TUTORIAL

Generative AI

Ravi Ranjan
Principal Data Scientist
Publicis Sapient

Graph Data-Driven Recommendation Systems Empowered by Generative AI

ODSC APAC

FREE | VIRTUAL
AUGUST 13

About the Speaker

in https://bit.ly/ravi-ranjan-03

Ravi Ranjan is working as Principal Data Science at Publicis Sapient (India), specializing in creating scalable ML solutions. A certified Google Cloud Architect, he has extensive experience in designing and implementing AI and ML systems, including scalable recommendation platforms. In addition, he actively contributes to and is a member of Kubeflow, an ML platform by Google.

publicis
sapient

Session Logistics

1. Recording of the session will be accessible on the ODSC platform.
2. The presentation and source code will be available on GitHub. [https://bit.ly/ODSC-APAC-202
3. We will address the Q&A at the end of the session.
4. Connecting to the speaker [Please send an introductory note in a LinkedIn invite] https://bit.ly/ravi-ranjan-03
5. Don't forget to tweet and share the session with **#ODSCAPAC**

publicis
sapient

Learning Outcome

- Innovative fusion of Generative AI with graph data to create personalised, timely, and immersive recommendations.
- How to build community detection on graph data?
- How to perform trend analysis using customer interaction data?

Today's Session

1. The Imperative of Modern Recommendation Systems
2. Recommendation Systems Redefined: Graph Data as the Key Differentiator
3. How does GenAI further elevate the power of Recommendation Systems?
4. Demo
5. QnA

Section - 1

What is Recommendation System?

"Serve the relevant items to users in an automated fashion to optimize short- and long-term business objectives."

Recommendation System is everywhere!

Amazon

Product, Video and music recommendations

Netflix

Video recommendation and artwork personalisation

Facebook

Content, ad, account and entity recommendations

Spotify

Music and podcast recommendations

Youtube

Video content and ad recommendations

Myntra

Product and personalized size recommendations

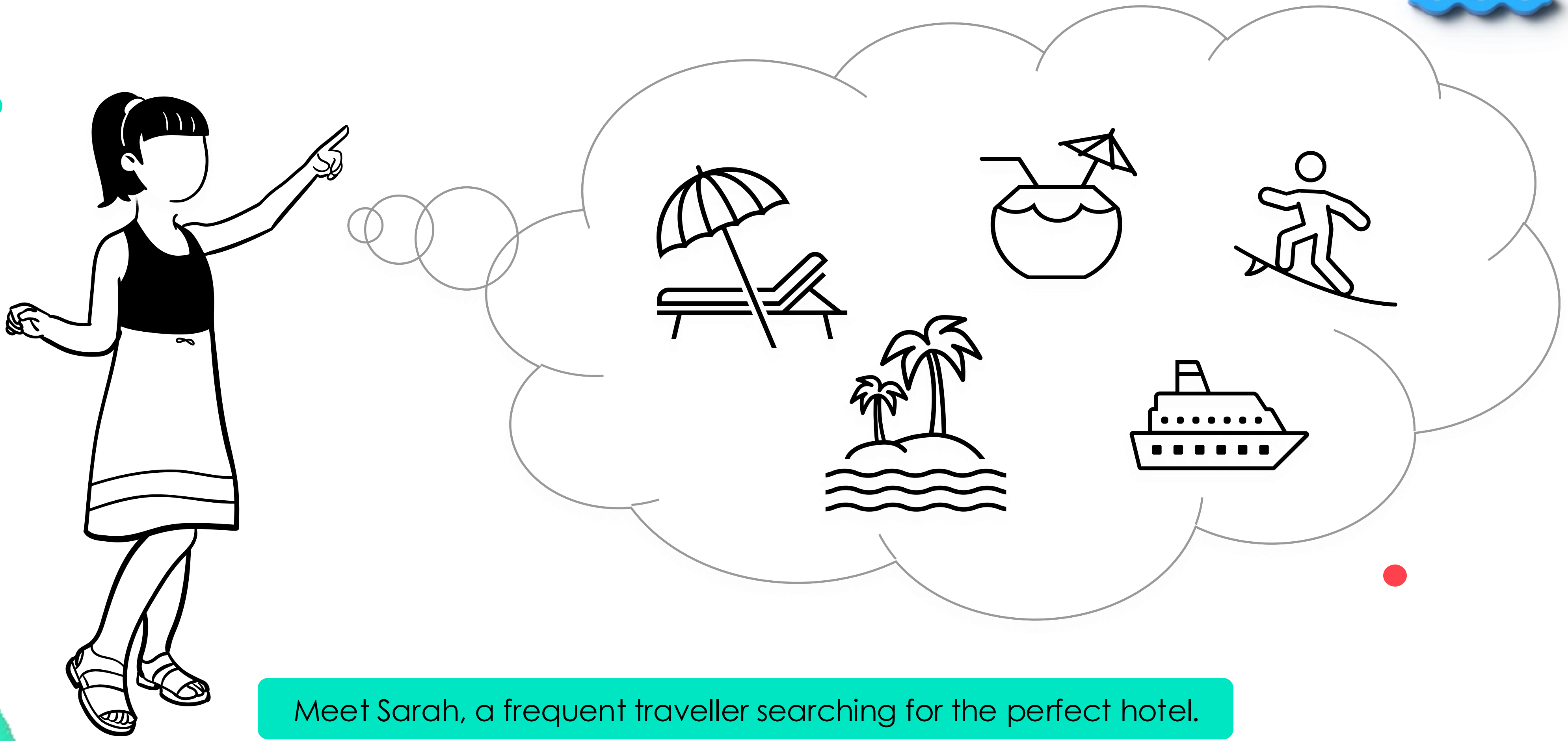
Coursera

Courses and learning path recommendations

LinkedIn

Connections, job and content recommendations

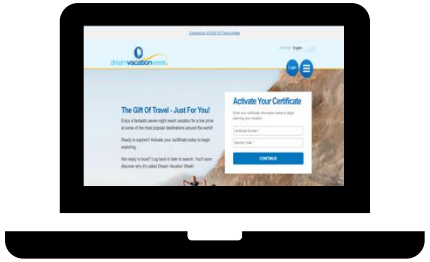
Case Study – Online Travel Booking



Meet Sarah, a frequent traveller searching for the perfect hotel.

Traditional Recommendation System

Sarah visits a popular travel website and searches for hotels



Dreamvacay.com

	Hotel 1	Hotel 2	Hotel 3	Hotel 4	Hotel 5
User 1	X		X		
User 2		X		X	
User 3	X	X			X
User 4			X		X
User 5			X	X	

User Interaction Matrix

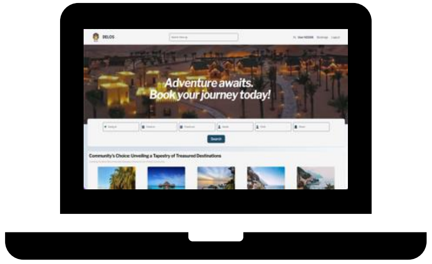
- Complex Relationships Ignored
- Long-Tail Problem
- Cold Start Problem
- Evolving Preferences

Section - 2

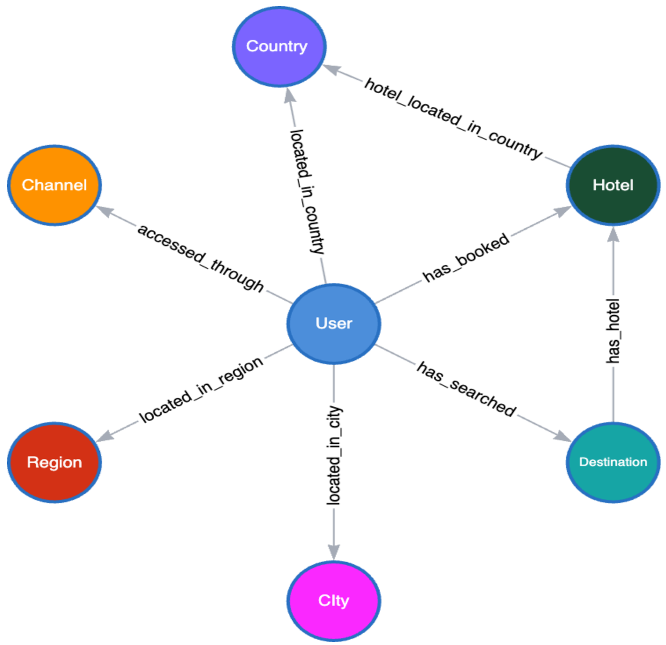
Recommendation Systems Redefined: Graph Data as the Key Differentiator

Graph Based Recommendation System

Sarah tries a new travel website with a graph-based recommendation system.



Recommend



- Real-time recommendations are adjusted based on current data and user behaviour.
- Richer Data Representation
- Context-Aware Recommendations
- Detecting Community Structure

What

Hotel Metadata

How

Interaction Data

Where, When

Context data

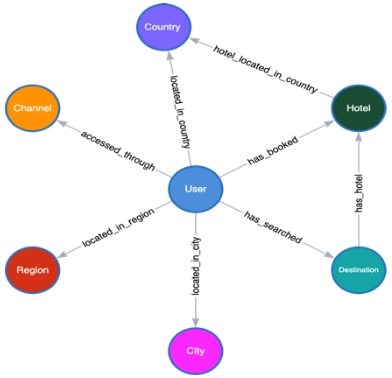
Who

User Profiles & Intent

Enrichment Layer

- Entity Extraction
- Entity Linking
- Image Features

Hotel Customer Graph



Intelligence Layer

Insights

Graph AI

Gen AI LLM

Use cases

- Traveller's Choice
- Most Booked Destinations
- Personalized Recommendation
- Top Rated Accommodations
- Trend Analysis
- Community Based Recommendations

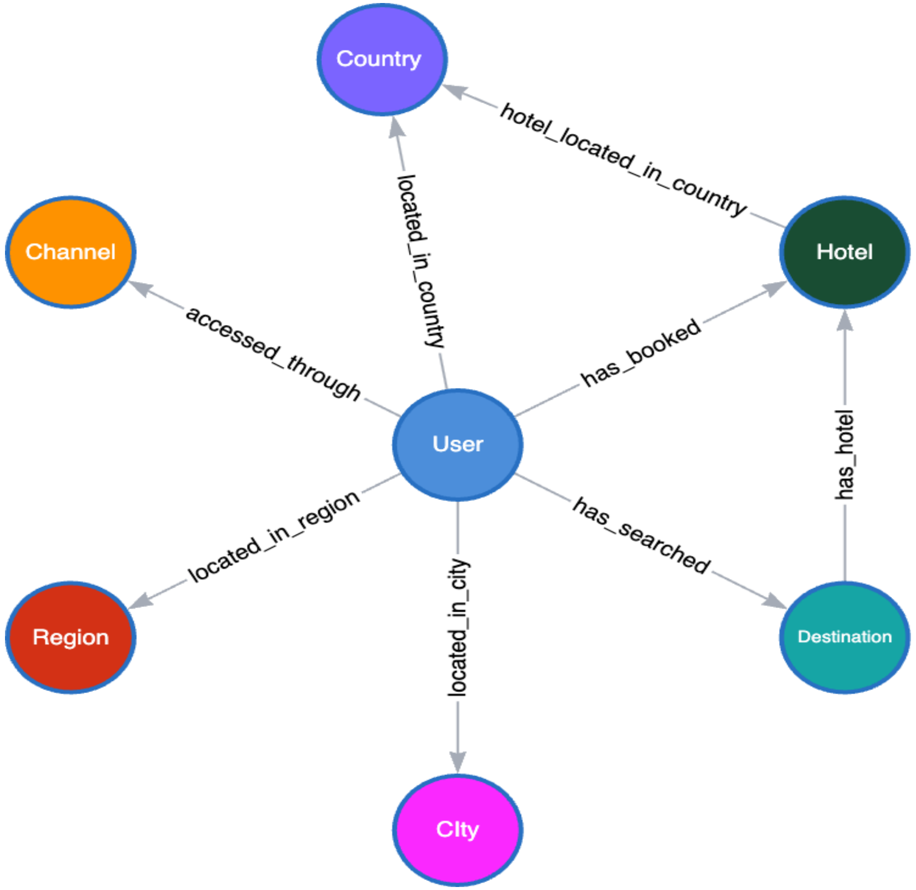
Why graph for recommendations?

Flexible: *Products having common features with more weightage (based on sales) are given priority during recommendation*

Explainability: *Uncovers relationships between different products using common linkages to provide optimum recommendations which can be easily explained*

Relevance: *Uses keywords/entities to reach relevant products in the graph*

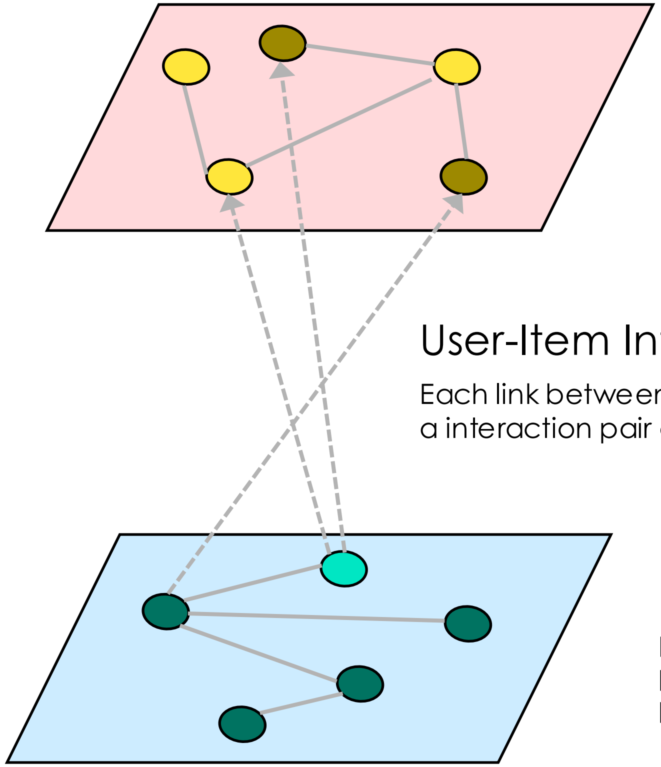
Agility: Uses hierarchical relationships to reduce search space and efficiently provide recommendation in different levels



2 Layer Graph

Item Layer

Each node represents an item. Each link between the nodes represents similarity between two items



User-Item Interaction

Each link between the two layers represents a interaction pair of item and user

User Layer

Each node represents a user. Each link between the users represents similarity between two users.

Recommend (High Level Design)

Data Sources

Hotel Metadata

Interaction Data

Context data

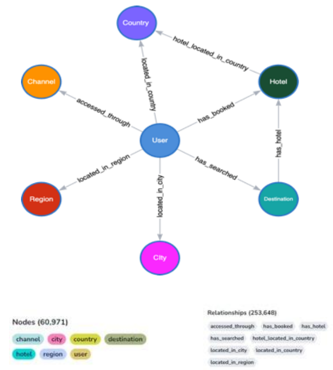
User Profiles & Intent

Feature Extraction and Enrichment

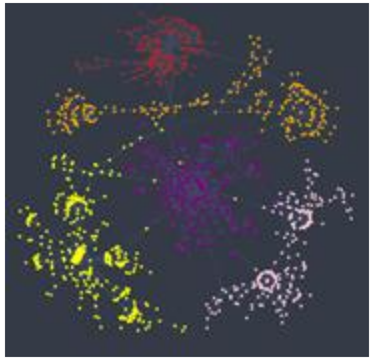
- Hotel Entities
- Users Entities
- Channel Preference
- Booking Information
- Search Information
- Ratings
- User Segment

Data Layer

Graph Representation



Community Representation



Intelligence

- User Similarity
- Frequently Booked Hotel
- Top Recommendation
- Community Detection
- Current Trends

Graph AIML (Rank)

- Cypher generation using Language Model
- Generative Hotel Description

GenAI LLM

Realtime Feedback

Consumption

- Dashboards
- Web Apps

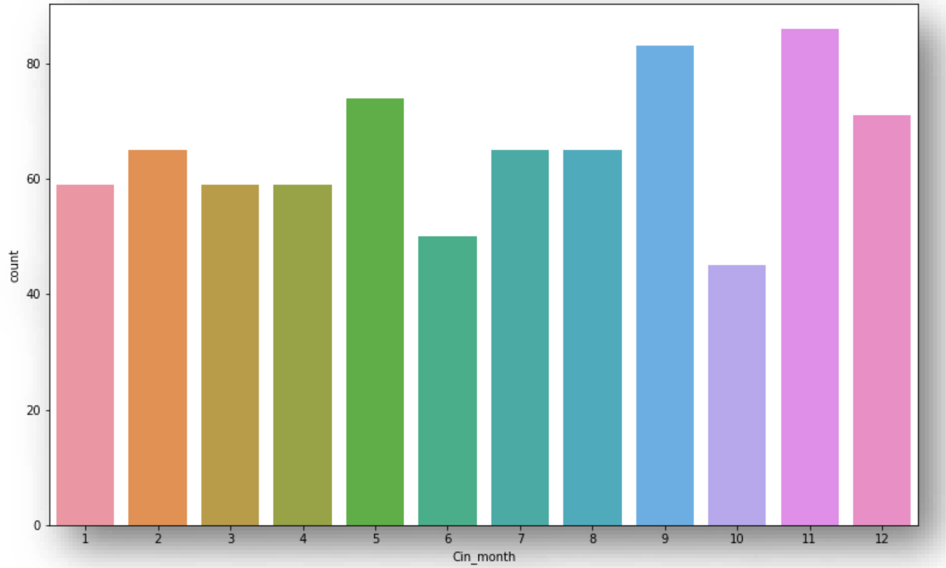
Data Preprocessing

Data Sampling

- The Expedia dataset had about 37 million data points.
- To optimize usability, feasibility, minimize bias, and facilitate data exploration, we employed sub-sampling, selecting a representative subset of data points.
- We sub-sampled about 0.1% of data points from every month to get about 37 thousand data points.

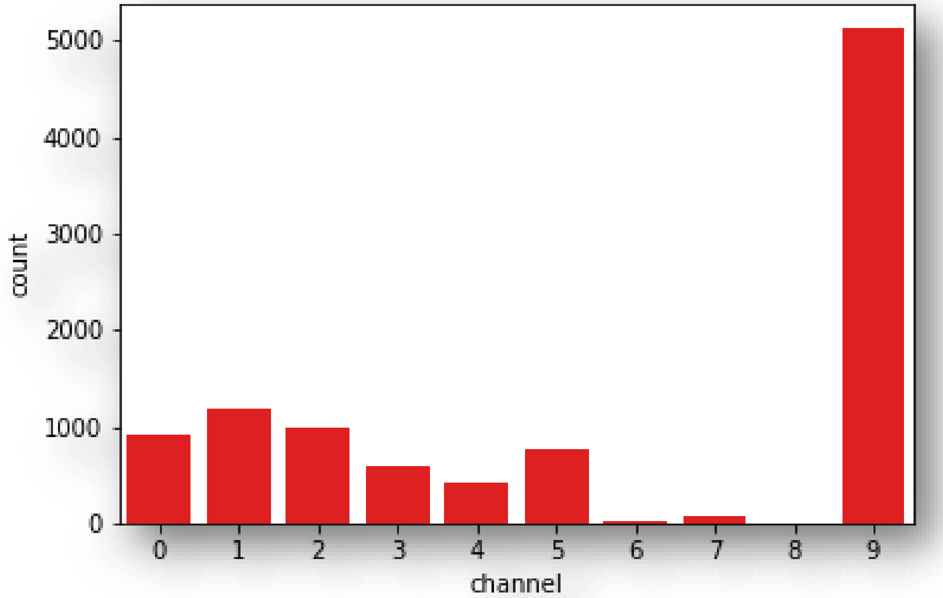
	date_time	site_name	posa_cont	user_locat	user_locat	user_locat	orig_desti	user_id	is_mobile	is_package	channel	srch_ci	srch_co
279311	14-08-2013 05:05	2	3	66	351	13431		1004051	0	0	2	27-04-2014	28-04-2014
462337	06-08-2013 18:25	2	3	66	363	24151		408068	0	0	9	26-08-2013	29-08-2013
97227	13-08-2014 08:30	37	1	69	747	20836		380954	1	0	1	01-01-2015	05-01-2015
463486	10-08-2013 08:11	2	3	198	208	54488		411698	1	0	0	08-09-2013	09-09-2013
40586	28-08-2014 10:14	2	3	66	348	25443	9.3326	152970	0	0	9	29-08-2014	30-08-2014
116747	23-08-2014 23:43	2	3	66	356	22202	536.0621	437458	1	0	9	28-08-2014	29-08-2014
432062	12-08-2014 14:16	2	3	66	220	19416	4870.723	321691	0	1	9	26-10-2014	30-10-2014
149561	22-08-2014 08:13	2	3	66	363	12346	84.2274	573429	0	0	9	21-11-2014	22-11-2014
262322	27-08-2014 18:19	2	3	66	348	53377	1561.349	942802	0	1	8	10-04-2015	17-04-2015
513314	18-08-2014 10:28	2	3	66	459	39300	180.5792	574017	0	0	0	19-08-2014	20-08-2014
523676	03-08-2014 05:16	2	3	231	88	2972		604242	0	0	9	13-08-2014	14-08-2014
553125	24-08-2014 14:30	2	3	66	226	42300	5162.576	677608	0	0	9	03-03-2015	06-03-2015
351852	20-08-2014 00:45	13	1	46	244	33092		34547	0	0	3	21-08-2014	25-08-2014
80108	01-08-2013 05:11	2	3	66	331	54953	316.7613	325679	0	0	9	05-08-2013	07-08-2013
320167	14-08-2014 18:16	2	3	66	337	25205		1113858	0	0	9	15-08-2014	16-08-2014
72777	16-08-2013 12:12	2	3	66	363	32842	2506.251	288887	0	0	9	01-09-2013	03-09-2013
339158	04-08-2014 17:24	2	3	66	220	35388	164.8122	1168135	0	0	0	06-08-2014	08-08-2014
245916	22-08-2014 09:51	2	3	66	174	5938	82.0193	887324	0	0	0	23-08-2014	24-08-2014

Data Exploration



Number of Hotels Booked Per Month

The number of hotels booked per month is useful in finding seasonal trends.



Channels

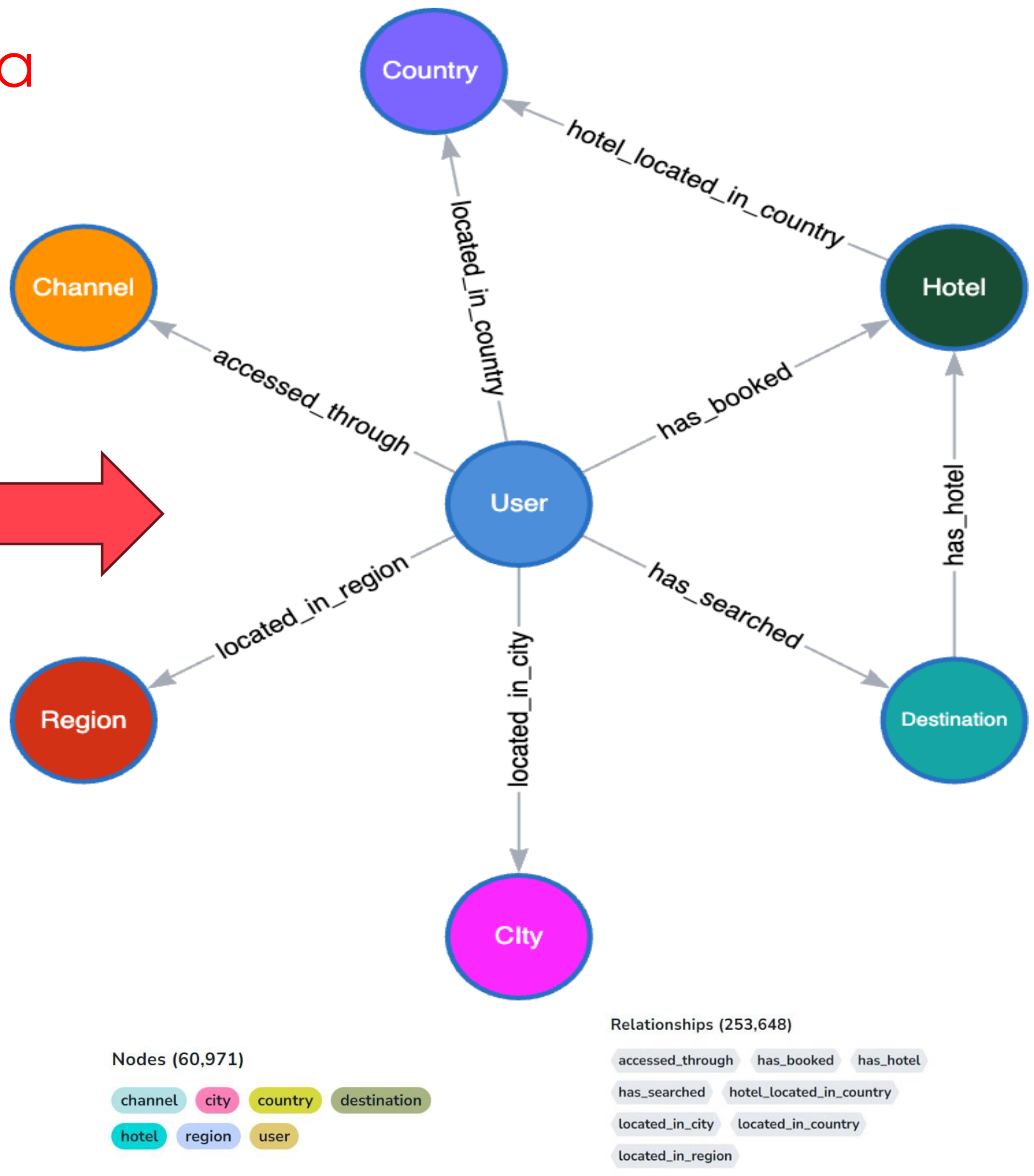
0 - Email
1 - Content
2 - Social Media
3 - Traditional
4 - Affiliate
5 - Influencer
6 - Events
7 - Word-of-Mouth
8 - Podcats
9 - SEO

Number of Booking through a given channel

Analyse the marketing channel that brings us the most bookings

Converting Tabular Data to Graph Data

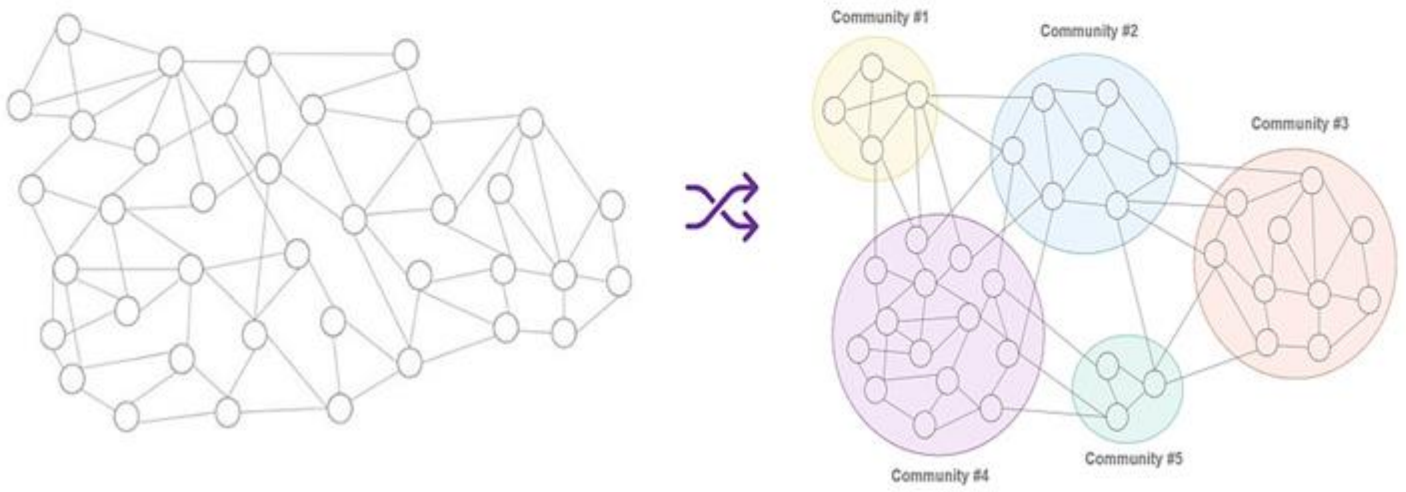
	date_time	site_name	posa_cont	user_locat	user_locat	user_locat	orig_desti	user_id	is_mobile	is_packag	channel	srch_ci	srch_co
279311	14-08-2013 05:05	2	3	66	351	13431		1004051	0	0	2	27-04-2014	28-04-2014
462337	06-08-2013 18:25	2	3	66	363	24151		408068	0	0	9	26-08-2013	29-08-2013
97227	13-08-2014 08:30	37	1	69	747	20836		380954	1	0	1	01-01-2015	05-01-2015
463486	10-08-2013 08:11	2	3	198	208	54488		411698	1	0	0	08-09-2013	09-09-2013
40586	28-08-2014 10:14	2	3	66	348	25443	9.3326	152970	0	0	9	29-08-2014	30-08-2014
116747	23-08-2014 23:43	2	3	66	356	22202	536.0621	437458	1	0	9	28-08-2014	29-08-2014
432062	12-08-2014 14:16	2	3	66	220	19416	4870.723	321691	0	1	9	26-10-2014	30-10-2014
149561	22-08-2014 08:13	2	3	66	363	12346	84.2274	573429	0	0	9	21-11-2014	22-11-2014
262322	27-08-2014 18:19	2	3	66	348	53377	1561.349	942802	0	1	8	10-04-2015	17-04-2015
513314	18-08-2014 10:28	2	3	66	459	39300	180.5792	574017	0	0	0	19-08-2014	20-08-2014
523676	03-08-2014 05:16	2	3	231	88	2972		604242	0	0	9	13-08-2014	14-08-2014
553125	24-08-2014 14:30	2	3	66	226	42300	5162.576	677608	0	0	9	03-03-2015	06-03-2015
351852	20-08-2014 00:45	13	1	46	244	33092		34547	0	0	3	21-08-2014	25-08-2014
80108	01-08-2013 05:11	2	3	66	331	54953	316.7613	325679	0	0	9	05-08-2013	07-08-2013
320167	14-08-2014 18:16	2	3	66	337	25205		1113858	0	0	9	15-08-2014	16-08-2014
72777	16-08-2013 12:12	2	3	66	363	32842	2506.251	288887	0	0	9	01-09-2013	03-09-2013
339158	04-08-2014 17:24	2	3	66	220	35388	164.8122	1168135	0	0	0	06-08-2014	08-08-2014
245916	22-08-2014 09:51	2	3	66	174	5938	82.0193	887324	0	0	0	23-08-2014	24-08-2014



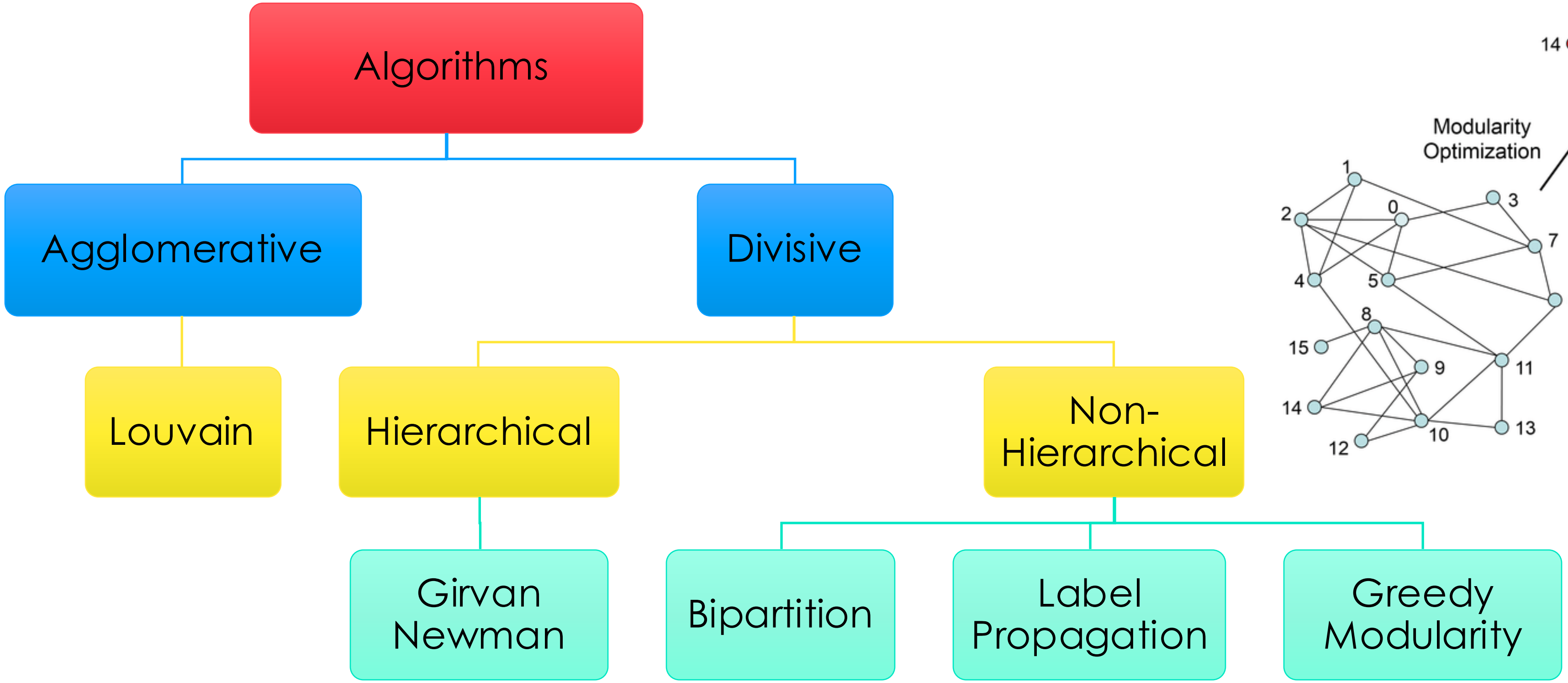
Community Detection

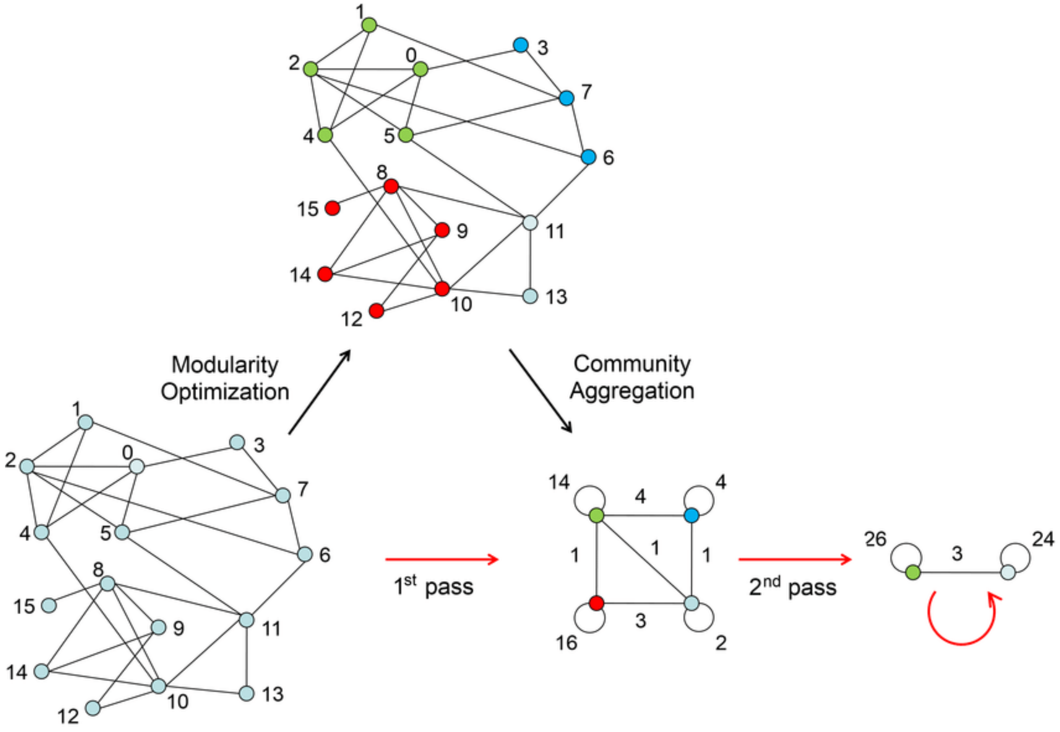
Importance of Community Detection in Graphs and its Relevance in Recommendation Systems

- Provides insights into structure and patterns by identifying densely connected groups of nodes within networks.
- Helps personalize recommendations.
- Mitigates the cold-start problem for new users.
- For example, in Social Media Platforms communities represent groups of friends with similar hobbies.



Community Detection Algorithms



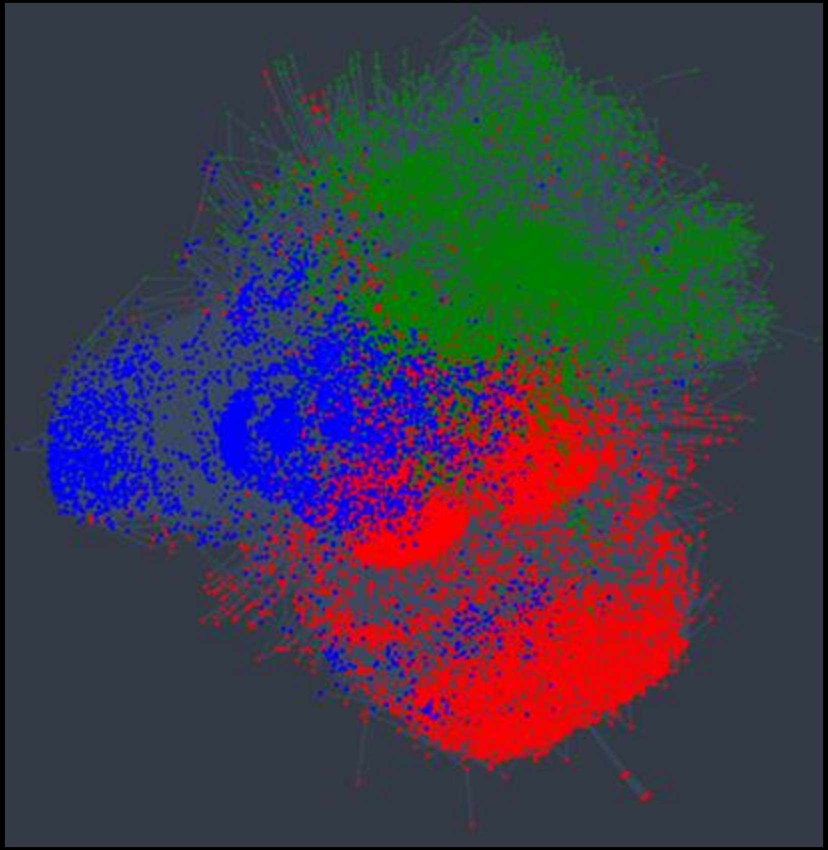


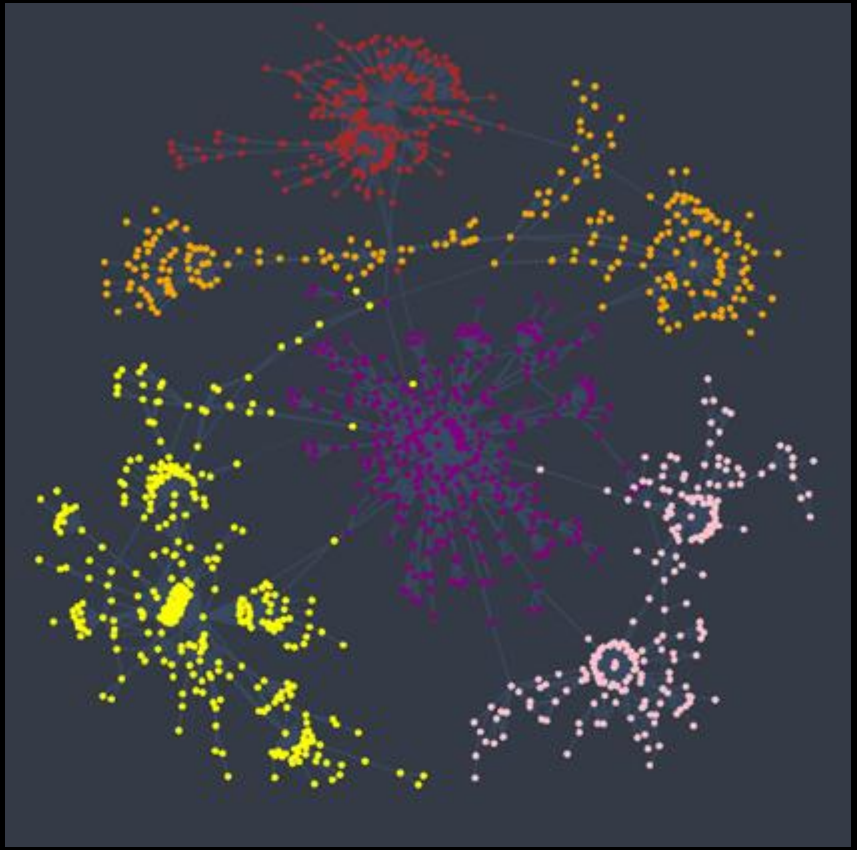
Performance Comparison

Algorithm	Modularity Score	Number of Communities
Bipartition	0.31	2
Louvain	0.5	26
Greedy Modularity	0.469	384
Label Propagation	0.137	678
Girvan Newman	-	-

- Louvain is the best-performing algorithm due to its highest modularity score and optimal number of communities.
- Girvan Newman algorithm was computationally very expensive.

Visualization of Louvain Communities





Cosmograph

Trend Analysis

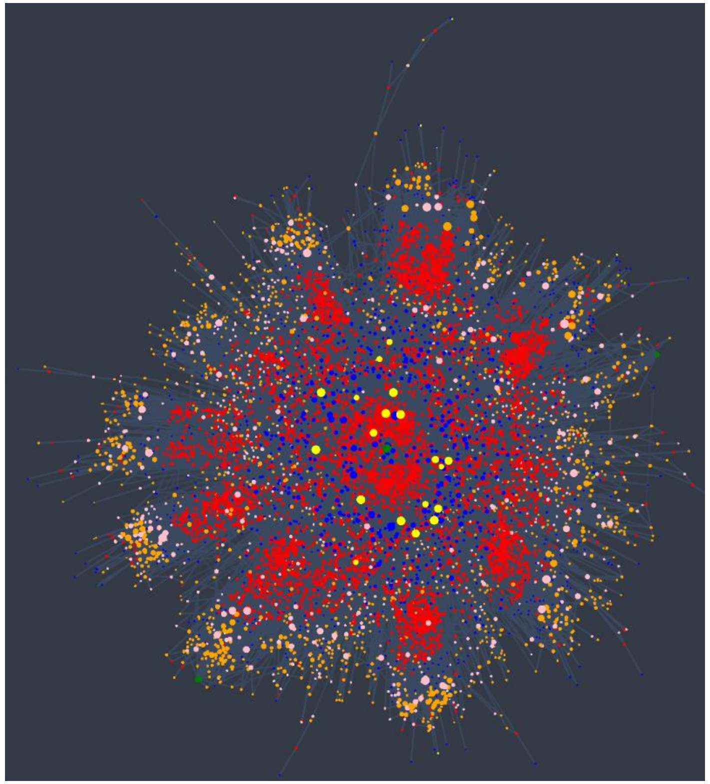
Community Wise

- Trending destinations and hotels for each community
- Below table shows an illustration of popular destinations and hotels for 5 communities:

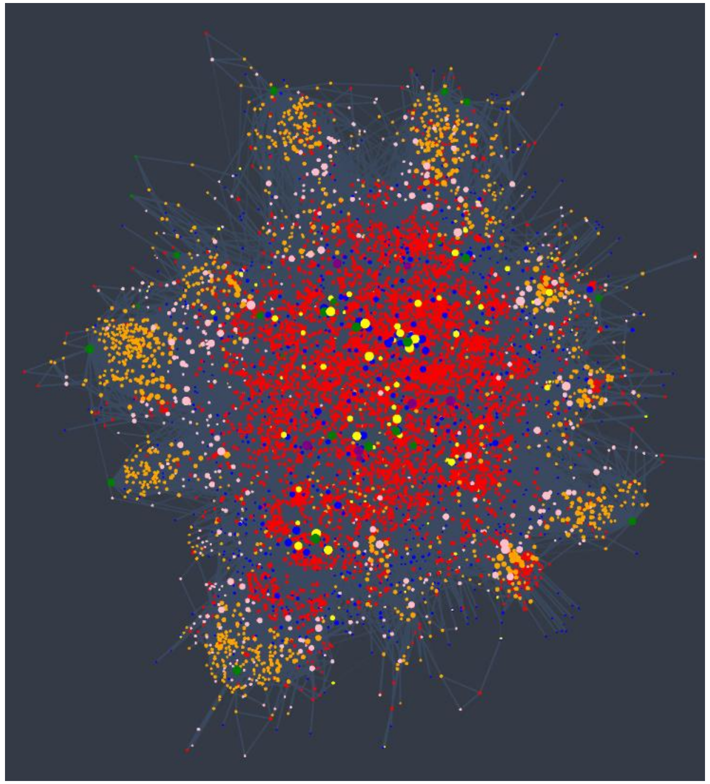
Community	Top_Destinations_id	Top_Hotels_id
0	[696, 146, 287, 20, 348]	[690, 3379, 21, 4254, 1108]
1	[218, 497, 612, 598, 586]	[31, 3350, 2945, 4114, 3642]
2	[192, 386, 545, 548, 151]	[1063, 577, 365, 981, 1837]
3	[342, 1620, 1044, 163, 2688]	[440, 3026, 2483, 5051, 3313]
4	[277, 752, 22, 519, 1219]	[2889, 1698, 739, 2914, 2822]

Popular Destinations and Hotels for various communities

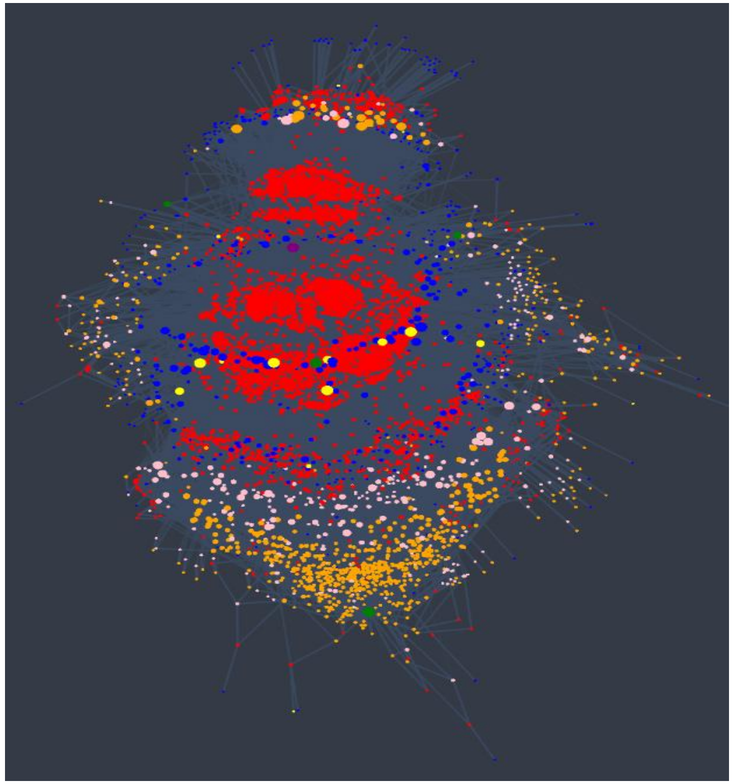
Top Communities Trend Visualizations



Community 1



Community 2



Community 3

- The size of a node in the above visualizations is proportional to its popularity and degree.

Seasonal Trends

- Trending hotels for every destination and season combination
- Below table shows an illustration of popular hotels for a particular destination based on the season :

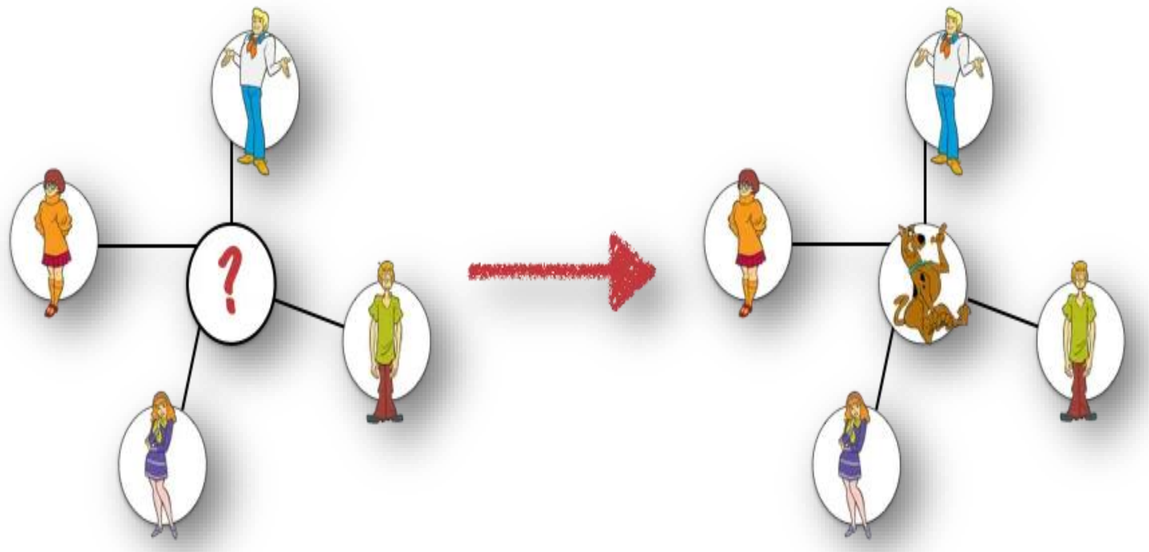
srch_destination_id	quarter	hotel_id	
200	1	[4746, 5300]	Spring
200	2	[6911, 11278]	Summer
200	3	[235, 2346, 2781, 3747]	Autumn
200	4	[6911, 6911, 2781, 8248]	Winter

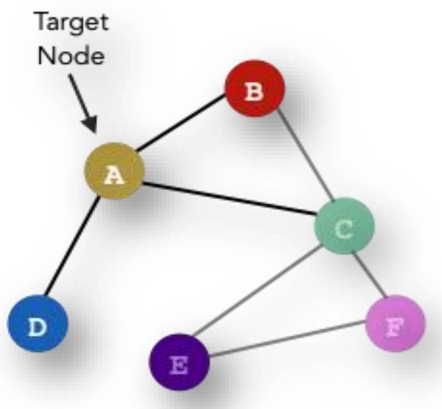
Popular Hotels for Destination ID 200

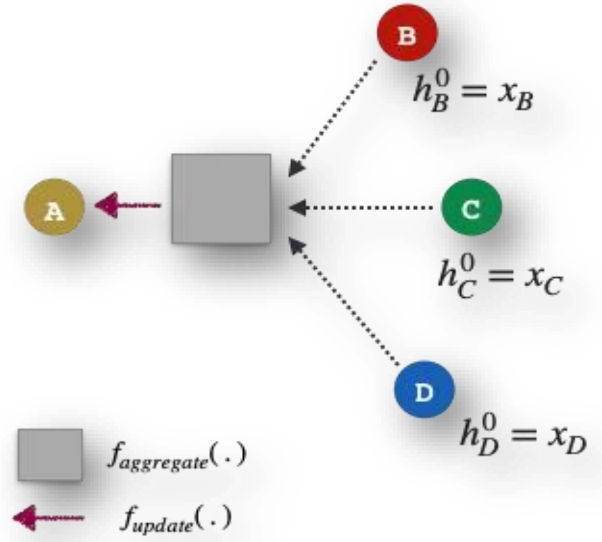
Recommendation Algorithms

Why use GraphSage Embeddings?

GraphSAGE is an **inductive** framework that leverages node attribute information to efficiently generate representations on previously unseen data.



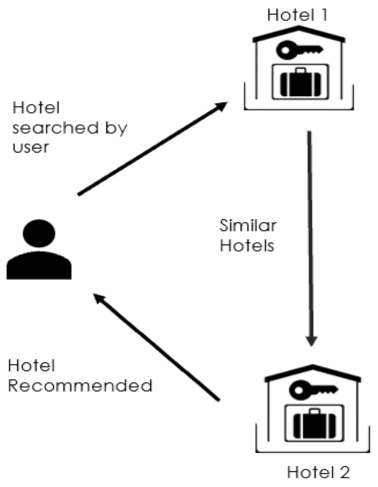




Content Based Filtering

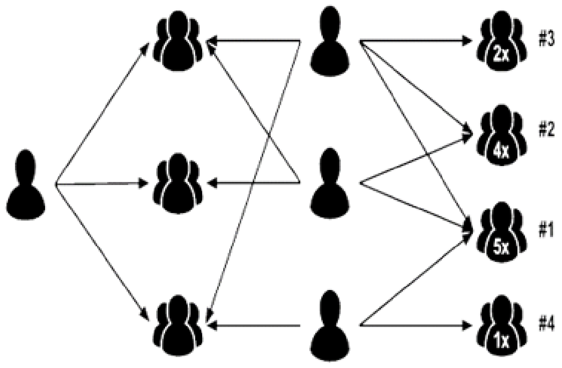
- The content-based approach utilizes user and/or item information to recommend similar items based on user preferences and previous actions

- Content-based methods aim to construct a model that explains user-item interactions using available features



Collaborative Filtering

- Collaborative filtering offers personalized re commendations to users by leveraging their similarities with other users or items.
- It relies on the assumption that users who have similar preferences in the past will have similar preferences in the future.

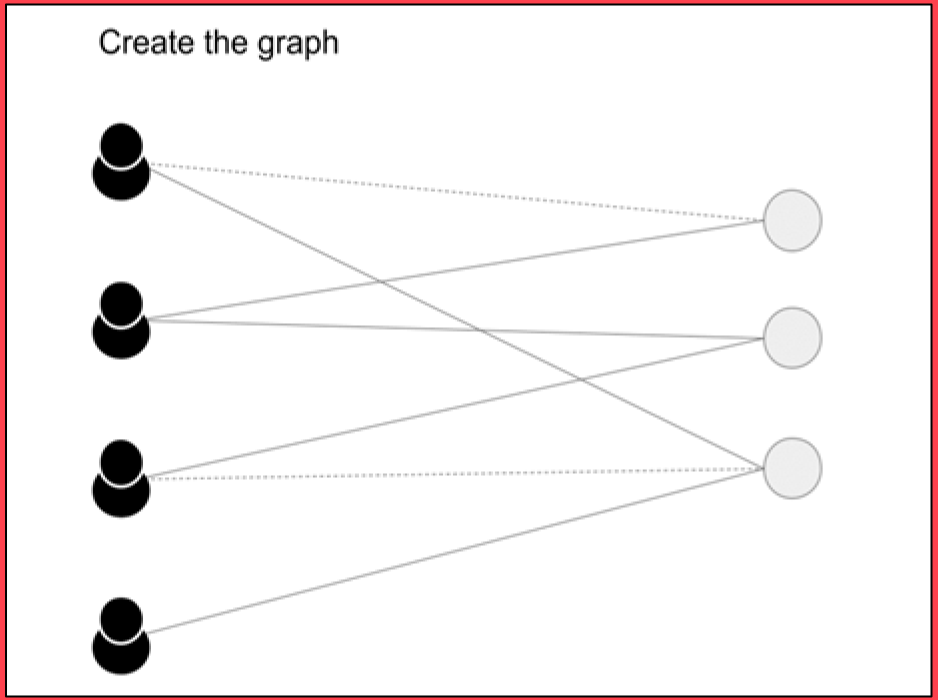


Graph Neural Networks

- GNNs as a class of deep learning models designed to handle graph-structured data.
- GNNs operate on nodes and edges of a graph, capturing information from neighboring nodes to learn representations.

How do GNN capture information from neighboring nodes and relations?

- Neural message passing enables communication among nodes in GNN-based recommender systems.
- It involves passing messages along edges and updating node embeddings based on aggregated information.



Error Metrics

The model utilizes user and destination embeddings to recommend hotels. By calculating the probabilities of hotel bookings, the model identifies hotels with the highest probabilities for personalized recommendations.

Recall and Precision as error metrics allows for the evaluation of recommendation system performance.

- **Recall** is the fraction of hotels that users have interacted with that can be found back in the recommended items.
- **Precision** is the fraction of recommended hotels that our user has indeed interacted with.

For Destination Id 678 and user 621166, different algorithms perform as follows:

Algorithm	**GNN**	**Collaborative Filtering**	**Content Based**
Recall	75%	50%	62.5%

DEMO

publicis
sapient

About the Team

Ravi Ranjan
Principal Data Scientist

Apurv Gude
Senior Data Scientist

THANK YOU

publicis
sapient

Feel free to connect over LinkedIn

Ravi Ranjan
@ravi-ranjan-03