ACM RecSys Challenge 2016

Job Recommendation System

Sonu Mishra, Manoj Reddy

Outline

1. **Introduction**
2. Data sets
3. Analysis and Preprocessing
4. Methodologies
5. Evaluation
6. Conclusion and Future work

Premier conference in the field of Recommender Systems

To be held during 15-19th September 2016, Boston @ MIT & IBM

RecSys Challenge:

- Build a job recommendation system for XING
- Given a XING user, the goal is to predict those job postings that a user will positively interact with (e.g. click, bookmark)
- Submission deadline: June 26, 2016

XING

Find jobs, contacts, events...

Advanced Search

XING

Fabian Abel

PREMIUM

- My start page
- My contacts 25
- My messages 51
- My Premium
  NEW: XING Arbeitsrechtsschutz
- Jobs
- Events

Network news | Comments and likes

Jobs we think you'll like

DevOps Engineer (m/f) for Data...
XING AG

Software Architekt (m/w) mit d...
adesso AG

Projektleiter (m/w) im Bereich...
adesso AG

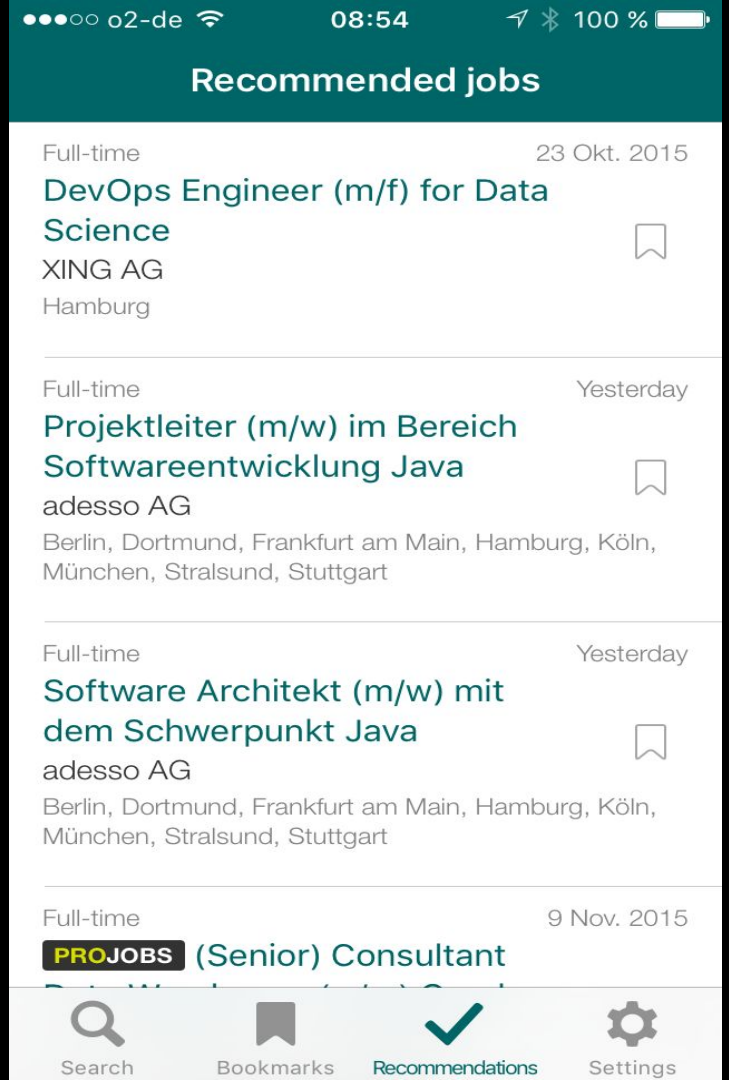
(Senior) Consultant Data Wareh...
empiricus GmbH - Agentur für I...

> 16 more job recommendations

Share something with your contacts

What's new?

XING


o2-de
08:54
100 %
Recommended jobs
Full-time
23 Okt. 2015
DevOps Engineer (m/f) for Data Science
XING AG
Hamburg
Full-time
Yesterday
Projektleiter (m/w) im Bereich Softwareentwicklung Java
adesso AG
Berlin, Dortmund, Frankfurt am Main, Hamburg, Köln, München, Stralsund, Stuttgart
Full-time
Yesterday
Software Architekt (m/w) mit dem Schwerpunkt Java
adesso AG
Berlin, Dortmund, Frankfurt am Main, Hamburg, Köln, München, Stralsund, Stuttgart
Full-time
9 Nov. 2015
PROJOBS (Senior) Consultant
Search
Bookmarks
Recommendations
Settings

1. Introduction
2. **Data sets**
3. Analysis and Preprocessing
4. Methodologies
5. Evaluation
6. Conclusion and Future work

Datasets

Users

- Job roles
- Career level
- Discipline
- Industry
- Country
- Region
- Work experience
- Education
- 1.5M records

Items

- Title
- Discipline
- Industry
- Country
- Region
- Type of employment
- Tags
- Creation time
- 1.3M records

Impressions

- User_ID
- Year
- Week
- Items
- 10M records

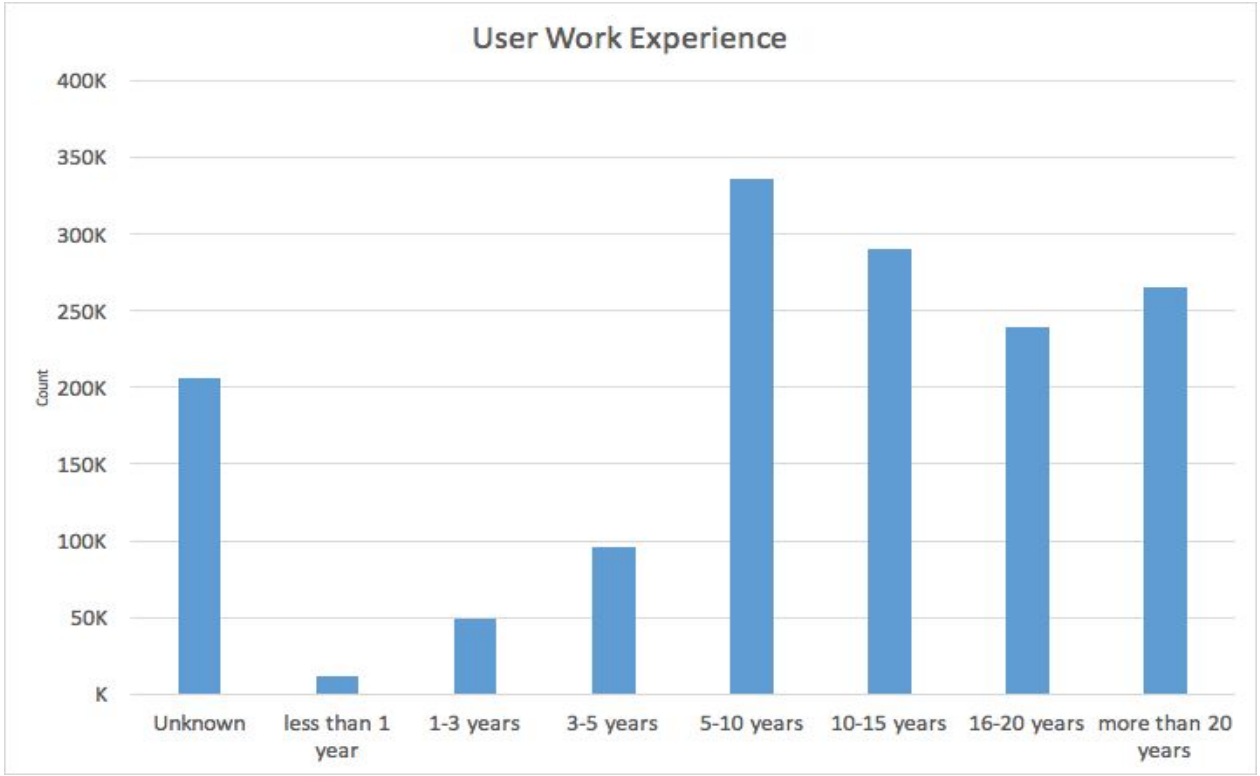
Interactions

- User_ID
- Item_ID
- Time
- Interaction_type:
  - 1 = clicked
  - 2= bookmarked
  - 3= reply/apply
  - 4=deleted
- 8M records

1. Introduction
2. Data sets
3. **Analysis and Preprocessing**
4. Methodologies
5. Evaluation
6. Conclusion and Future work

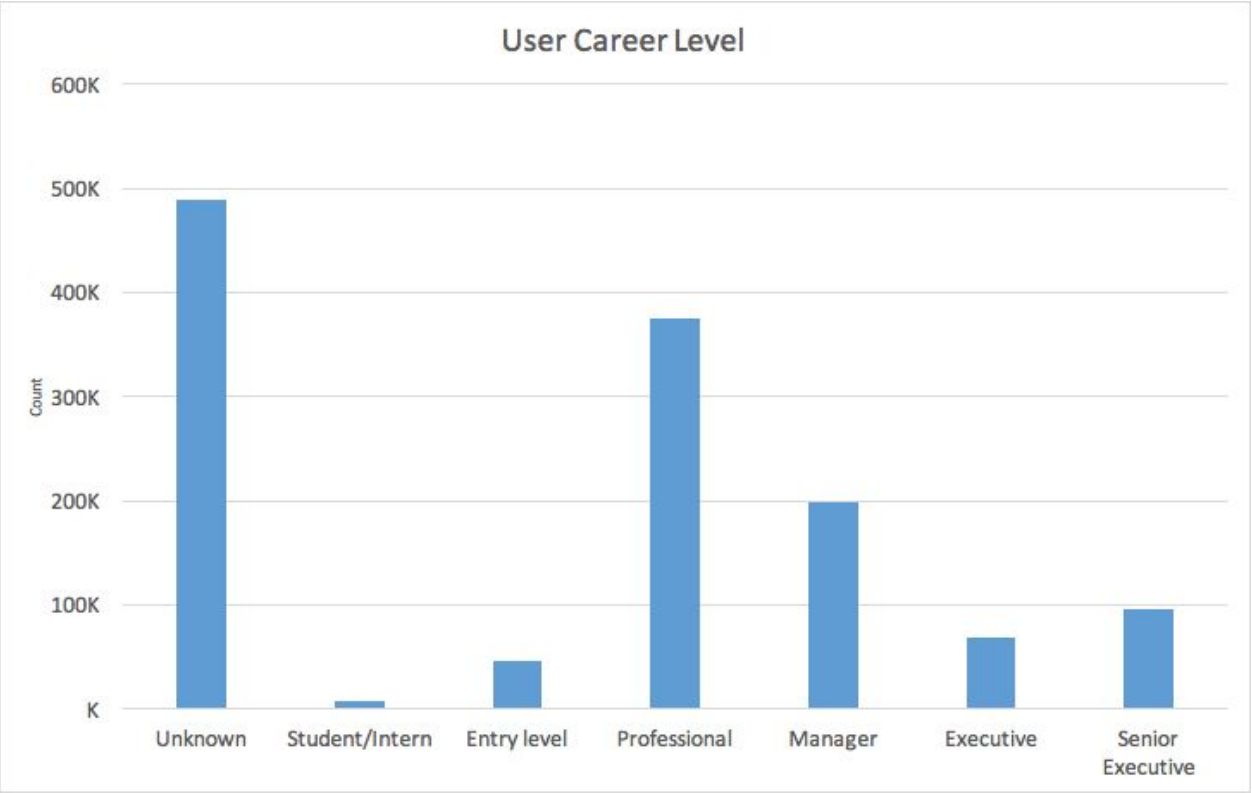
Dataset Analysis

Users



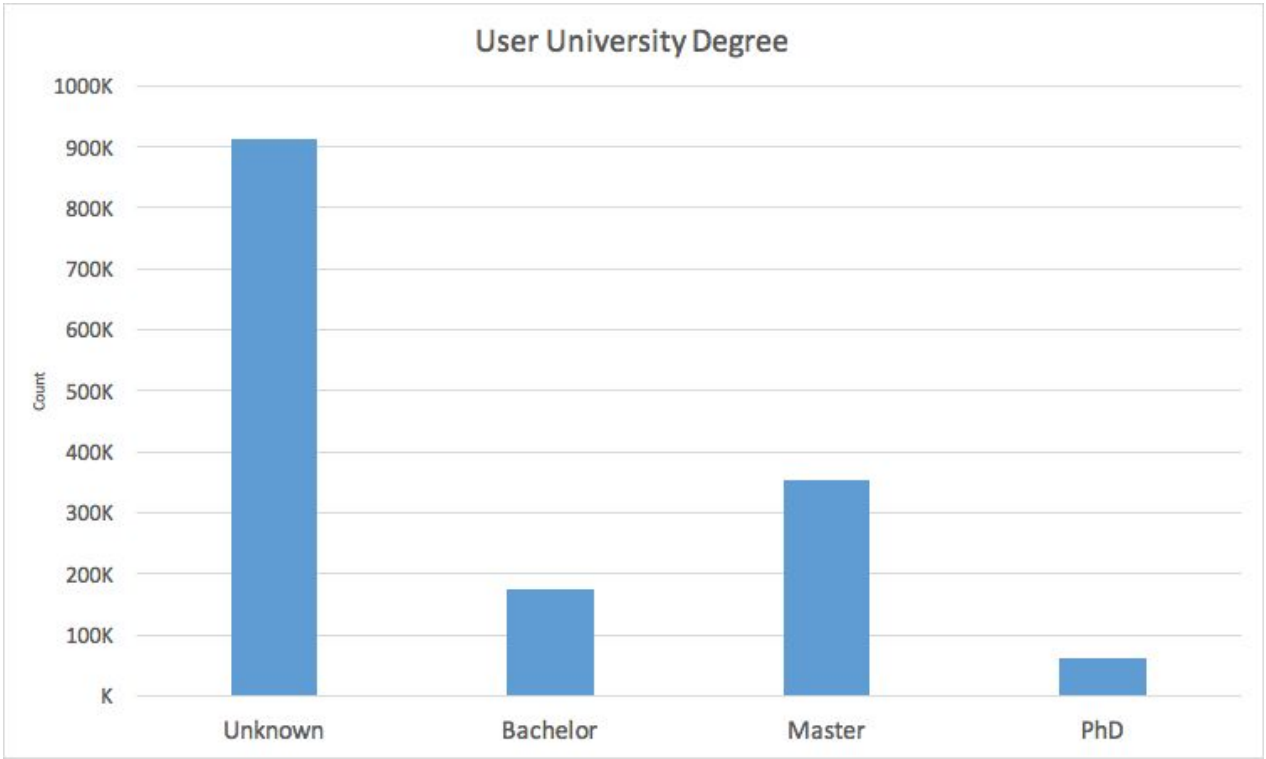
Dataset Analysis

Users



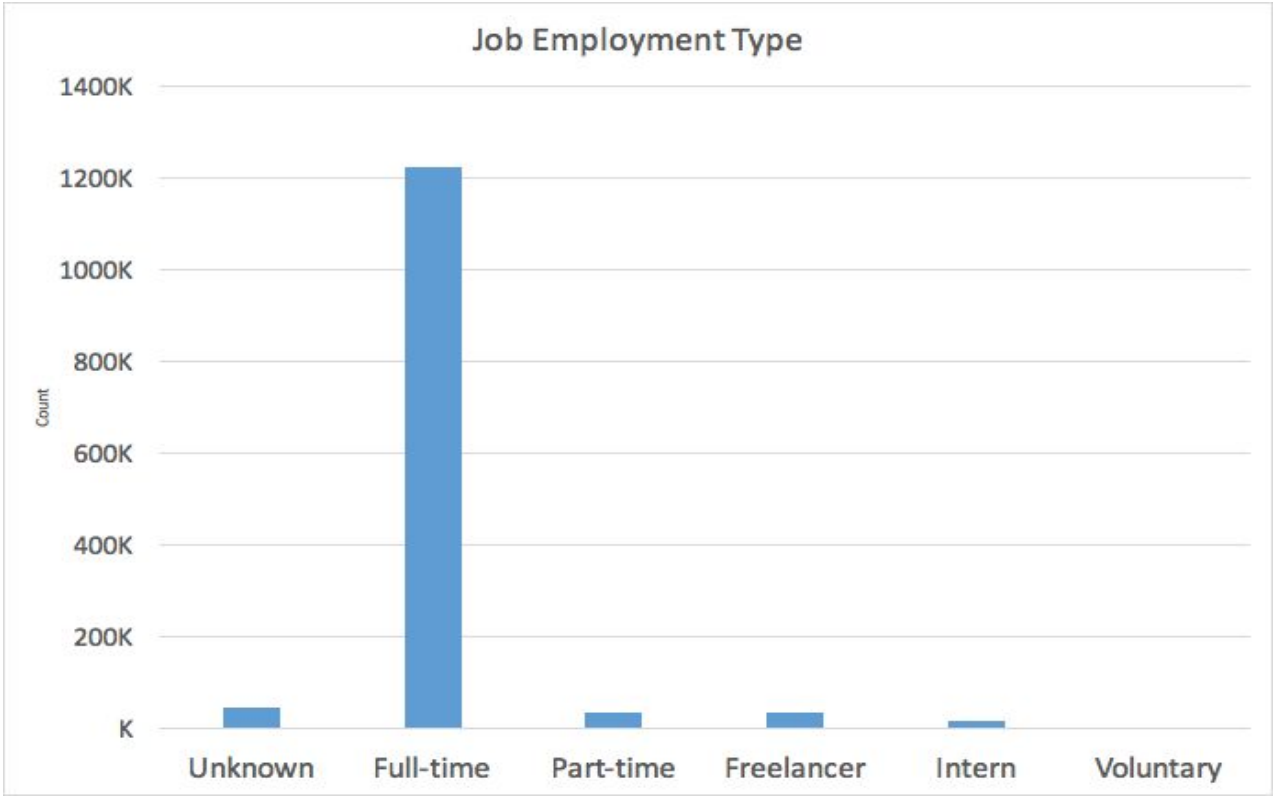
Dataset Analysis

Users



Dataset Analysis

Items



Dataset Analysis

Items

Preprocessing

Majority of features are categorical -- not ideal for clustering or finding similarity

1-hot encoding

- Users: 10 → 110 features
- Items: 8 → 87 features
- [Python] Pandas

1. Introduction
2. Data sets
3. Analysis and Preprocessing
4. **Methodologies**
5. Evaluation
6. Conclusion and Future work

Impressions

XING's existing Recommendation System

- Cannot apply traditional techniques -- No explicit user feedback
- No guarantee that the item was in the "viewport" of the user

- Sort recent items based on their impression frequencies
- Does not include all 150K test users; 18K new users
- Not thorough but a good start

Interactions

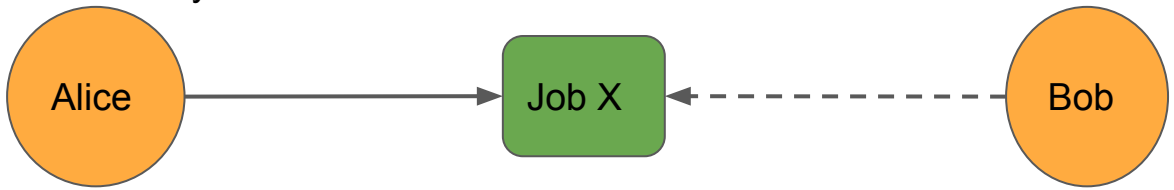
Users' feedback on XING's existing Recommendation System

- Contains information about user intent:
  - 1 = clicked, 2 = bookmarked, 3 =clicked on apply, 4 = deleted

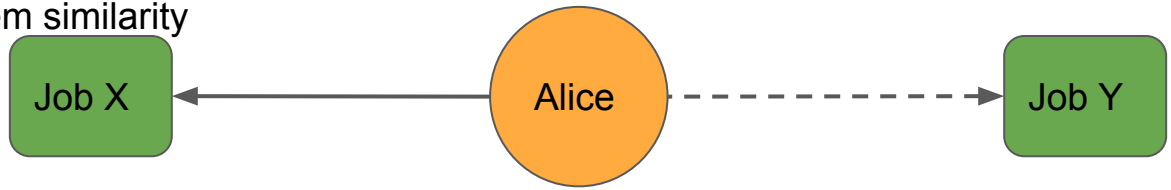
- Recommend the active jobs marked as 3, followed by 2 and 1
- Ignore the items that have been rated 4 by a user
- Ties can exist

Collaborative Filtering

- Leverage the notion of homophily in
  - User-user similarity



  - Item-item similarity



- Challenge: similarity computation, sparsity, gray/black sheep

Collaborative Filtering

How to we know if two items or users are similar?

K-means Clustering

- Number of clusters: 100, 1K, 5K
- Distance measure : Euclidean distance
- Library: SciPy.kmeans2

Collaborative Filtering

Cosine Similarity

- Similarity $(U_i, U_j) = U_i . U_j / |U_i| \ |U_j|$
- Similarity $(I_i, I_j) = I_i . I_j / |I_i| \ |I_j|$

Q: What are we using?

A: Both. Cosine similarity, but limited to the cluster

Collaborative Filtering

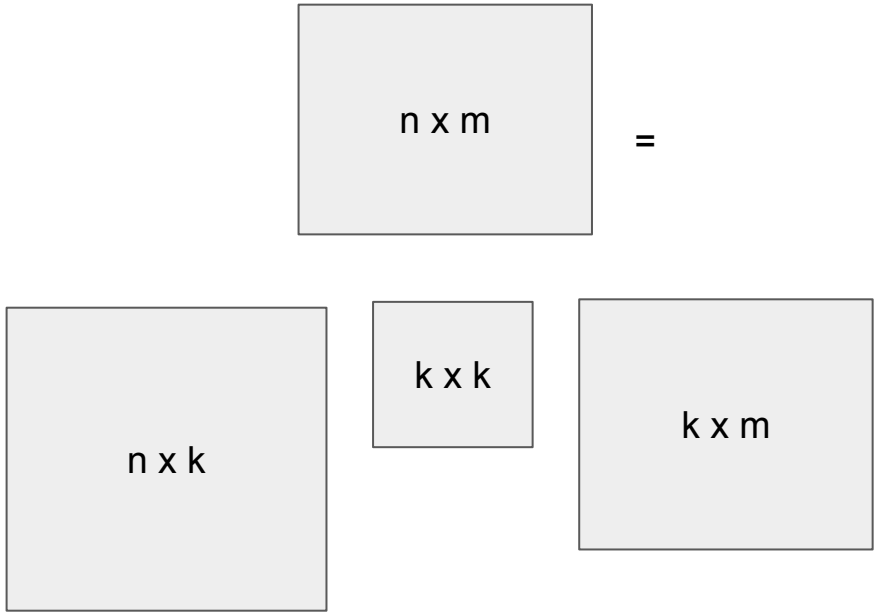
Populating the Interaction table

User-User similarity $$I\ (u,i) = \frac{\sum_{v \epsilon C} S(u,v) I(v,i)}{\sum_{v \epsilon C} S(u,v)}$$

Item-Item similarity $$I\ (u,i) = \frac{\sum_{j \epsilon C} S(i,j) I(u,j)}{\sum_{j \epsilon C} S(i,j)}$$

Singular Value Decomposition (SVD)

- Construct a user-item matrix for all users using the interaction data
- Very sparse matrix since there are over 1 million users and 300,000 active jobs
- Given a user, multiply the appropriate row with the two matrices:
  - (1 x k)*(k x k)*(k x m)
  - K = 50, 100 etc.



User and Item similarity

- Assign a score for each job and rank them based on their value
- Impressions: Frequency of item shown to a user
- Interactions: Value * w_1, except for (‘4’)
- Users and Items: Overlap between job roles * w_2
- Other components:
  - User_career_level == Item_career_level (+ w_3 points)
  - User_discipline_level == Item_discipline_level (+ w_4 points)
  - User_industry == Item_industry (+ w_5 points)
  - User_region == Item_region (+ w_6 points)

Learn weights from the data

- Treat it as a regression problem to learn the weights
- The possible output values are 0 (4), 1, 2 and 3
- Each user-item pair is a data point
- Features are:
  - Number of items overlap in job roles
  - If career level matches then 1 else 0
  - If discipline matches then 1 else 0
  - If industry matches then 1 else 0
  - If region matches then 1 else 0
- Used Linear Regression

1. Introduction
2. Data sets
3. Analysis and Preprocessing
4. Methodologies
5. **Evaluation**
6. Conclusion and Future work

Evaluation

- Function of Precision@k and Recall

```
function score(S, T) = {
        score = 0.0
        foreach (u, t) in T:                    //t = set of relevant items for user u
                r = S(u)                        //r = ordered list of recommended items for user u
                score += 20 * (precisionAtK(r, t, 2) + precisionAtK(r, t, 4) + recall(r, t) +
                        userSuccess(r, t)) + 10 * (precisionAtK(r, t, 6) + precisionAtK(r, t, 20))
                return score
}
```

- userSuccess(r, t) = 1 if at least one relevant item was returned; else 0
- Maximum 5 submissions per day

Evaluation

Baseline Score: **26,857.38**. (Rank: **57**)

Strategy

- Score = # overlaps in user job roles and item title * 3
  + # overlaps in user job roles and item tag * 2
  + I (discipline and region matches) * 2
  + I (industry and region matches) * 1
- Only consider active items and items.career_level == users.career_level

Evaluation

Only Interaction Dataset

Score: **180,112.15** (Rank: **47**)

Strategy

- For each user, sort job items in descending order of interaction type
- Recommend the active jobs marked as 3, followed by 2 and 1
- Ignore the items that have been marked 4 by a user

Evaluation

Only Impression Dataset

Score: **279,062.28** (Rank: **32**)

Strategy

- Sort job items according to their impression frequency
- Consider only recent (>2015 week 45) jobs that are still active

Evaluation

Combining all Datasets

Score: **386,703.38** (Rank: **23**)

Strategy

- Score = impression frequency
  - \+ # overlaps in user job roles item-title * 15
  - \+ I (career level matches) * 12
  - \+ I (discipline ID matches) * 10
  - \+ I (industry ID matches) * 5
  - \+ I (region matches) * 2
  - \+ Interaction score * 10

Evaluation

Combining all Datasets

Score: **456,487.86** (Rank: **16**)

Strategy

- Score = impression frequency
    - \+ # overlaps in user job roles item-title * 10
    - \+ I (career level matches) * 12
    - \+ I (discipline ID matches) * 10
    - \+ I (industry ID matches) * 5
    - \+ I (region matches) * 2
    - \+ Interaction score * 100

Evaluation

Combining all Datasets

Score: **458,017.20** (Rank: **15**)

Strategy

- Weights learn using linear regression

Evaluation

Populating sparse interaction matrix using

1. SVD with rank 50 approximation: **34,084.99**
2. CF User-User similarity (absolute): **77,859.45**
3. CF User-User similarity (weighted): **85,491.27**

Feed the updated interaction in our previous model -- Work in Progress

1. Introduction
2. Data sets
3. Analysis and Preprocessing
4. Methodologies
5. Evaluation
6. **Conclusion and Future work**

Conclusion

Working on an ongoing RecSys 2016 challenge to build a job recommendation system for XING.

- Studied the data thoroughly and got some really good insights
- Combined all datasets and heuristically assigned weights to achieve good results
- Applied conventional learning approaches to learn the weights
- Applied methods like SVD and CF and are in process of integrating with our overall model

Future Work

- User behavior analysis using temporal information
- Handling 18K new users more appropriately
- Exploring non-linear models that will better suit our system

Suggestions/Questions?