PAYROLL

PAYCHEX®

Payroll | Benefits | HR | Insurance

RECOMMENDATION ENGINE

Our Vision

Vision

Implement a recommendation system to suggest the prioritized purchasing list of high-level products for each client

Goals

- Cluster clients based on K-Modes model applying clients features
- Run ALS Model in each cluster to generate the recommendation
- Develop the evaluation metric

Members

Data Scientist
Hyungkyu Lim

Research Engineer
Andrea Clark-Sevilla

Delivery Consultant
Zongyan Yang

Research Engineer
Sahar Hajiseyednasir

Data Overview

Dataset Size: 382,525 rows × 41 columns

Description: Each row represents a distinct client, characteristic features, and the corresponding purchased products

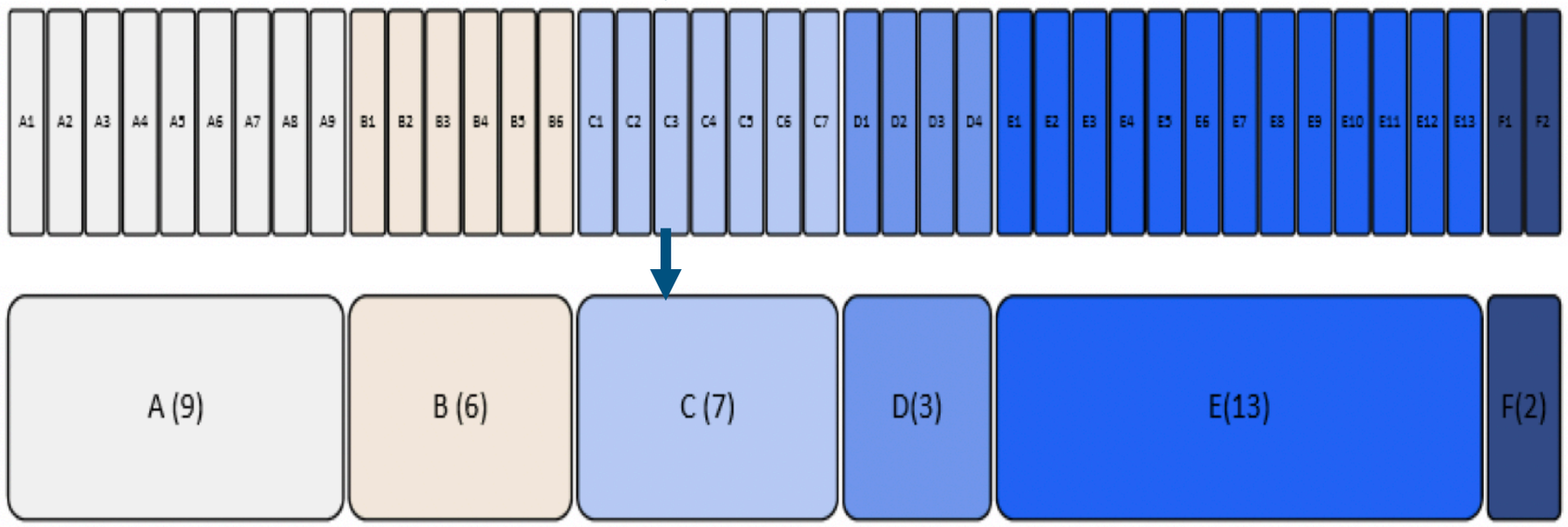
	Client_Id	A1	A2	A3	A4	A5	D1	B1	D2	A6	...	B5	E9	E10	E11	E12	E13	B6	Rep.Level	Size	Industry
0	1	0	0	0	1	1	0	0	0	1	...	0	0	0	0	0	0	0	Level 4	Size 3	Industry 10
1	2	0	0	0	0	0	0	0	0	0	...	0	0	1	0	0	1	0	Level 5	Size 4	Industry 6
2	3	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	Level 5	Size 4	Industry 10
3	4	0	0	0	1	1	0	0	0	1	...	0	0	0	0	0	0	0	Level 4	Size 2	Industry 12
4	5	0	0	0	1	1	0	0	0	0	...	0	0	0	0	0	0	0	Level 4	Size 3	Industry 10

Categorical Client Features:

- Rep. Level (6)
- Size (5)
- Industry (13)

Binary Item Features:

- Not Purchased (0)
- Purchased (1)



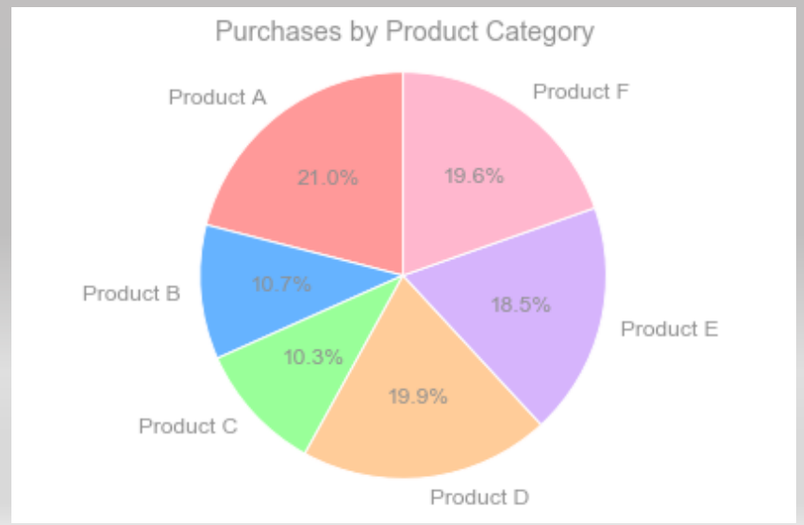
What do we have?
Low-level products

What do clients want?
High-level product groups

Exploratory Analysis

Frequency Distribution

Products A, D, E, F distinctively makes up around 20% for frequency of all the purchases, while B and C makes up 10% separately.



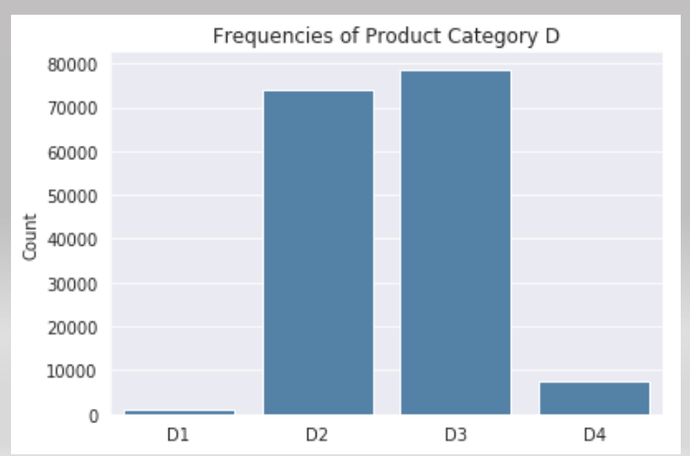
Market Basket Analysis

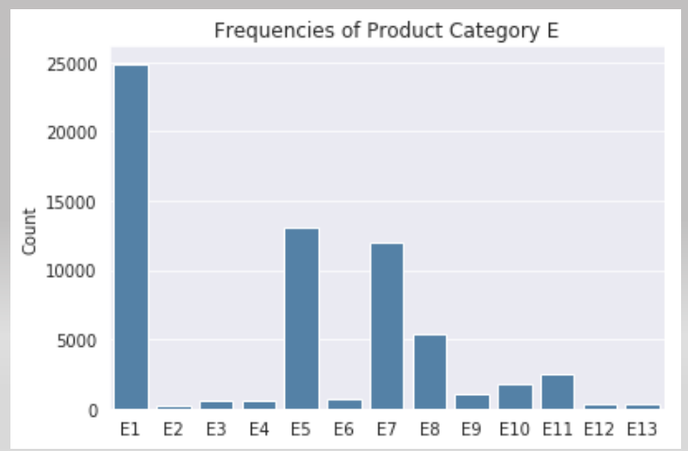
Products D, E, F occupy high frequent itemsets in transactions.

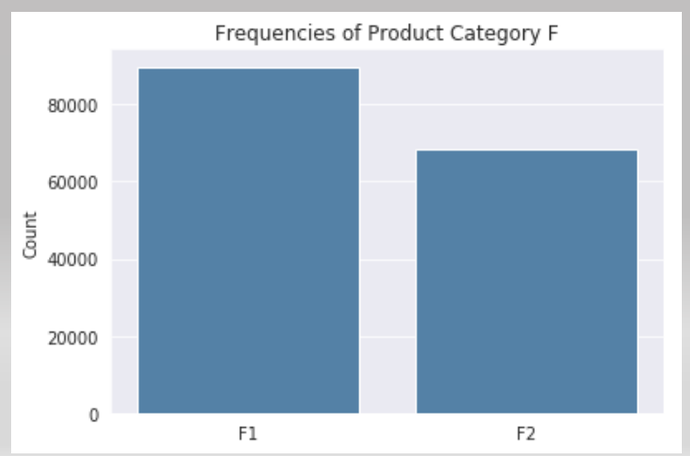
Frequent Patterns	Support
(E, F)	13%
(D, F)	12%
(D, E)	11%
(A, F), (C, E)	9%
(A, D), (A, E), (C, F),	7%
(B, A), (B, D), (B, E), (B, F) (D, E, F)	6%
(C, D)	5%

Products Frequency Overview

All product categories have **low-level products** with **no** or **very few** counts, 30% of clients have no purchase of any products







3-Step Process to Conduct the Clients Clustering

K-modes Algorithm

The k-modes algorithm tries to minimize the sum of within-cluster Hamming distance from the mode of the cluster, summed over all clusters.

Hamming Distance Metrics

Term Definition: the number of columns where the two vectors differ. Dissimilarity of binary and categorical data when Euclidean Distance can't work

Eg. :Distance of $\{\begin{matrix}1&0&1&1&1&0&1\\1&0&0&1&0&0&1\end{matrix}\}$ is 2

K-Modes Clustering

Much-needed alternative to k-means: minimize the sum of within-cluster Hamming distance from the mode consisted of "1" and "0" markers, summed over all clusters.

Step 1

Given number K, mode vectors are chosen at random.

Step 2

Observations are assigned to the Hamming-Distance-closest centroid.

Step 3

New cluster modes are calculated, each from the observations associated with a previous cluster mode.

Step 4

Steps 2 and 3 are repeated until the assignment of individuals to the closest centroid is stable at a local minimum cost.

Clustering Analysis

Dummy Coding

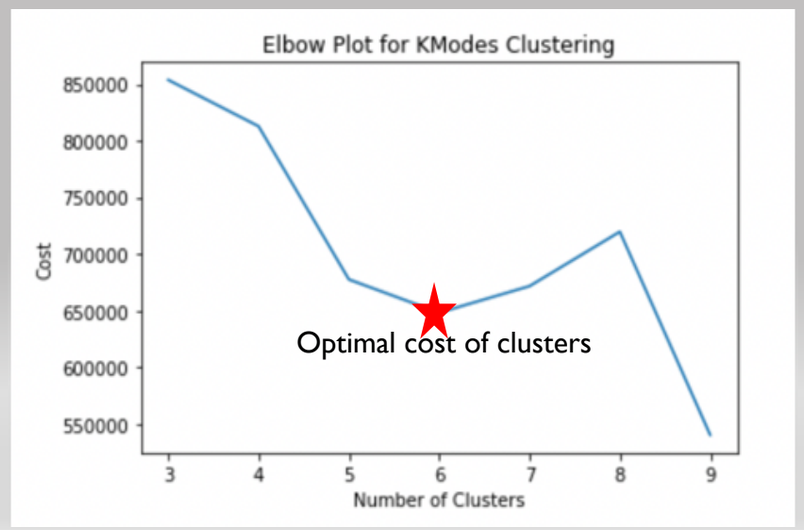
Encode the categorical clients' features into dichotomous variables. "1" represents "true" and "0" represents "false".

Rep. Level	Size	Industry
Level 1	Size 2	Industry 3

Level 1	Level 2	Size 1	Size 2	Industry 1	Industry 2	Industry 3
1	0	0	1	0	0	1

Optimal K

The optimal number of clusters is 6



Clustering Results

Most clients of the groups are in industry level 10 except the clients in group 3 (Industry level 5).

	Company Size 2	Company Size 3	Company Size 4	Company Size 5
Rep. Level 2		Group 1 127193 clients	Group 2 49057 clients	
Rep. Level 5	Group 3 46505 clients	Group 4 72702 clients	Group 5 30945 clients	
Rep. Level Unknown				Group 6 56123 clients

Group 5 Results Details

Counts for Size:	
Size1	0
Size2	0
Size3	0
Size4	30945
Size5	0
Counts for Rep. Level:	
Rep_Level1	0
Rep_Level2	0
Rep_Level3	0
Rep_Level4	0
Rep_Level5	30945
Rep_Unkown	0

Counts for Industry:	
Industry1	2538
Industry2	5786
Industry3	1888
Industry4	431
Industry5	3744
Industry6	1670
Industry7	243
Industry8	0
Industry9	2816
Industry10	6444
Industry11	148
Industry12	5192

What Alternating Least Squares Collaborative Filtering

Data Characteristics

Implicit Purchase History Data

- Doesn't directly reflect the interest of the client.
- No negative preference measured directly.
- The numerical value of implicit feedback denotes confidence, the actual purchase counts.
- Evaluation requires appropriate measures.

Client_Id	A1	A2	A3	A4	A5	D1	B1	D2	A6	...	B5	E9	E10
1	0	0	0	1	1	0	0	0	1	...	0	0	0
2	0	0	0	0	0	0	0	0	0	...	0	0	1
3	0	0	0	0	0	0	0	0	0	...	0	0	0

Collaborative Filtering

Make recommendations based on a user's product interaction history combined with interaction the history of all other users.

Basic Assumptions:

- Users with similar purchase history have common preferences.
- If a client A purchase the same product as client B, A is more likely to share B's purchase on a different product than that of a randomly chosen client.

Popular approaches:

- User-based or Item-based;
- **Alternating Least Square (ALS):** use item-based or user-based similarities to deduce unknown relationships between users and items

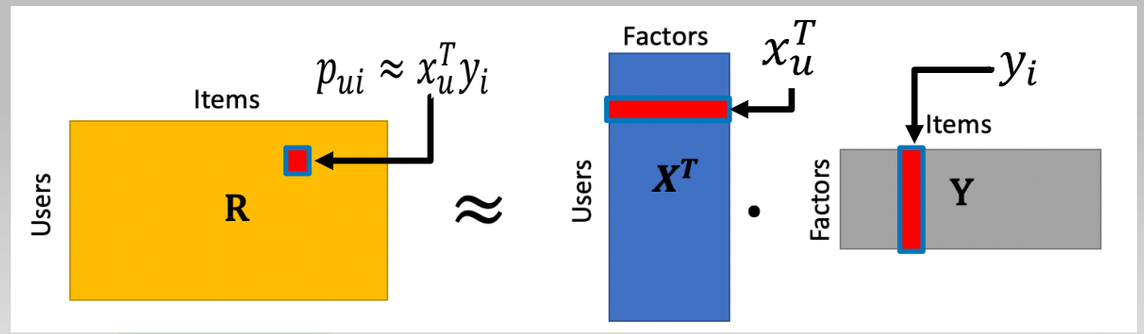
Alternating Least Squares

A matrix factorization method to uncover latent user and item feature. Through the large matrix of user/item interactions, ALS can figure out the hidden features that relate them to each other in a much smaller matrix of user features and item features.

Prediction is found by taking the inner product of the two vectors when achieving minimized user cost and item cost.

$$\hat{p}_{ui} = x_u^T y_i$$

$$Cost = \sum_{u,i \in R} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2)$$



y_i = latent item-factors vector for user u
x_u = latent user-factors vector for item i
p_{ui} = preference for item i by user u
c_{ui} = confidence in observing p_{ui}

How: Model Construction

Data Preprocessing

Take all the purchasing histories for each client and put these into ALS matrix format.
Items with a larger number of purchases by a client does not carry more weight in our preference matrix of purchases.

Client ID	Product	Quantity
2	D	1
2	E	2

Binary Label:
1 → purchased
0 → not purchased

Client ID	Product	Preference
2	A	0
2	B	0
2	C	0
2	D	1
2	E	1
2	F	0

Create a Training and Validation Set

Hide a certain percentage of the client/item interactions chosen at random during the training phase. Then, check during the test phase whether the items that were recommended the user actually ended up purchasing.

- The training set consists of 20% masked interactions, in which these were converted to a 0 (no interaction)
- The test set is simply a copy of the original data

E.g.

1	0	1	0	0
0	1	1	0	1
0	0	0	1	0
1	0	0	0	0
0	1	1	0	0

Train

1	0	1	0	0
0	0	1	0	1
0	0	0	0	0
1	0	0	0	0
0	1	0	0	0

Test

1	0	1	0	0
0	1	1	0	1
0	0	0	1	0
1	0	0	0	0
0	1	1	0	0

- That the users frequently ended up purchasing the items most recommended to them can conclude the system is working.

Model Implementation & Results

A ranked list of items recommended for each client.
The order of item represents the priority, so assign more "weight" to relevant items appearing higher up on the list than those that appear lower down.

Model Evaluation

Step 1. Single-Item Precision

For each recommended item, the Precision

$$Precision = \frac{\#\ of\ relevant\ recommendations}{total\ \#\ of\ recommendations}$$

Step 2. Single-Client Precision

For each client with N recommended items and M actually-purchased items, the Average Precision

$$AP@N = \frac{1}{M}\sum_{k=1}^{N} P(k) * rel(k)$$

Step 3. Single-Cluster Precision

For all the users in each cluster, the Mean Average Precision

$$MAP@N = \frac{1}{|U|}\sum_{u=1}^{U} AP_u$$

Step 1. Example

Client #2: Mask Product C, Precision = $\frac{\text{C, E, D}}{\text{F, C, E, D, B}} = \frac{3}{5}$

Actual Purchase	C, E, D, A
Training Set	Mask, E, D, A
Recommendation	F, C, E, D, B

Step 2. Example 1

Recommendation List				
1st	2nd	3rd	4th	5th
0	1	1	1	0

Precision @ k's				
1st	2nd	3rd	4th	5th
0	1/2	2/3	3/4	0

AP@5

(1/3)* [(1/2) + (2/3) + (3/4)] ≈ 0.64

Step 2. Example 2

Recommendation List				
1st	2nd	3rd	4th	5th
1	1	1	0	0

Precision @ k's				
1st	2nd	3rd	4th	5th
1/1	2/2	3/3	0	0

AP@5

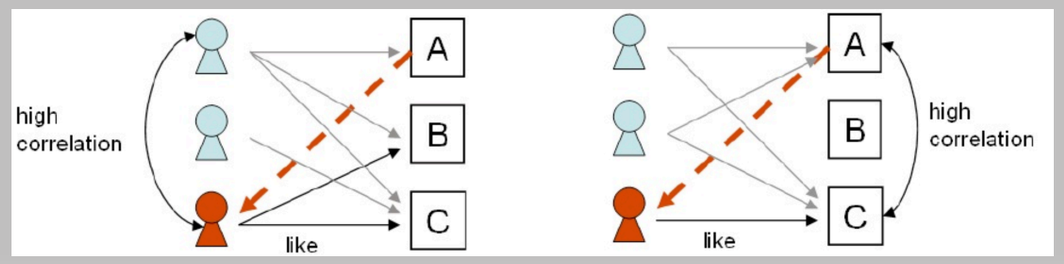
(1/3)* [(1/1) + (2/2) + (3/3)] = 1

Step 3. Results

Mean Average Precision @ 5					
Group 1	Group 2	Group 3	Group 4	Group 5	Group 6
35.9%	50.7%	31.4%	46.5%	33.9%	42.1%

Key Insights & Challenges

1. Finding the right algorithm for dealing with implicit client feedback
- Since we do not have explicit feedback in the form of ratings, many traditional collaborative-filtering approaches do not work for our problem
- Evaluating a ranked list of recommendations instead of dealing with a predicted rating requires a more specialized metric- cannot just use MSE!
- **Challenge:** Is an item-based model or a user-based model more appropriate for our problem?



K-modes → ALS

Cluster the clients first and then use purchase history to make recommendations
What is a good number for k?

2. Clustering reduced the complexity of the problem
- By narrowing down the client space, we can give more targeted recommendations
- Faster computationally
- **Challenge**: Choosing the "correct" number of clusters is challenging and not well-defined

3. Using average mean precision as a metric allows us to evaluate the model on relevancy
- The order in which the recommendations are made matters, as we want relevant items to show up first
- **Challenge:** Does not work for evaluating recommendations for customers that have not purchased anything before, i.e., not based on client's features entirely

4. **Challenge:** We don't know what are products and clients' features are so we can't really interpret our results intuitively.

1. We would like to have another model against which to compare ALS
- We can explore frequent model association with Apriori Algorithm to use another item-based approach within each cluster.
2. Making recommendations for customers that have no purchase history
- Doing A/B testing with our models on these customers would be beneficial to see if our models are learning accurate associations between users and items

PAYCHEX®

Payroll • HR • Retirement • Insurance

THANK YOU!
QUESTIONS?

UNIVERSITY of ROCHESTER