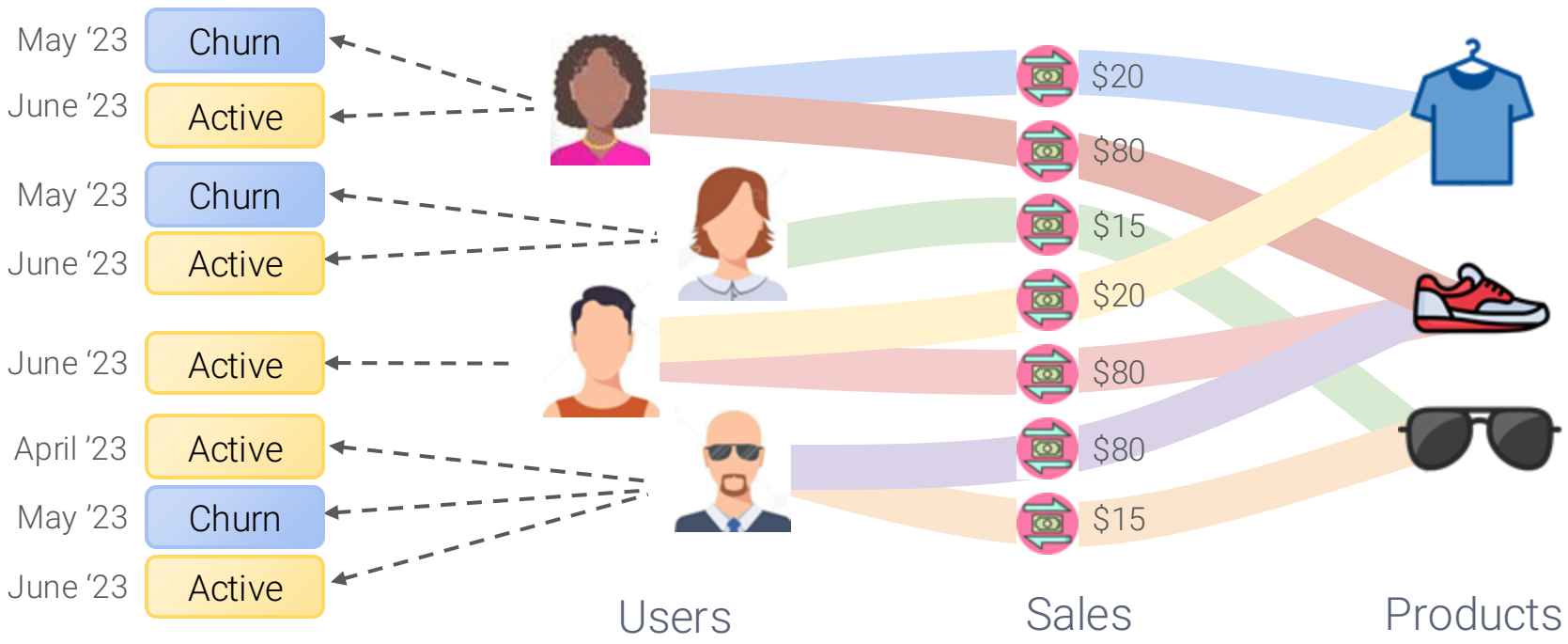
Relational Entity Graph:

Create connections via primary-foreign keys



Connect the Training Table

Training labels together with timestamps are attached to the graph



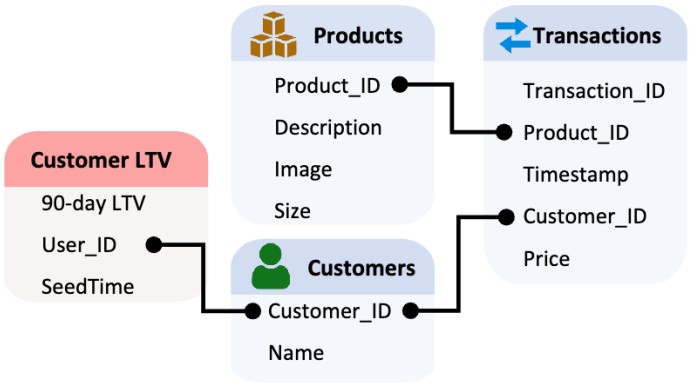
Stanford CS246: Relational GNN

CS246: Mining Massive Datasets

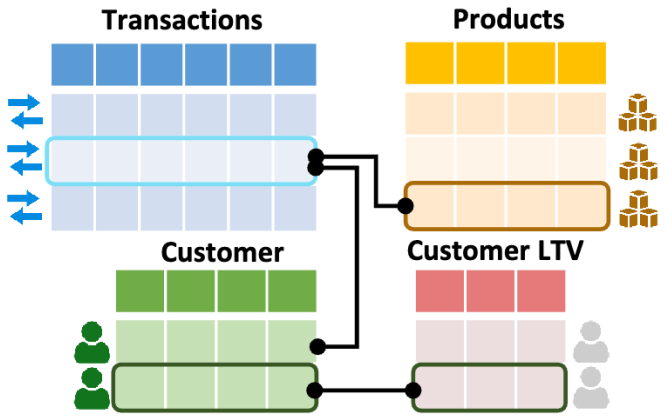
Jure Leskovec, Stanford University

http://cs246.stanford.edu

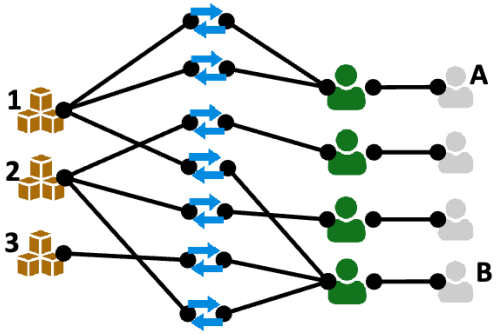
Relational Deep Learning



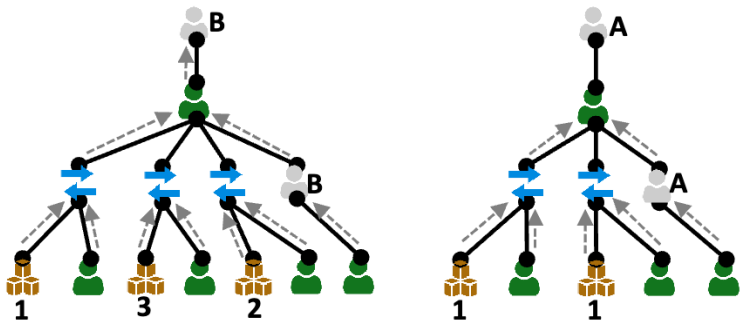
(a) Rel. Tables with Training Table



(b) Entities Linked by Foreign Keys



(c) Relational Entity Graph

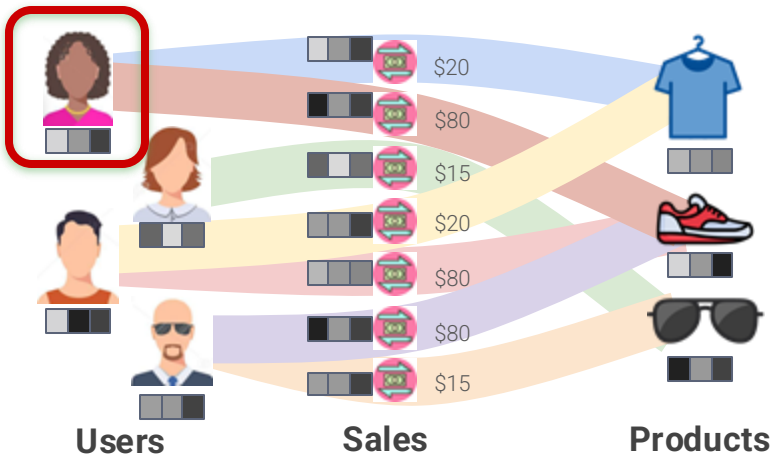


(d) Graph Neural Network

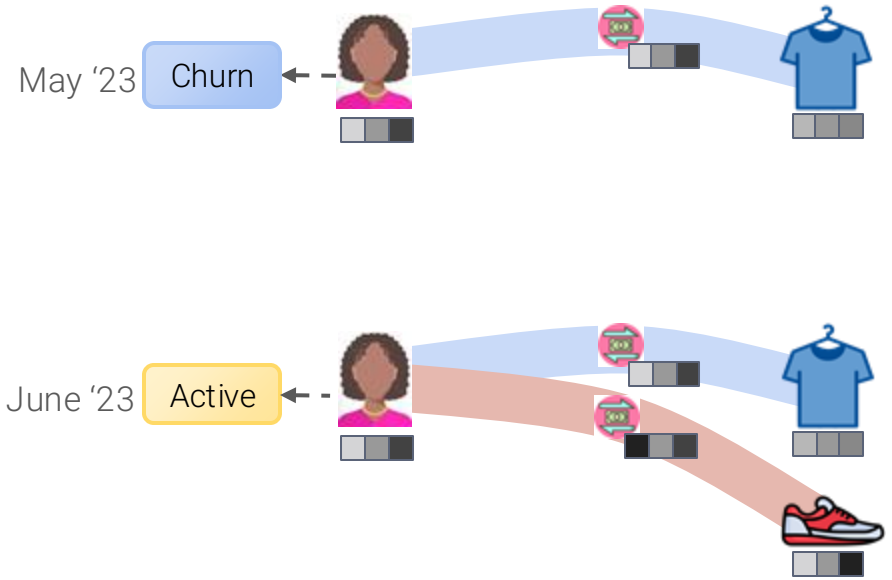
GNN on the Entity Graph

Node's neighborhood defines a computation graph

Nodes learn how to *optimally* use information from neighbors to obtain enhanced node representations

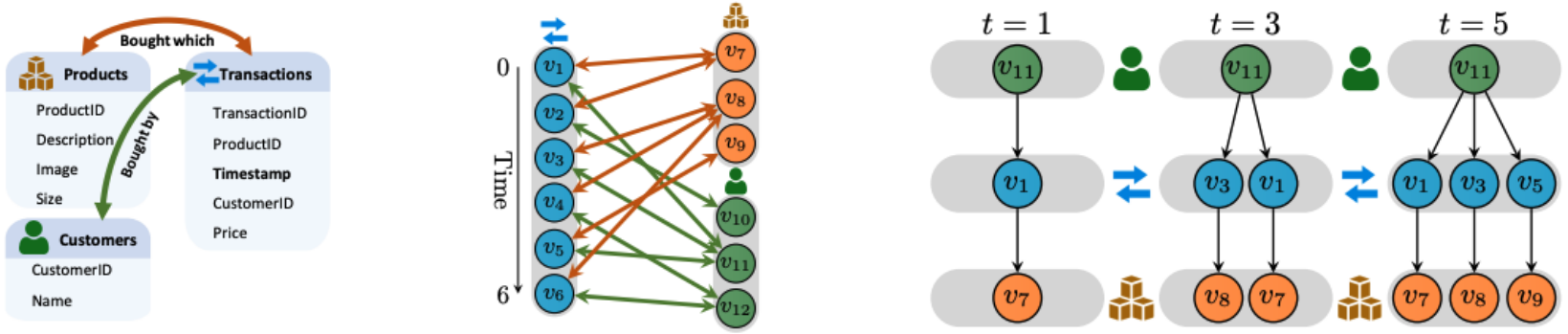


Entity Graph



GNN computation graphs

GNNs on Temporal Graphs

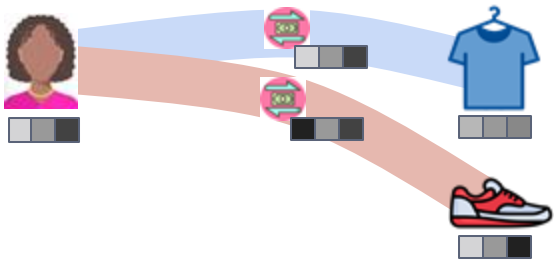


(a) Schema Graph (b) Relational Entity Graph (c) Computation Graphs for different time t

- The computation graph for each node is **time-dependent**
- Message+Aggregation becomes **time-dependent**
- Sampling over neighbors is **time-dependent**

GNN vs Feature Engineering

GNN-based features:



vs.

Hand-engineered features:

SUM(TRANSACTIONS.Price) over (-30, 0) days	AVG(TRANSACTIONS.Price) over (-30, 0) days	...

GNN aggregation is **learnable version of hand-crafted features**!

GNNs give better performance by learning optimal features.

SQL joins vs. Graph edges

Definitions:

- A table R is a set of entities $R = \{r_1, \ldots, r_n\}$
- Each entity is a tuple $r_i = (r_{i1}, r_{i2}, \ldots, r_{in})$
- Given two tables, R, S, a join operation is a subset of a Cartesian product:

- Aggregation will specify which rows are kept

Ex.1: Cross Join

Employee table

LastName	DepartmentID
Rafferty	31
Jones	33
Heisenberg	33
Robinson	34
Smith	34
Williams	NULL

Department table

DepartmentID	DepartmentName
31	Sales
33	Engineering
34	Clerical
35	Marketing

```
SELECT *
FROM employee INNER JOIN department ON 1=1;
```

Employee.LastName	Employee.DepartmentID	Department.DepartmentName	Department.DepartmentID
Rafferty	31	Sales	31
Jones	33	Sales	31
Heisenberg	33	Sales	31
Smith	34	Sales	31
Robinson	34	Sales	31
Williams	NULL	Sales	31
Rafferty	31	Engineering	33
Jones	33	Engineering	33
Heisenberg	33	Engineering	33
Smith	34	Engineering	33
Robinson	34	Engineering	33
Williams	NULL	Engineering	33
Rafferty	31	Clerical	34
Jones	33	Clerical	34
Heisenberg	33	Clerical	34
Smith	34	Clerical	34
Robinson	34	Clerical	34
Williams	NULL	Clerical	34
Rafferty	31	Marketing	35
Jones	33	Marketing	35
Heisenberg	33	Marketing	35
Smith	34	Marketing	35
Robinson	34	Marketing	35
Williams	NULL	Marketing	35

https://en.wikipedia.org/wiki/Relational_algebra

SQL joins vs. Graph edges

- Connection between SQL Joins and graph edges

Ex.2: Inner Join

$R \bowtie S := \{r \cup s | r \in R \wedge s \in S \wedge \mathrm{Fun}(r,s)\}$

$\mathrm{Fun}(r,s): r.pkey = s.fkey$

Employee table

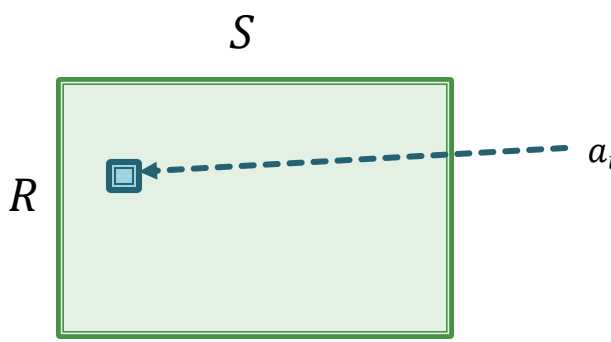
LastName	DepartmentID
Rafferty	31
Jones	33
Heisenberg	33
Robinson	34
Smith	34
Williams	NULL

Department table

DepartmentID	DepartmentName
31	Sales
33	Engineering
34	Clerical
35	Marketing

```
SELECT employee.LastName, employee.DepartmentID, department.DepartmentName
FROM employee
INNER JOIN department ON
employee.DepartmentID = department.DepartmentID;
```

Employee.LastName	Employee.DepartmentID	Department.DepartmentName
Robinson	34	Clerical
Jones	33	Engineering
Smith	34	Clerical
Heisenberg	33	Engineering
Rafferty	31	Sales



$$a_{ij} = \begin{cases} 1, \text{if } r_i.pkey = s_j.fkey \\ 0, \text{otherwise} \end{cases}$$

These are exactly the pkey-fkey edges!

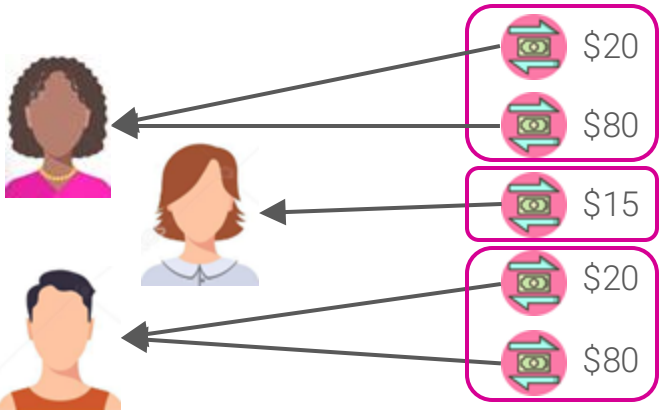
https://en.wikipedia.org/wiki/Relational_algebra

GNNs perform a JOIN+AGG

Input:

USER_ID
1
2
3

USER_ID	VALUE	DATE
1	20	01-01
1	80	01-02
2	15	01-01
3	20	01-02
3	80	01-03



$$\bigoplus_{w \in \mathcal{N}(v)} \mathbf{t}(t_w) \cdot \mathbf{x}_w$$

GNN can learn:

```
SELECT SUM(VALUE)
FROM SALES
WHERE DATE > 01-01
GROUP BY USER_ID
```

USER_ID	SUM(VALUE)
1	80
2	0
3	100

Learnable Aggregation $\bigoplus$

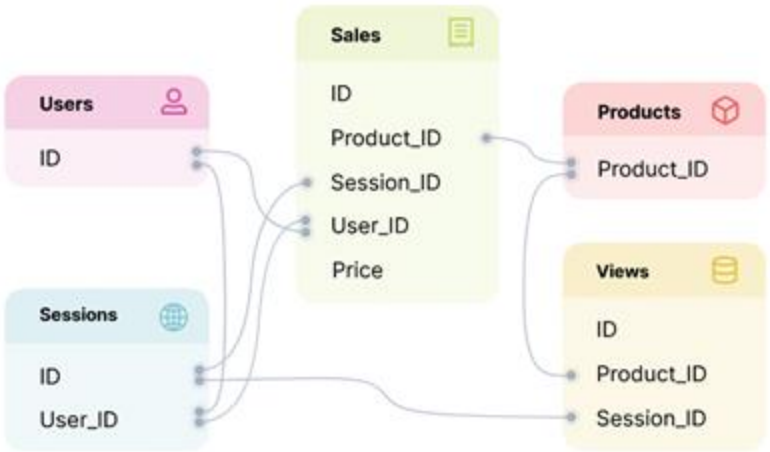
Temporal embedding $\mathbf{t}(t_w)$

Fact representation $\mathbf{x}_w$

Benefits of GNNs

GNNs learn how to aggregate information:

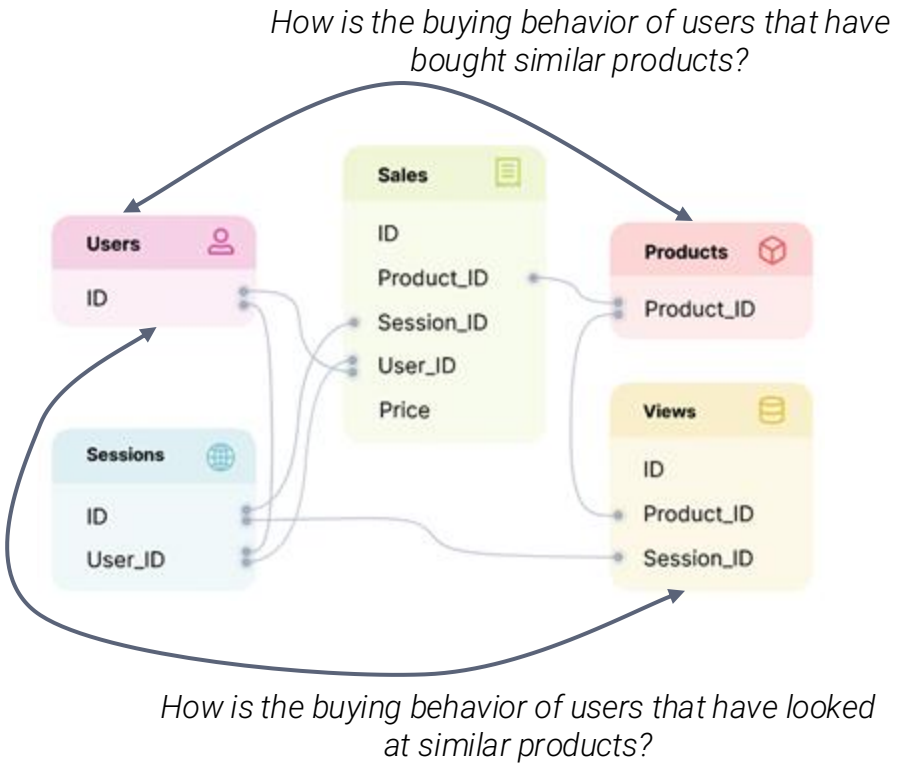
- They can discard neighboring node information that is irrelevant for the given downstream task
- They can detect fine-grained patterns within local neighborhoods (e.g., buying behavior over the last year)



Benefits of GNNs

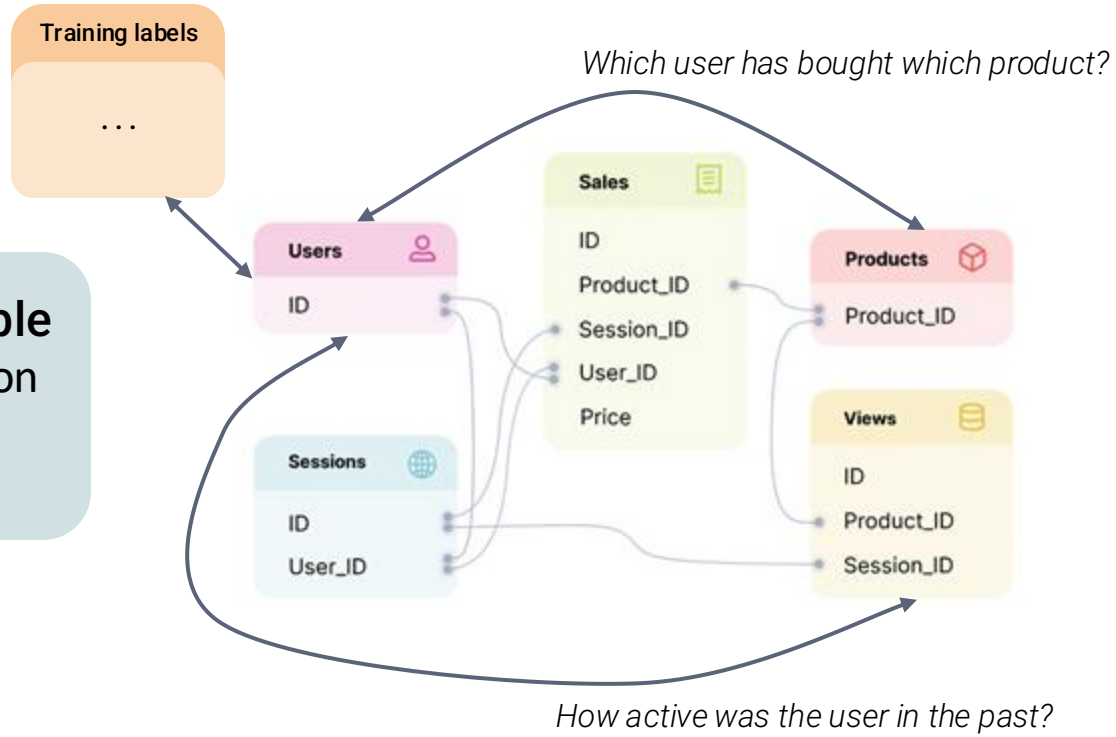
GNNs exchange information *across* training examples:

- Instead of treating examples as isolated, there now exists an inter-dependency *between* entities (*e.g.*, users with similar features, users with similar behavior)
- GNN can *use* these features to enrich an entity's representation



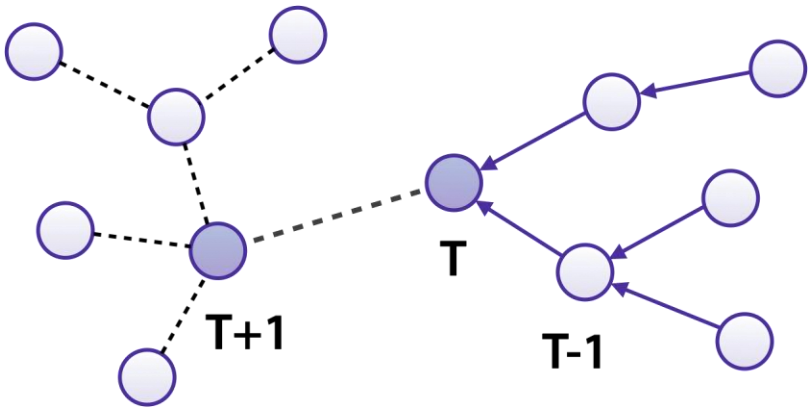
Benefits of GNNs

Multi-hop reasoning *across* **table boundaries** can catch information which is **hard** to pre-compute *beforehand*



Benefits of GNNs

- **First-class temporal support:**
  - Capture fine-grained relative and seasonal features via temporal embeddings
  - Avoid data leakage via temporal sampling directly during data loading



Full Vision Described in Paper

Relational Deep Learning: Graph Representation Learning on Relational Tables

Matthias Fey[2,*], **Weihua Hu**[2,*], **Kexin Huang**[1,*], **Jan Eric Lenssen**[2,3,*], **Rishabh Ranjan**[1,*], **Joshua Robinson**[1,*], **Rex Ying**[4], **Jiaxuan You**[5], **Jure Leskovec**[1,2]

[*]Equal contribution. Listed in alphabetic order.

[1]Stanford University
[2]Kumo.AI
[3]Max Planck Institute for Informatics
[4]Yale University
[5]University of Illinois at Urbana-Champaign

Available at: https://relbench.stanford.edu

Stanford CS246: RELBENCH

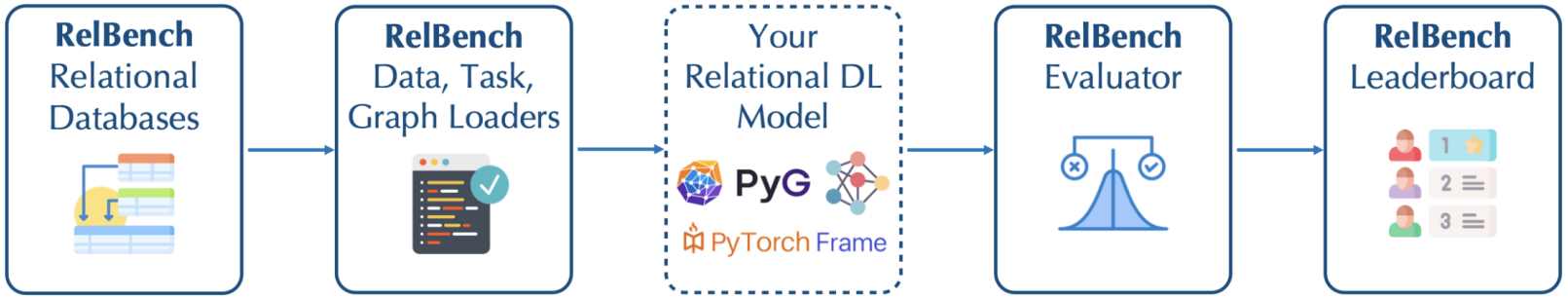
CS246: Mining Massive Datasets

Jure Leskovec, Stanford University

http://cs246.stanford.edu

Enabling Research on RDL

- Relbench is more than just a collection of Databases



- Automatically download datasets
- Load database and task tables
- Standardized evaluation protocol:
  - Prevents temporal leakage from test set
- Framework-agnostic data structures: use your favorite ML stack!

PyF and PyG Integration

- Load as a PyG graph
- Train GNN end-to-end
- Temporal neighbor sampling
- Use PyTorch Frame to encode tables

RelBench Datasets

7 Diverse Datasets

E-Commerce

- rel-amazon
- rel-avito
- rel-hm

Social

- rel-event
- rel-stack

Sports

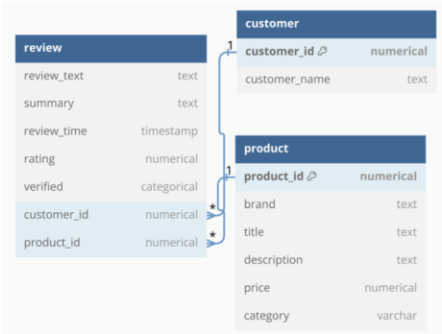
- rel-f1

Medical

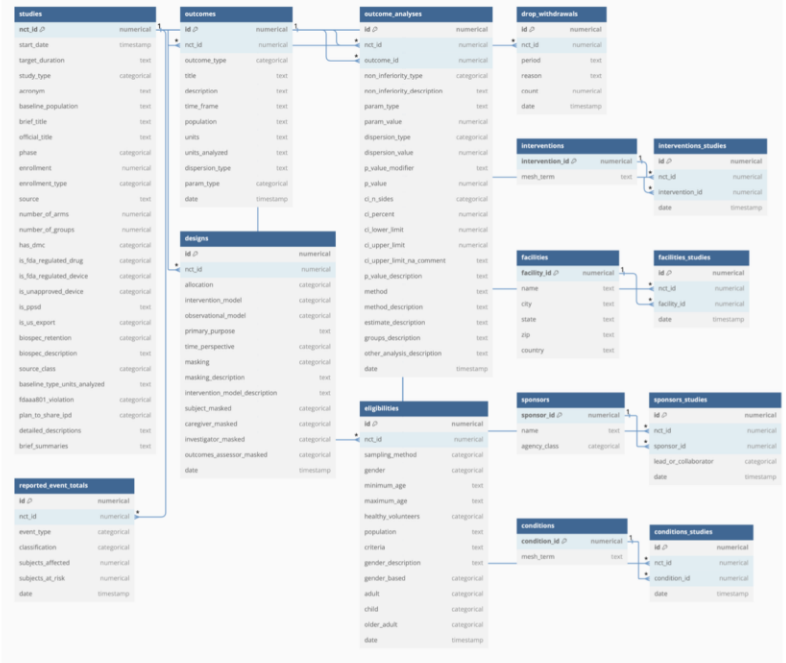
- rel-trial

RelBench Datasets

Rich Schemas



rel-amazon



rel-trial

3 to **15** tables

74k to **41M** rows in a DB

15 to **140** columns in a DB

Time span from **2 weeks** to **55 years**

RelBench Tasks

30 Real-World Predictive Tasks

Entity Classification

- rel-amazon
  - user-churn
  - item-churn
- rel-stack
  - user-badge
- rel-trial
  - study-outcome

…

Entity Regression

- rel-amazon
  - user-ltv
  - item-ltv
- rel-avito
  - ad-ctr
- rel-f1
  - driver-position

…

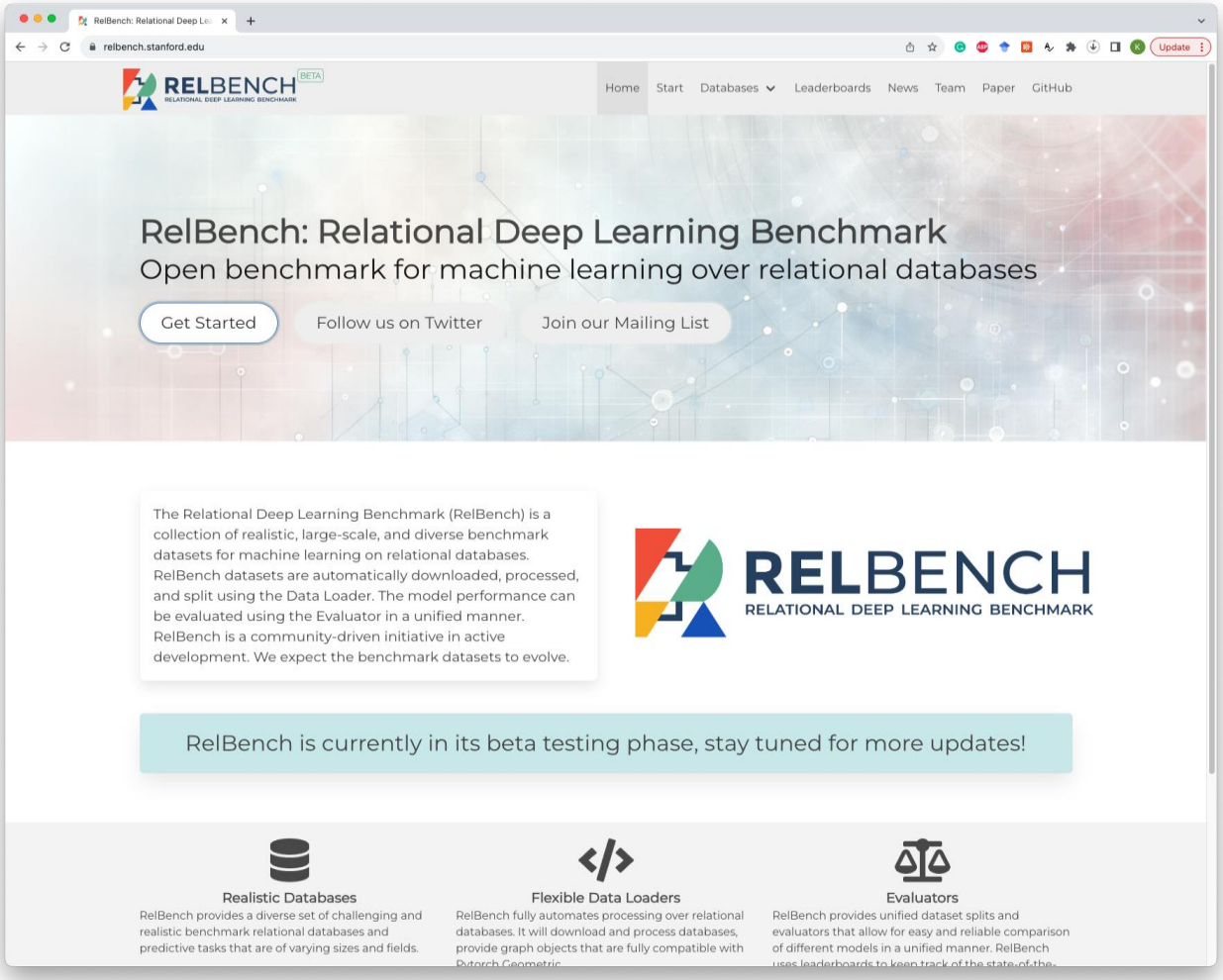
Recommendation

- rel-amazon
  - user-item-purchase
- rel-avito
  - user-ad-visit
- rel-stack
  - user-post-comment
  - post-post-related

…

See website to get started

https://relbench.stanford.edu



Stanford CS246: GNN vs expert Data Scientist

CS246: Mining Massive Datasets

Jure Leskovec, Stanford University

http://cs246.stanford.edu

Expert Data Scientist

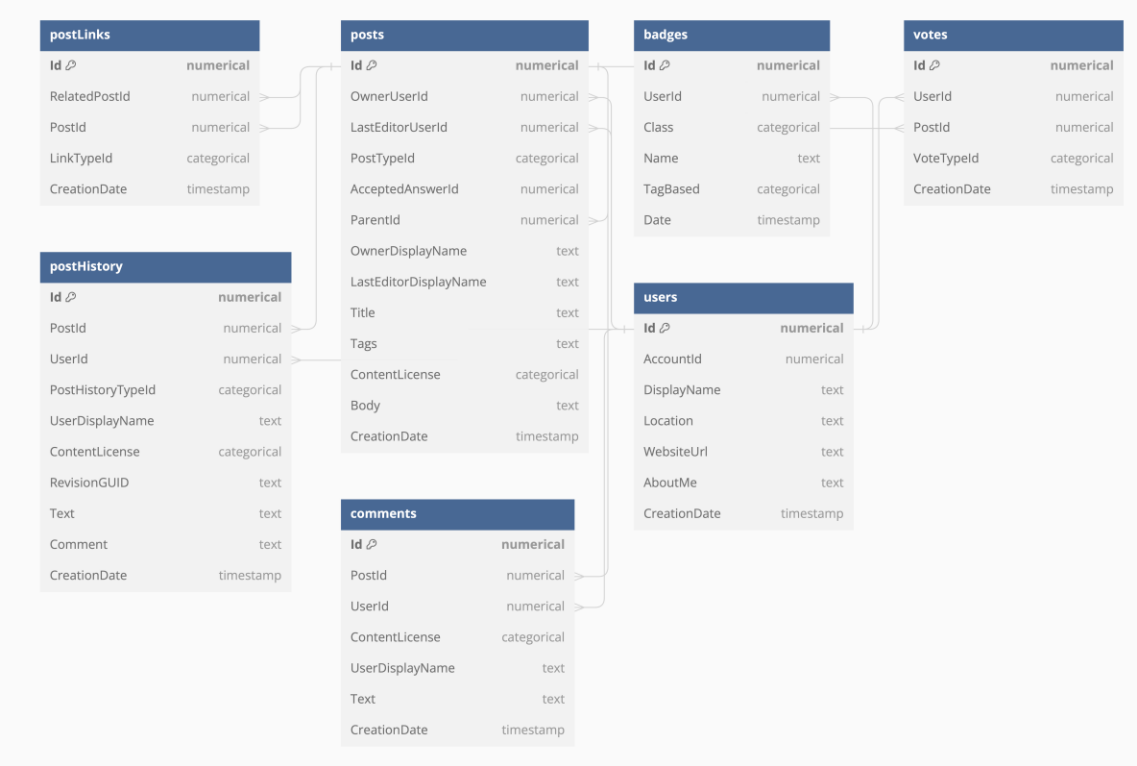
(Alejandro)

- Recruited Experienced Data Scientist
- 5 years in industry *(specializing in financial databases)*
- Responsible for full model building lifecycle *(more detail next)*

Representative Relbench task

Q: Will a user be active in the next 6 months?

Stack Exchange Database



Expert Data Scientist Workflow

Task: Will a user be active in the next 6 months?

Expert Data Scientist Workflow

Manual work

Exploratory Data Analysis (EDA)

4hrs

Task: Will a user be active in the next 6 months?

Expert Data Scientist Workflow

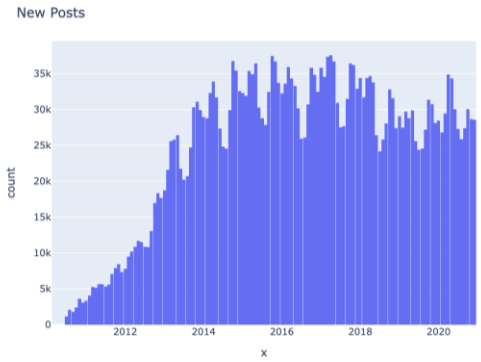
Manual work

Task: Will a user be active in the next 6 months?

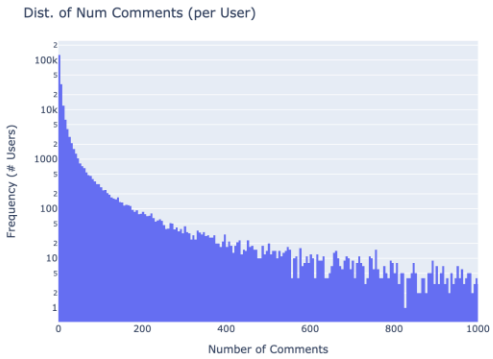
Exploratory Data Analysis (EDA)

4hrs

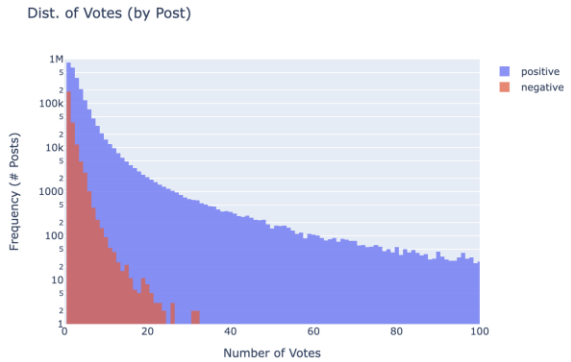
Example observations



Activity is seasonal



Comments follow power law



Negative votes are infrequent

Expert Data Scientist Workflow

Manual work

Exploratory Data Analysis (EDA) — 4hrs

↓

Feature Ideation — 0.5hr

Example features

- ✓ weeks_since_last_comment
- ✓ badge_score
- ✓ location_is_null

Task: Will a user be active in the next 6 months?

Expert Data Scientist Workflow

Manual work

Exploratory Data Analysis (EDA) — 4hrs

↓

Feature Ideation — 0.5hr

↓

SQL query writing — 5hr

Example features

- ✓ weeks_since_last_comment
- ✓ badge_score
- ✓ location_is_null

Task: Will a user be active in the next 6 months?

```
create table churn_feats_train as

with labels as (
    select * from train_labels
),

badge_freqs as (
    select
        Name,
        count(*) / (sum(count(*)) over ()) as badge_incidence
    from badges
    group by Name
),

badge_feats as (
    select
        labels.OwnerUserId as user_id,
        labels.timestamp,
        coalesce(count(distinct badges.Id), 0) as num_badges,
        coalesce(sum(log(1 / badge_freqs.badge_incidence)), 0) as badge_score
    from labels
    left join badges
        on
            labels.OwnerUserId = badges.UserId
            and labels.timestamp > badges.Date
    left join badge_freqs
        on badges.Name = badge_freqs.Name
    group by all
),

user_feats as (
```

100s of lines of code

Expert Data Scientist Workflow

Manual work

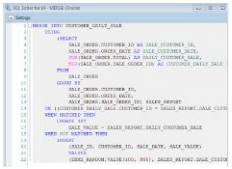
Task: Will a user be active in the next 6 months?

Exploratory Data Analysis (EDA) — 4hrs

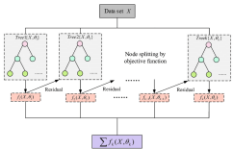
Example features

- ✓ weeks_since_last_comment
- ✓ badge_score
- ✓ location_is_null

Feature Ideation — 0.5hr



SQL query writing — 5hr



XGBoost hparam sweep — 2hr

Expert Data Scientist Workflow

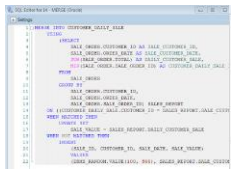
Manual work

Task: Will a user be active in the next 6 months?

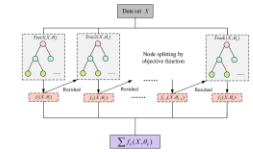
Exploratory Data Analysis (EDA) — 4hrs

Example features

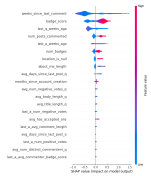
- ✓ weeks_since_last_comment
- ✓ badge_score
- ✓ location_is_null



Feature Ideation — 0.5hr



SQL query writing — 5hr



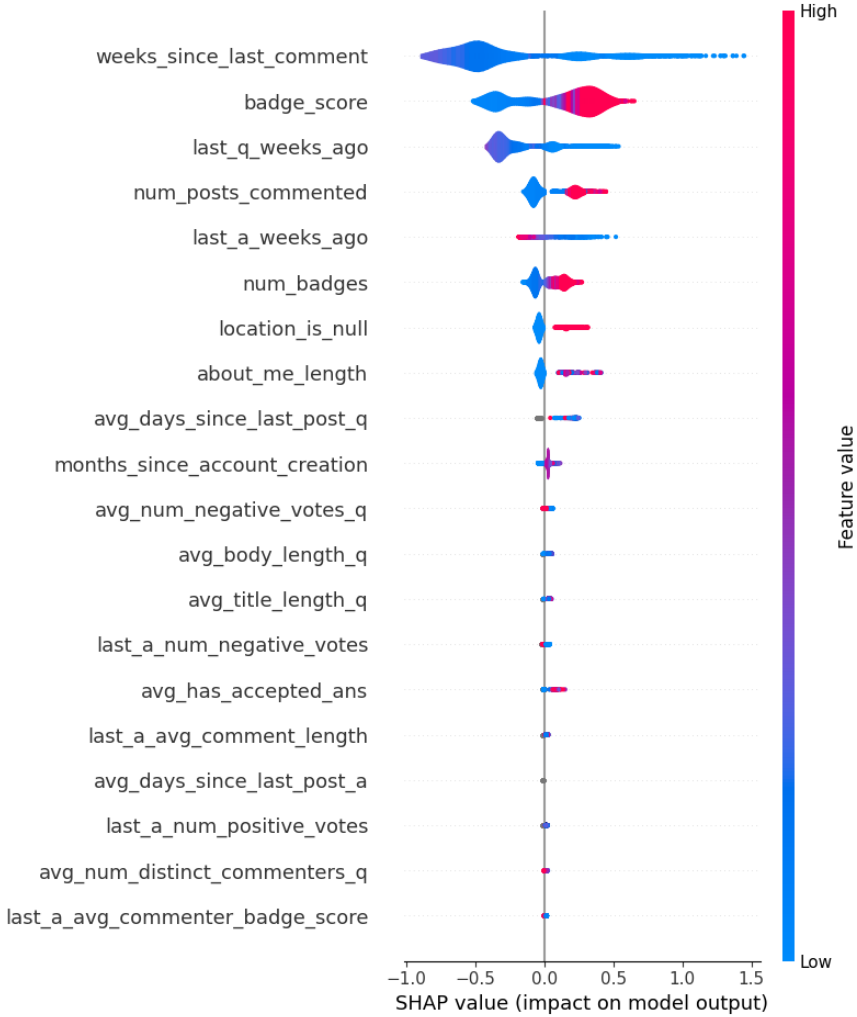
XGBoost hparam sweep — 2hr

SHAP (feature importance analysis) — 1hr

SHAP feature importance analysis

Selected Observations

- *Website "seniority" **predictive** (total number of comments, badge score etc.)*
- *Time since last active / commented **is predictive***
- *Completed bio (about me, location etc.) **is predictive***
- *Number of positive/negative votes **not predictive***
- *Interacting with "senior" community members **not predictive***



Final list of features

~12 hours of high-quality expert work

[22]:

	Label Corr.	Label MI	NaN %
num_badges	0.229	0.057	0.0%
num_questions_last_6mo	0.209	0.067	89.7%
badge_score	0.206	0.069	0.0%
ans_acceptance_rate	0.195	0.094	96.4%
avg_comment_length	0.135	0.045	0.0%
num_posts_commented	0.131	0.061	0.0%
avg_num_tags	0.131	0.077	89.7%
avg_has_accepted_ans	0.130	0.106	89.7%
num_comments	0.130	0.057	0.0%
num_answers_last_6mo	0.125	0.037	96.4%
last_q_num_tags	0.111	0.014	13.6%
avg_num_positive_votes_a	0.108	0.070	96.4%
last_a_is_accepted_ans	0.106	0.049	73.3%
last_q_body_length	0.096	0.007	13.6%
about_me_length	0.096	0.010	0.0%
avg_num_comments_a	0.084	0.088	96.4%
last_q_has_accepted_ans	0.076	0.009	13.6%
avg_num_distinct_commenters_a	0.073	0.073	96.4%
avg_body_length_q	0.067	0.051	89.7%
avg_num_positive_votes_q	0.060	0.078	89.7%
last_a_body_length	0.056	0.023	73.3%
avg_body_length_a	0.034	0.033	96.4%
last_q_title_length	0.023	0.004	13.6%
last_a_num_comments	0.023	0.040	73.3%
last_q_num_comments	0.015	0.003	13.6%
avg_avg_comment_length_a	0.012	0.047	96.4%
last_q_avg_comment_length	0.009	0.001	13.6%
display_name_is_null	0.004	0.001	0.0%
last_a_num_distinct_commenters	0.002	0.057	73.3%
last_q_num_distinct_commenters	-0.002	0.006	13.6%
last_q_num_positive_votes	-0.006	0.002	13.6%
last_a_num_positive_votes	-0.009	0.030	73.3%
last_a_avg_comment_length	-0.010	0.022	73.3%
avg_num_comments_q	-0.011	0.078	89.7%
avg_commenter_badge_score_q	-0.015	0.048	89.7%
avg_title_length_q	-0.022	0.064	89.7%
last_q_num_negative_votes	-0.026	0.002	13.6%
avg_num_distinct_commenters_q	-0.037	0.096	89.7%
avg_avg_comment_length_q	-0.043	0.057	89.7%
last_q_avg_commenter_badge_score	-0.046	0.006	13.6%
last_a_avg_commenter_badge_score	-0.048	0.023	73.3%
avg_commenter_badge_score_a	-0.060	0.032	96.4%
website_url_is_null	-0.061	0.032	0.0%
last_a_num_negative_votes	-0.085	0.051	73.3%
months_since_account_creation	-0.113	0.011	0.0%
avg_num_negative_votes_q	-0.114	0.100	89.7%
location_is_null	-0.136	0.051	0.0%
avg_days_since_last_post_a	-0.168	0.035	97.1%
avg_days_since_last_post_q	-0.169	0.055	93.5%
avg_num_negative_votes_a	-0.170	0.088	96.4%
last_q_weeks_ago	-0.346	0.073	13.6%
weeks_since_last_comment	-0.392	0.091	34.3%
last_a_weeks_ago	-0.394	0.049	73.3%
last_a_num_tags	nan	0.065	73.3%

Stanford CS46: Results

CS246: Mining Massive Datasets

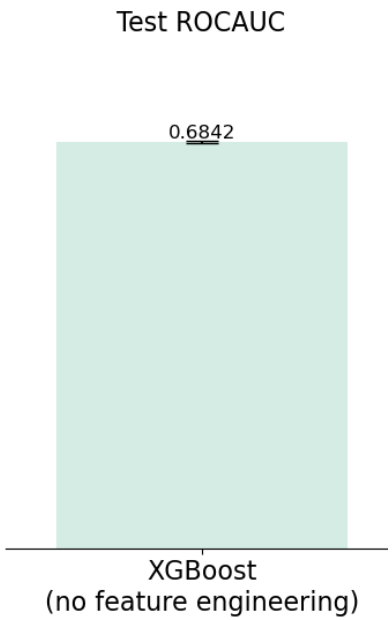
Jure Leskovec, Stanford University

http://cs246.stanford.edu

Results

Task: Will a user be active in the next 6 months?

Naive baseline

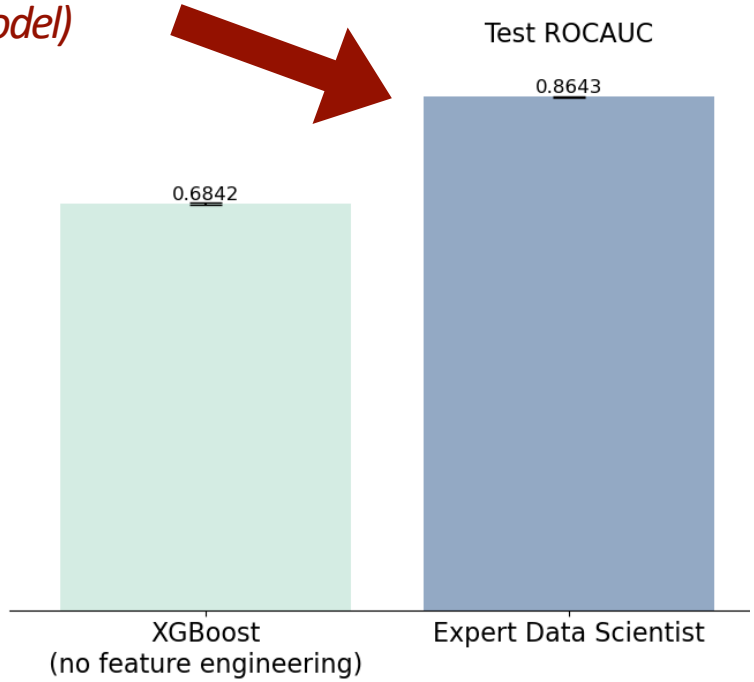


users	
Id	numerical
AccountId	numerical
DisplayName	text
Location	text
WebsiteUrl	text
AboutMe	text
CreationDate	timestamp

Results

12 hours manual work
~682 lines of code
1hr model training

(2 days to model)



**Work measured as the marginal effort to solve a new task*

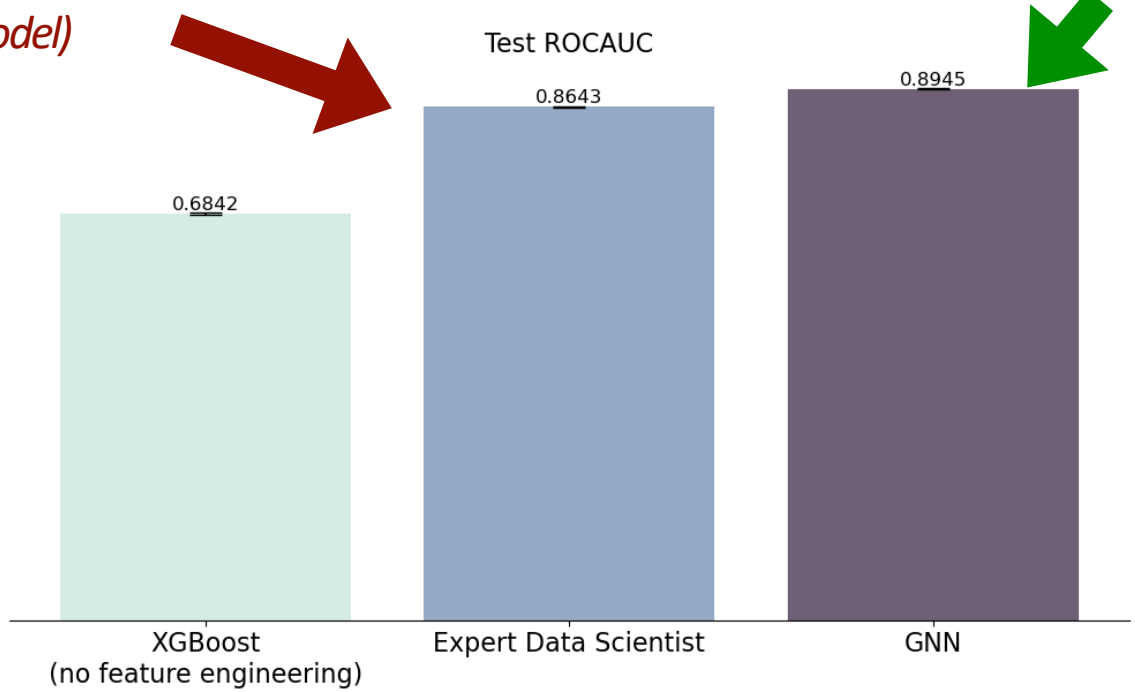
Results

Task: Will a user be active in the next 6 months?

12 hours manual work
~682 lines of code
1hr model training

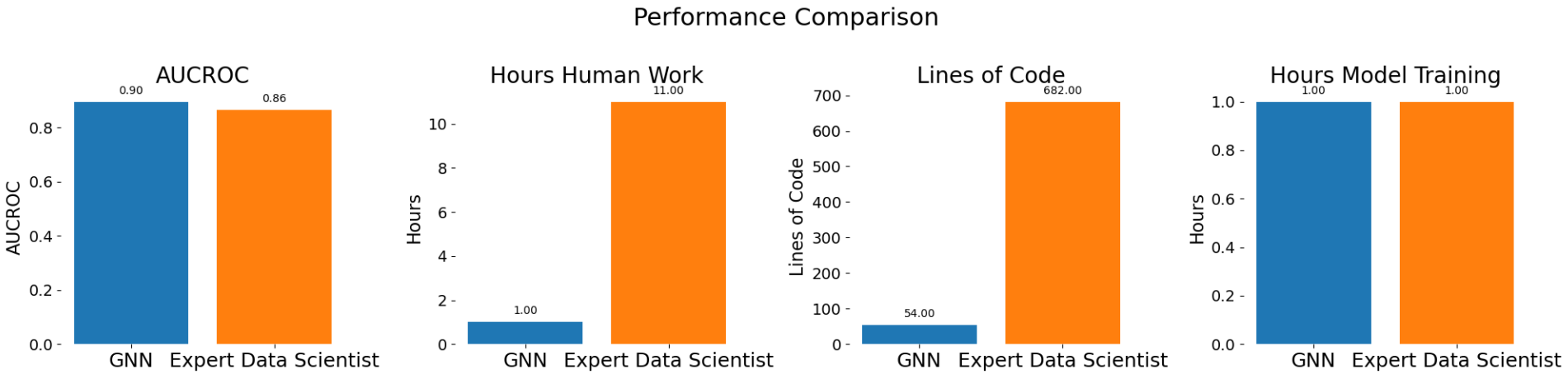
(2 days to model)

<1 hour manual work
54 lines of code
1hr model training



Results

Task: Will a user be active in the next 6 months?



More Relbench tasks