Recommendation Systems

LVC 1: Recommendation Systems Part 1

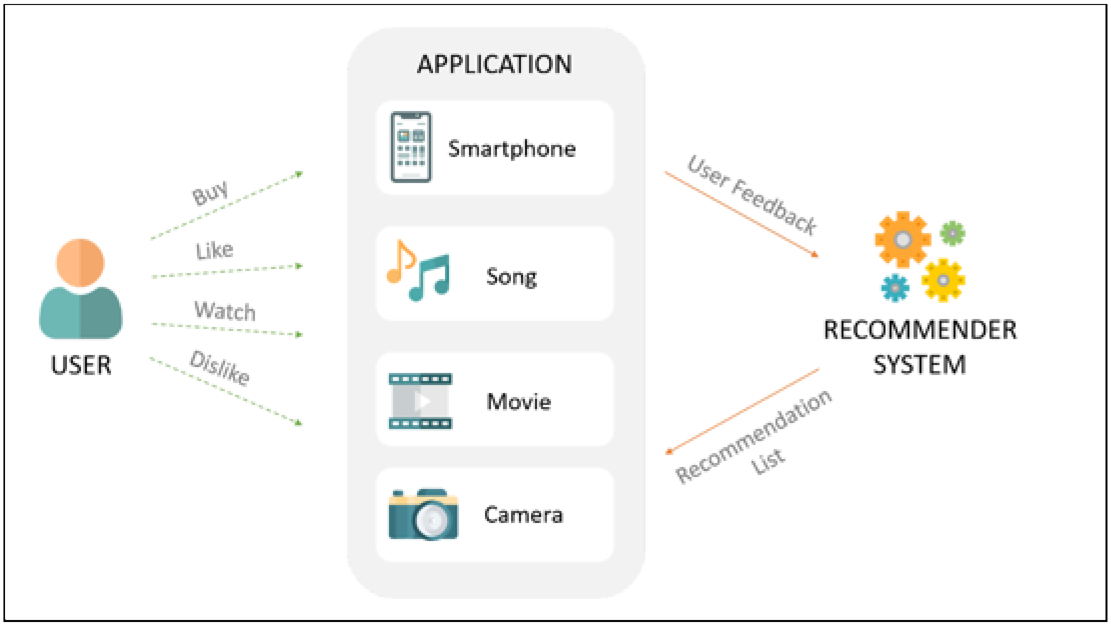
Recommendation systems: The term is self-explanatory, i.e., a system or an automated system that provides suggestions/recommendations.

For example, when you watch a movie of some particular genre on Netflix or watch some video on YouTube, they start to give you suggestions/recommendations for similar content types.

Other examples could be, when you do shopping on Amazon, after viewing some product, it starts suggesting a similar product. Or after listening to music on some app, like Spotify or Apple Music, it starts to suggest other songs you may also like to listen to.

There are several such examples where recommendation systems are being used today, but if you notice, they all carry a similar type of trait, i.e., using customer data to provide recommendations as shown in the below image.

Therefore, we can say that recommendation systems use customer data (what you watch, what you buy, what music you listen to, etc.) and find patterns in that data and based on these patterns provide personalized suggestions to improve customer experience.



Why do we need Recommendation Systems? Why not just search?

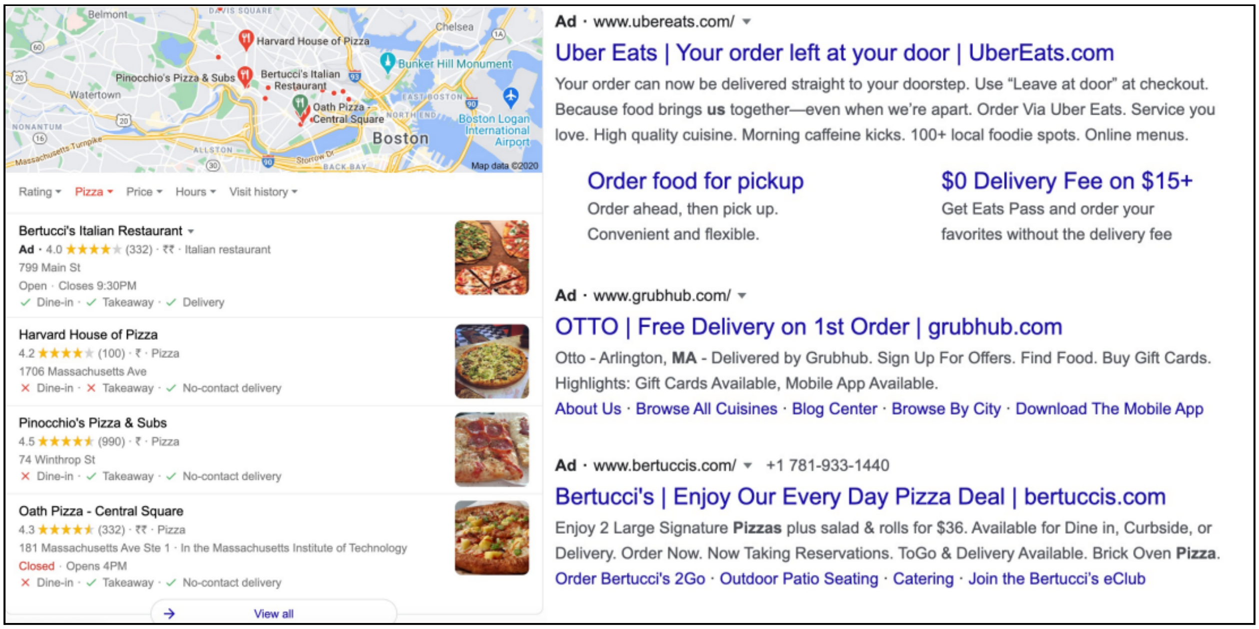
Let us understand this with an example. Approximately, every hour on YouTube, 30,000 hours worth of video is uploaded. That means, if we want to watch everything on YouTube, we have to live our lives 3000 times!! We have too much information to consume and too many options for us to choose from and most of them are not relevant or known to us. Only a very small fraction of this content is potentially relevant to us. So **how do we decide** which content is relevant to us?

Recommendation systems help us!! They constitute a considerable part of our **everyday lives** and below are a few examples where recommendation systems help us make a decision.

- What food should you eat today?
- Which activities should you plan for the upcoming weekend?
- Where should you plan your holidays?
- Whom should you date (and marry)?
- What are the professional connections of your interest?
- What content on YouTube or music on Spotify will fancy you?
- Which advertisements should you be subjected to?

Why not just search?

For example, if we run a search query in Google asking for **Pizza near me**, it shows us a lot of options as seen in the below figure. Google also shows the relevant advertisements. If the restaurant is just within a walkable distance, then we may think of some coupons for a discount, or if the restaurant is far away from our place, then we need options to select the delivery partner to deliver to our address. Google search provides a lot of options.



Bertucci's Italian Restaurant ▾
Ad · 4.0 ★★★★★ (332) · ₹₹ · Italian restaurant
799 Main St
Open · Closes 9:30PM
✓ Dine-in · ✓ Takeaway · ✓ Delivery

Harvard House of Pizza
4.2 ★★★★★ (100) · ₹ · Pizza
1706 Massachusetts Ave
✗ Dine-in · ✗ Takeaway · ✓ No-contact delivery

Pinocchio's Pizza & Subs
4.5 ★★★★★ (990) · ₹ · Pizza
74 Winthrop St
✗ Dine-in · ✓ Takeaway · ✓ No-contact delivery

Oath Pizza - Central Square
4.3 ★★★★★ (332) · ₹₹ · Pizza
181 Massachusetts Ave Ste 1 · In the Massachusetts Institute of Technology
Closed · Opens 4PM
✗ Dine-in · ✓ Takeaway · ✓ No-contact delivery

Ad · www.ubereats.com/ ▾

Uber Eats | Your order left at your door | UberEats.com

Your order can now be delivered straight to your doorstep. Use "Leave at door" at checkout. Because food brings **us** together—even when we're apart. Order Via Uber Eats. Service you love. High quality cuisine. Morning caffeine kicks. 100+ local foodie spots. Online menus.

Order food for pickup
Order ahead, then pick up. Convenient and flexible.

$0 Delivery Fee on $15+
Get Eats Pass and order your favorites without the delivery fee

Ad · www.grubhub.com/ ▾

OTTO | Free Delivery on 1st Order | grubhub.com

Otto - Arlington, **MA** - Delivered by Grubhub. Sign Up For Offers. Find Food. Buy Gift Cards. Highlights: Gift Cards Available, Mobile App Available.

About Us · Browse All Cuisines · Blog Center · Browse By City · Download The Mobile App

Ad · www.bertuccis.com/ ▾ +1 781-933-1440

Bertucci's | Enjoy Our Every Day Pizza Deal | bertuccis.com

Enjoy 2 Large Signature **Pizzas** plus salad & rolls for $36. Available for Dine in, Curbside, or Delivery. Order Now. Now Taking Reservations. ToGo & Delivery Available. Brick Oven **Pizza**.

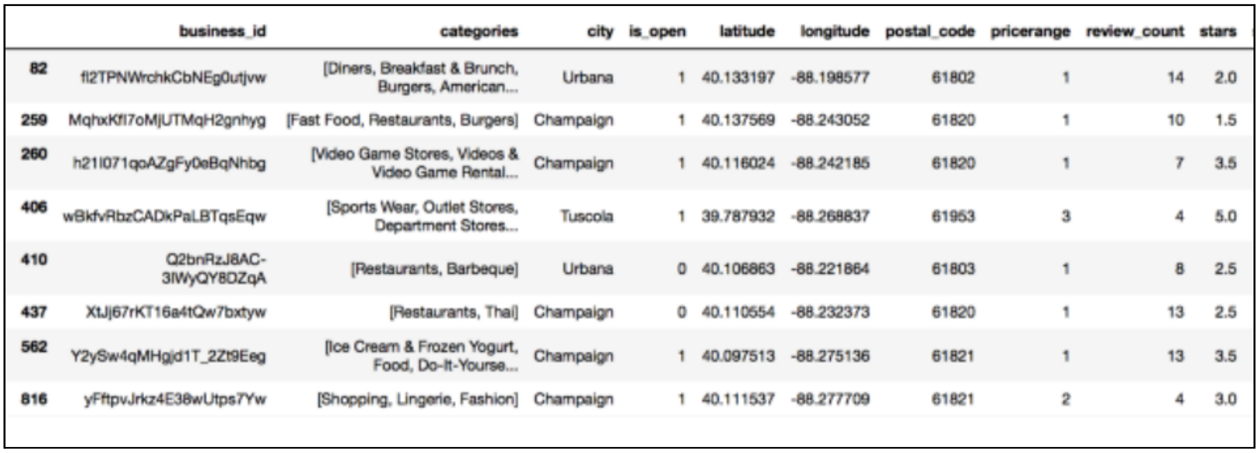
Order Bertucci's 2Go · Outdoor Patio Seating · Catering · Join the Bertucci's eClub

But, **which of these many options should be recommended to you? Which of these many should be advertised?**

To answer these questions, we need a lot of data to analyze. Let us now understand the data representation for recommendation systems with the help of the **Yelp dataset.**

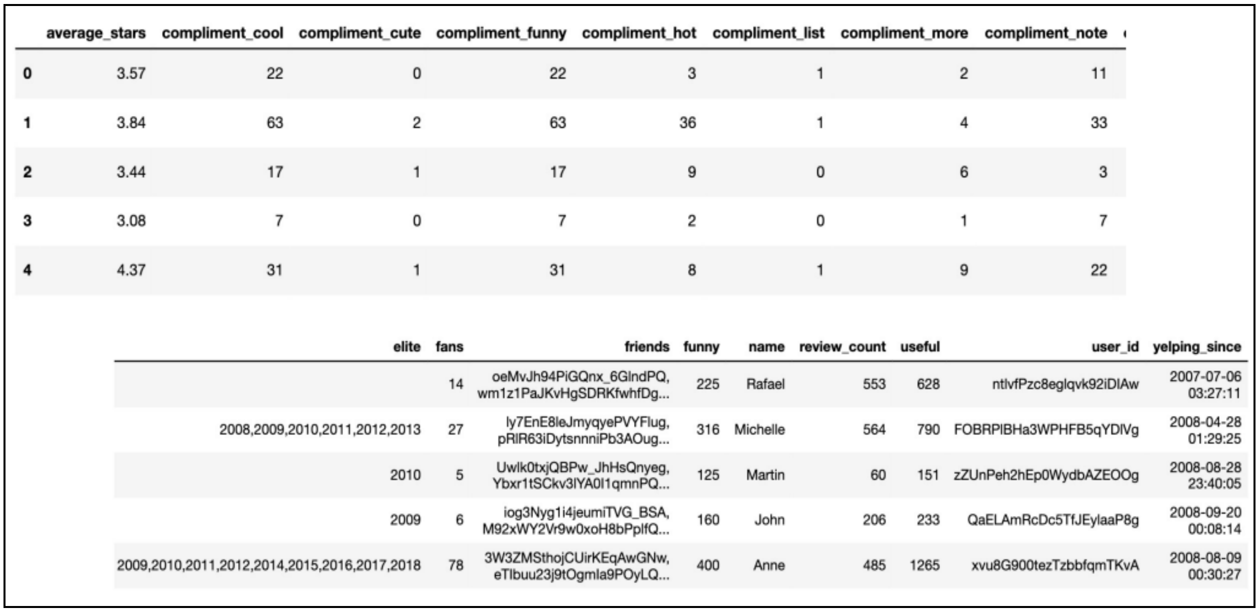
Yelp is a crowd-sourced review platform that collects reviews from multiple users and publishes them about local businesses. It contains five different types of datasets. They are as follows:

- **Business:** Contains business data including location data, attributes, and categories.



	business_id	categories	city	is_open	latitude	longitude	postal_code	pricerange	review_count	stars
82	fI2TPNWrchkCbNEg0utjvw	[Diners, Breakfast & Brunch, Burgers, American...	Urbana	1	40.133197	-88.198577	61802	1	14	2.0
259	MqhxKfI7oMjUTMqH2gnhyg	[Fast Food, Restaurants, Burgers]	Champaign	1	40.137569	-88.243052	61820	1	10	1.5
260	h21I071qoAZgFy0eBqNhbg	[Video Game Stores, Videos & Video Game Rental...	Champaign	1	40.116024	-88.242185	61820	1	7	3.5
406	wBkfvRbzCADkPaLBTqsEqw	[Sports Wear, Outlet Stores, Department Stores...	Tuscola	1	39.787932	-88.268837	61953	3	4	5.0
410	Q2bnRzJ8AC-3IWyQY8DZqA	[Restaurants, Barbeque]	Urbana	0	40.106863	-88.221864	61803	1	8	2.5
437	XtJj67rKT16a4tQw7bxtyw	[Restaurants, Thai]	Champaign	0	40.110554	-88.232373	61820	1	13	2.5
562	Y2ySw4qMHgjd1T_2Zt9Eeg	[Ice Cream & Frozen Yogurt, Food, Do-It-Yourse...	Champaign	1	40.097513	-88.275136	61821	1	13	3.5
816	yFftpvJrkz4E38wUtps7Yw	[Shopping, Lingerie, Fashion]	Champaign	1	40.111537	-88.277709	61821	2	4	3.0

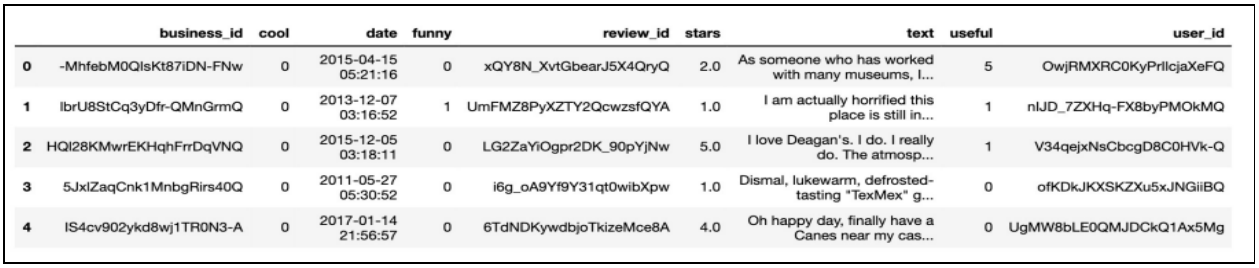
- **User:** User data including the user's friend mapping and all the metadata associated with the user.



	average_stars	compliment_cool	compliment_cute	compliment_funny	compliment_hot	compliment_list	compliment_more	compliment_note
0	3.57	22	0	22	3	1	2	11
1	3.84	63	2	63	36	1	4	33
2	3.44	17	1	17	9	0	6	3
3	3.08	7	0	7	2	0	1	7
4	4.37	31	1	31	8	1	9	22

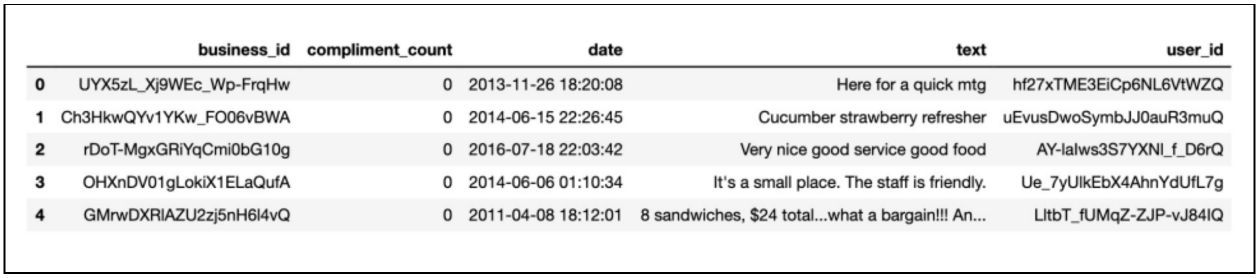
elite	fans	friends	funny	name	review_count	useful	user_id	yelping_since
	14	oeMvJh94PiGQnx_6GlndPQ, wm1z1PaJKvHgSDRKfwhfDg...	225	Rafael	553	628	ntlvfPzc8eglqvk92iDIAw	2007-07-06 03:27:11
2008,2009,2010,2011,2012,2013	27	ly7EnE8leJmyqyePVYFlug, pRIR63iDytsnnniPb3AOug...	316	Michelle	564	790	FOBRPlBHa3WPHFB5qYDlVg	2008-04-28 01:29:25
2010	5	Uwlk0txjQBPw_JhHsQnyeg, Ybxr1tSCkv3lYA0l1qmnPQ...	125	Martin	60	151	zZUnPeh2hEp0WydbAZEOOg	2008-08-28 23:40:05
2009	6	iog3Nyg1i4jeumiTVG_BSA, M92xWY2Vr9w0xoH8bPplfQ...	160	John	206	233	QaELAmRcDc5TfJEylaaP8g	2008-09-20 00:08:14
2009,2010,2011,2012,2014,2015,2016,2017,2018	78	3W3ZMSthojCUirKEqAwGNw, eTlbuu23j9tOgmla9POyLQ...	400	Anne	485	1265	xvu8G900tezTzbbfqmTKvA	2008-08-09 00:30:27

- **Review:** Contains full review text data including the user_id, who wrote the review, and the business_id the review is written for.



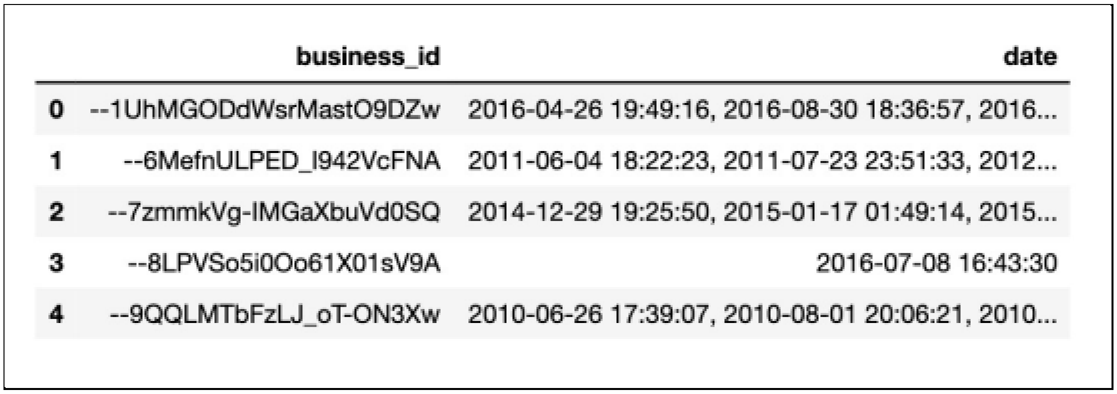
	business_id	cool	date	funny	review_id	stars	text	useful	user_id
0	-MhfebM0QIsKt87iDN-FNw	0	2015-04-15 05:21:16	0	xQY8N_XvtGbearJ5X4QryQ	2.0	As someone who has worked with many museums, I...	5	OwjRMXRC0KyPrIlcjaXeFQ
1	IbrU8StCq3yDfr-QMnGrmQ	0	2013-12-07 03:16:52	1	UmFMZ8PyXZTY2QcwzsfQYA	1.0	I am actually horrified this place is still in...	1	nIJD_7ZXHq-FX8byPMOkMQ
2	HQl28KMwrEKHqhFrrDqVNQ	0	2015-12-05 03:18:11	0	LG2ZaYiOgpr2DK_90pYjNw	5.0	I love Deagan's. I do. I really do. The atmosp...	1	V34qejxNsCbcgD8C0HVk-Q
3	5JxlZaqCnk1MnbgRirs40Q	0	2011-05-27 05:30:52	0	i6g_oA9Yf9Y31qt0wibXpw	1.0	Dismal, lukewarm, defrosted-tasting "TexMex" g...	0	ofKDkJKXSKZXu5xJNGiiBQ
4	lS4cv902ykd8wj1TR0N3-A	0	2017-01-14 21:56:57	0	6TdNDKywdbjoTkizeMce8A	4.0	Oh happy day, finally have a Canes near my cas...	0	UgMW8bLE0QMJDCkQ1Ax5Mg

- **Tips are written by a user of a business**: Tips are shorter than reviews and tend to convey quick suggestions.



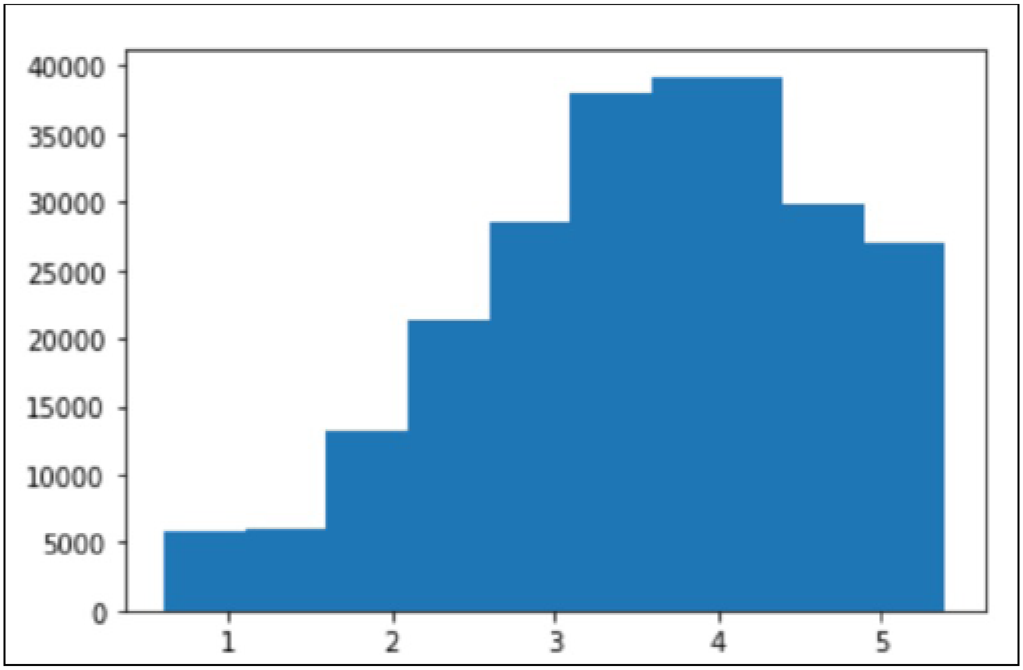
	business_id	compliment_count	date	text	user_id
0	UYX5zL_Xj9WEc_Wp-FrqHw	0	2013-11-26 18:20:08	Here for a quick mtg	hf27xTME3EiCp6NL6VtWZQ
1	Ch3HkwQYv1YKw_FO06vBWA	0	2014-06-15 22:26:45	Cucumber strawberry refresher	uEvusDwoSymbJJ0auR3muQ
2	rDoT-MgxGRiYqCmi0bG10g	0	2016-07-18 22:03:42	Very nice good service good food	AY-laIws3S7YXNl_f_D6rQ
3	OHXnDV01gLokiX1ELaQufA	0	2014-06-06 01:10:34	It's a small place. The staff is friendly.	Ue_7yUlkEbX4AhnYdUfL7g
4	GMrwDXRlAZU2zj5nH6l4vQ	0	2011-04-08 18:12:01	8 sandwiches, $24 total...what a bargain!!! An...	LltbT_fUMqZ-ZJP-vJ84IQ

- **Check-ins:** List of timestamps for each check-in on a business.



	business_id	date
0	--1UhMGODdWsrMastO9DZw	2016-04-26 19:49:16, 2016-08-30 18:36:57, 2016...
1	--6MefnULPED_I942VcFNA	2011-06-04 18:22:23, 2011-07-23 23:51:33, 2012...
2	--7zmmkVg-IMGaXbuVd0SQ	2014-12-29 19:25:50, 2015-01-17 01:49:14, 2015...
3	--8LPVSo5i0Oo61X01sV9A	2016-07-08 16:43:30
4	--9QQLMTbFzLJ_oT-ON3Xw	2010-06-26 17:39:07, 2010-08-01 20:06:21, 2010...

Let us see sample statistics from the available data. The below image shows the distribution of star ratings across restaurants and people. The X-axis denotes that the ratings range from 1 to 5, and the Y-axis denotes the count. The information shows that we have a peak between ratings 3 and 4. A fewer number of people also gave ratings 1 and 2. This is the basic information that we can extract from the known reviews.



But what fraction of reviews are known?

Yelp is a rich dataset. It has approximately 2M users and 200k businesses. However, we have 0.4T total possible interactions, assuming that each user gives a review for a business only one time.

Users = ~2M

Total possible reviews = ~2M x ~200k = ~0.4T

Businesses = ~ 200k

Known reviews = ~ 8M

Fraction known = ~ 8M / 0.4 T = 2 x 10^{-5}

The known reviews are only 8 Million out of 0.4 Trillion. That is, only 2 out of every 100K reviews are known and the rest are unknown. Finding these unknown reviews is the **goal of the recommendation system.** Now, we are ready to formalize the recommendation system problem statement.

Problem Statement:

The prediction problem of the recommendation system is estimating the likelihood of **matching the given user and the given provider, at a given time, and in a given context.**

That is, providing the list of search criteria to match as per user interest so that users select the item within a few clicks. This, in turn, saves the time of the user and provides the best product of their choice or preference which could lead to an increment in the business growth.

For example, LinkedIn would like to recommend the user according to the domain or area in which the user works. So if the person works as a data scientist, LinkedIn would like to provide follow recommendations for other people who work in the data science field.

LinkedIn: Connect people professionally

Facebook: Filter friends' feed

Poshmark, Etsy: Organize the content of the display

Amazon, Retail: Filter products and sellers

Tinder, Match: Find a suitable partner

Netflix, YouTube, Spotify: Entertainment of interest

In the simplest version, the prediction problem statement can be given as:

N users

M providers or items (or other users)

Given user '*i*' and item '*j*'

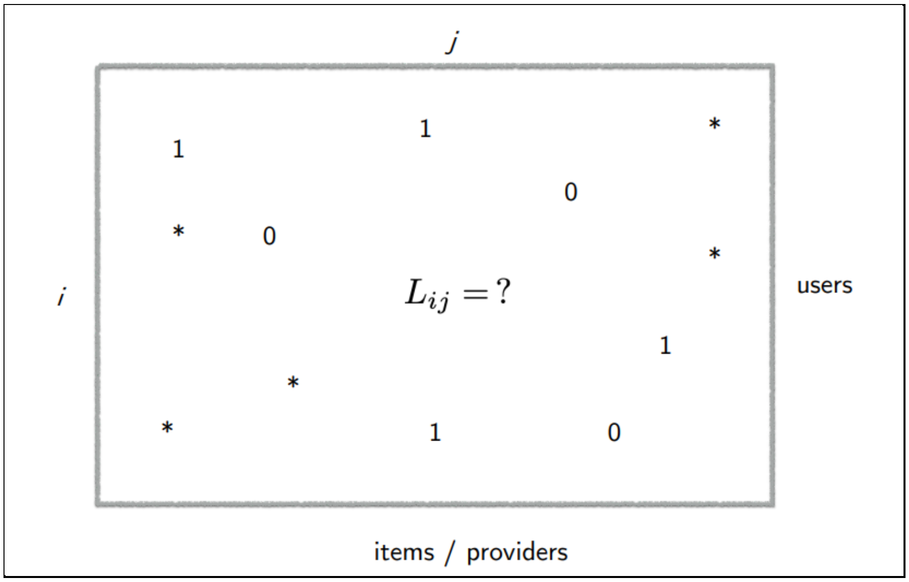
find the likelihood of '*i*' matching with '*j*'

$L_{ij} = ?$

L_{ij}- The likelihood of the user '*i*' matching with the item '*j*'. For example,

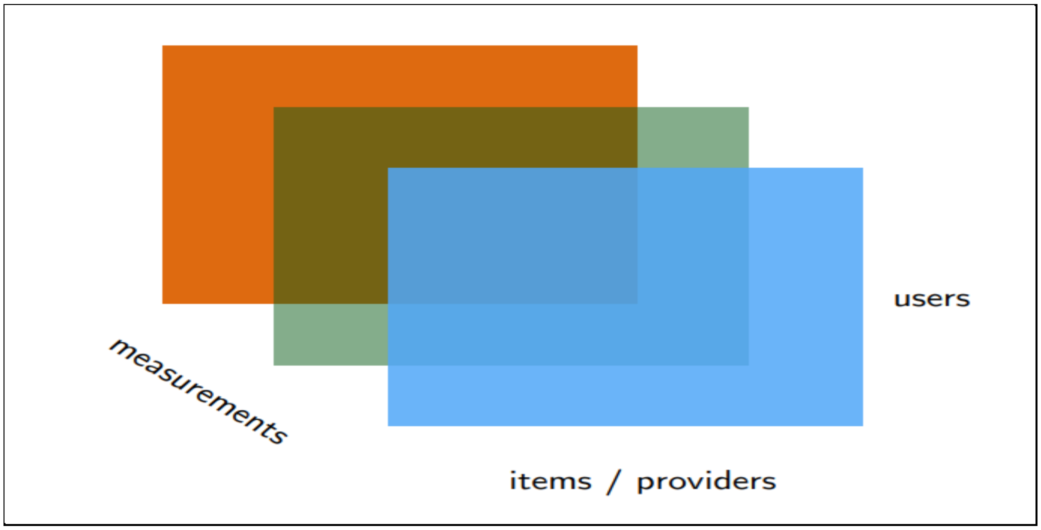
- If it is a movie, what is the likelihood that the user '*i*' would like to watch the movie '*j*'
- If it is a restaurant, what is the likelihood that the user '*i*' would like to eat in restaurant '*j*'

The same can be viewed in a matrix representation, where 'i' will be the user and 'j' will be the provider as shown in the below figure. The value 1 represents the match and 0 represents no match and * represents the unknowns. We should complete the matrix by finding the unknowns (*).



Tensor

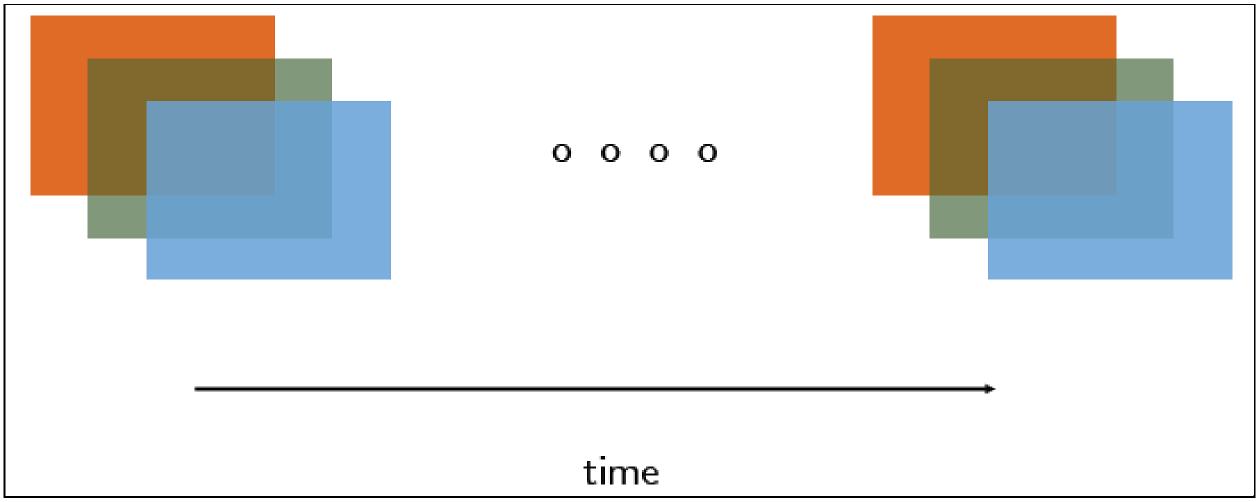
In the above example, we have understood the users and the items. But in real-time, we have more pieces of information, for example, in the Yelp dataset, we have measurements like the user, business, tips, check-ins, and reviews, and this information is related to each other. This multilinear relationship among many slices of data can be represented by **tensors**.



Time-Varying Tensor:

The information that is available is changing over time. For example, there is a famous chef at a specific restaurant who is the main reason for the high rating and after two years, we cannot always expect that that chef still works for the same restaurant.

A time-varying tensor is a very high-dimensional complex object that changes over time with very partial information. We can use the information which is available at present and at a particular timestamp we can forecast what's going to come next.



Finally, our prediction problem statement is that we should predict all the unknown values in this **highly complex time-varying tensor**, which is extremely noisy and some of the information is unstructured.

Before moving on to finding the solutions for this problem statement, let us discuss some of the challenges of recommendation systems.

Challenges in Recommendation Systems:

1. **User-provided data**
    - Not generated at random: Reviews should be entered by individual users, they should not be generated randomly.
    - Can be strategic: Sometimes, users may be forced in some situations to provide unrealistic reviews to get certain benefits from the provider.

- Can be driven by innate preference: A user watches a movie of his favorite actor and rates it nicely irrespective of whether the movie is good or bad (or) a user watches a movie of an actor which he doesn’t like and rates it badly.

2. **Provider/Item**
   - Can be systematically manipulated
     - Convenient location
     - Strategically modified content

Example: Assume, in a specific area we have a lot of apartments where we need services for housekeeping. An agency that is too far from the area gets manipulated and shown very near to the apartment so that people can select that agency for services.

3. **Side effects of recommendation systems**
   - Information bubble
   - Feedback loops
   - Too powerful platforms

Example: A user is planning to buy a car. He goes to a car dealership and looks for a specific color but the waiting period for that color is more than a month which the user doesn't want to do. He wants the car immediately and the color available to pick is white.

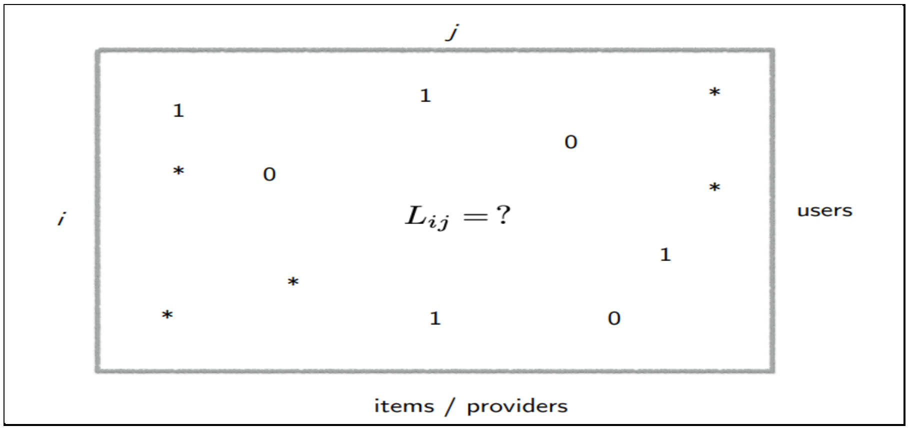
Likewise, most of the users who don't want the waiting period buy a white car. Thus, dealers are selling more white cars, hence manufacturers manufacture more white cars assuming that the white car is the one that most of the buyers love. This causes feedback loops.

We need to prevent this kind of situation. Recommendation systems should not just exploit what works well, they also need to provide enough exploration. And this is a problem of decision-making systems that utilizes the prediction coming out of recommendation systems.

4. **Regulation of platforms**
   - Interplay with recommendation systems

To avoid monopolies and misinformation on social media platforms, there should be regulations to control them.

Now, let us consider the **simple problem statement: complete the matrix,** and we will discuss the simple solutions available for that.



Like every machine learning algorithm, a recommender system makes predictions based on users' historical behaviors. Specifically, it's to predict user preference for a set of items based on past experience. To achieve this task, there exist many methods. Here, we will discuss the two basic methods, averaging and content-based filtering.

Averaging

One simple way of doing this is, we assume that every user is the same (i.e., all rows of the matrix are the same) and we predict values by the average of the column, or in our case, prediction is the average rating given to some restaurant.

- A simple assumption to get started
    - All users are identical
    - That is, all rows of the matrix are the same
- We wish to predict L_{ij}
    - All rows in the column ‘j’ are the same
    - Each observation in column ‘j’ is an outcome of a coin toss with bias L_{ij}
- Solution
    - Estimate L_{ij} as the average of all observations in column j

How accurate?

For example, a movie that has a rating of 4 out of 5 with 5 users rated is less accurate than a movie that has a rating of 3 out of 5 with 500 users ratings.

By the law of large numbers, if the number of observations (user ratings) grows, the estimation is more accurate. So, a few good ratings do not mean the provider is excellent. How to correct this?

The Central Limit Theorem gives the estimation error scale as $1/\sqrt{n}$ with 'n' observations. Now the improved estimate will be average + $1/\sqrt{n}$.

- By the law of large numbers
  - As the number of observations in column j grows
  - The estimate converges to the true likelihood
- But how large should it be?
  - One (or a few) good rating(s) does not mean the place is excellent
- By the Central Limit Theorem
  - The estimation error scale is $1/\sqrt{n}$ with n observations
- An improved estimate: average + $1/\sqrt{n}$

If we assume all users are identical, then the prediction will be the column average and we correct it by adding $1/\sqrt{n}$. Similarly, if we assume all items are identical, then the prediction will be the row average and we correct it by adding $1/\sqrt{n}$.

Now, put these simple estimators together, then the final prediction will be:

$$2L_{ij} = L_i + 1/\sqrt{n_i} + L_j + 1/\sqrt{n_j}$$

L_i is the average of observed entries in row i

n_i is the number of observed entries in row i

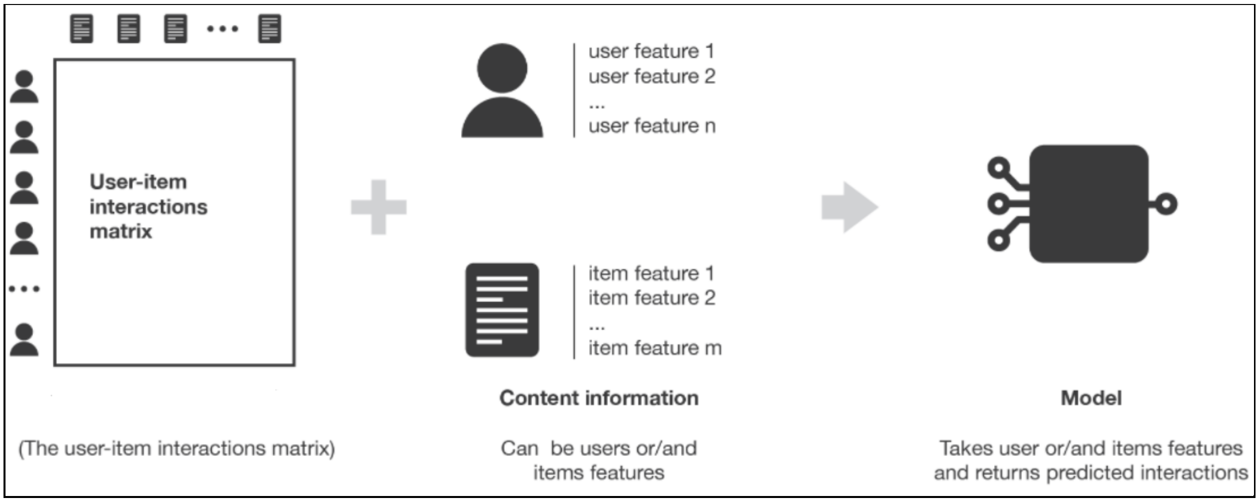
L_j is the average of observed entries in column j

n_j is the number of observed entries in column j

Content-based filtering

Content-based approaches use additional information about users and/or items. If we consider the example of a movie recommender system, this additional information can be, for example, the age, the sex, the job, or any other personal information for users as well as the category, the main actors, the duration, or other characteristics for the movies (items).

Then, the idea of content-based methods is to try to build a model, based on the available "features", that explain the observed user-item interactions. Still considering users and movies, we will try, for example, to model the fact that young women tend to rate better some movies, and young men tend to rate better some other movies, and so on.



User-item interactions matrix
user feature 1
user feature 2
...
user feature n
item feature 1
item feature 2
...
item feature m
Content information
Model
(The user-item interactions matrix)
Can be users or/and items features
Takes user or/and items features and returns predicted interactions

If we manage to get such a model, then, making new predictions for a user is pretty easy: we just need to look at the profile (age, sex, …) of this user and, based on this information determine relevant movies to suggest.

- A little more involved assumption
    - Users and items have features
        - That are observed and can predict the likelihood
- Let features of user 'i' be x_i
- Let features of item 'j' be y_i
- Then, the goal is to learn f where $L_{ij} = f(x_i, y_i)$

This is a supervised learning problem

- Labeled data:
    - Each observed entry in the matrix (i, j) corresponds to labeled data $((x_i, y_i); L_{ij})$
- Learning problem:
    - Learn the model/function that maps features to label

Appendix

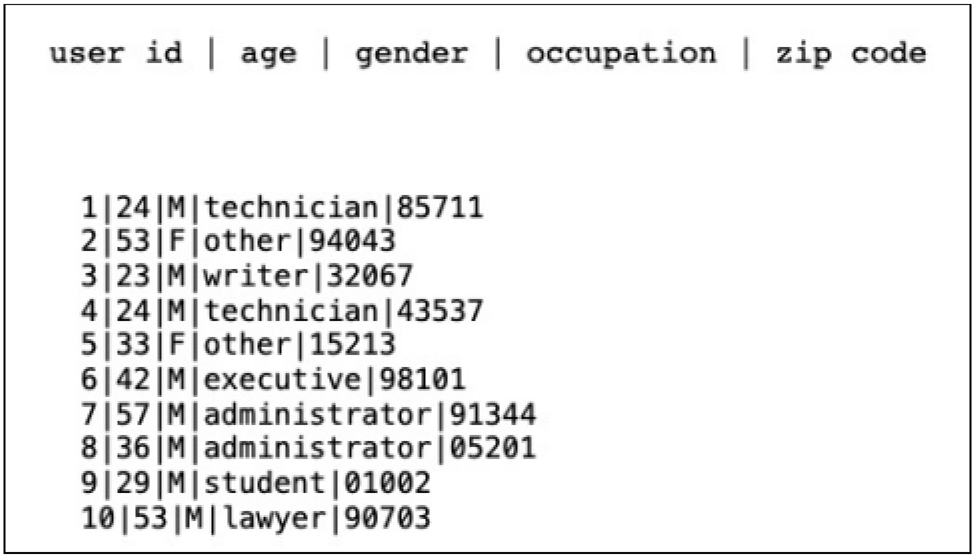
Converting Content to features:

Let us revisit the user data. How do we convert these “attributes” or “content” to features?

Age: It is a number, That’s easy

Gender: Two classes or binary. We can convert into 0 / 1

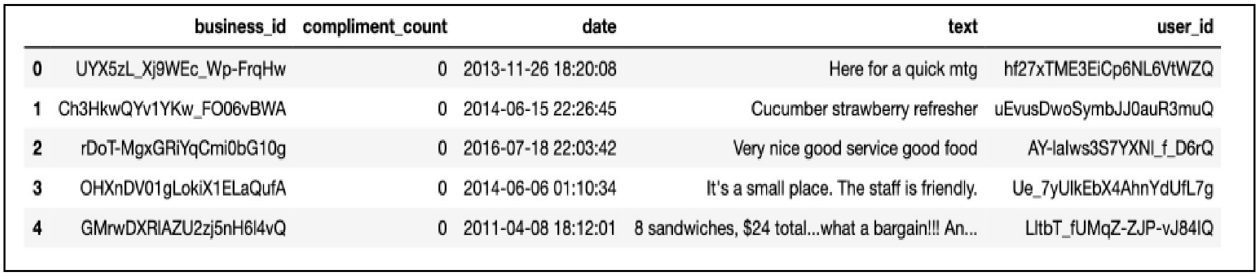
Occupation: Categorical variables can be converted into numeric values using one-hot encoding.



```
user id | age | gender | occupation | zip code


  1|24|M|technician|85711
  2|53|F|other|94043
  3|23|M|writer|32067
  4|24|M|technician|43537
  5|33|F|other|15213
  6|42|M|executive|98101
  7|57|M|administrator|91344
  8|36|M|administrator|05201
  9|29|M|student|01002
  10|53|M|lawyer|90703
```

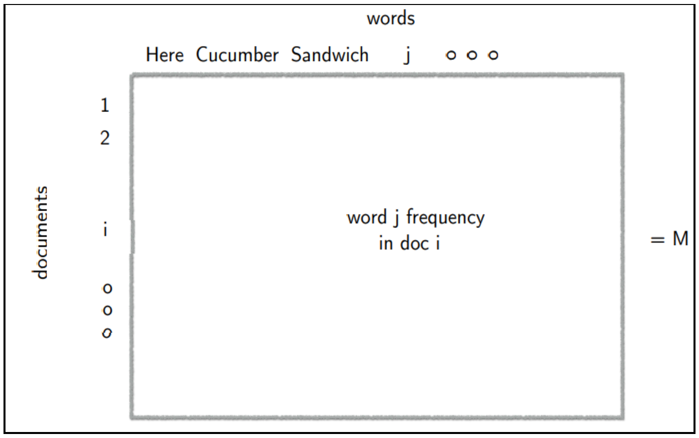
But what about “Tip” data? The column ‘text’ has free-form text.



	business_id	compliment_count	date	text	user_id
0	UYX5zL_Xj9WEc_Wp-FrqHw	0	2013-11-26 18:20:08	Here for a quick mtg	hf27xTME3EiCp6NL6VtWZQ
1	Ch3HkwQYv1YKw_FO06vBWA	0	2014-06-15 22:26:45	Cucumber strawberry refresher	uEvusDwoSymbJJ0auR3muQ
2	rDoT-MgxGRiYqCmi0bG10g	0	2016-07-18 22:03:42	Very nice good service good food	AY-laIws3S7YXNl_f_D6rQ
3	OHXnDV01gLokiX1ELaQufA	0	2014-06-06 01:10:34	It's a small place. The staff is friendly.	Ue_7yUlkEbX4AhnYdUfL7g
4	GMrwDXRlAZU2zj5nH6l4vQ	0	2011-04-08 18:12:01	8 sandwiches, $24 total...what a bargain!!! An...	LltbT_fUMqZ-ZJP-vJ84lQ

We should convert these texts to a vector of numbers:

- Create word frequency in documents matrix M



- Create word frequency in documents matrix M
- Perform Principal Component Analysis of M
    - Each document receives k co-ordinates
        - via k principal components
- This is the vector representing the text features