Recommender-Systems-Lesson-1

(0:31 - 1:05)

Well, let's begin. Okay, so a bit of background. Yeah, on WeBip, there's a shared folder with the teaching materials, including the introduction of the course and the slide here.

(1:12 - 1:29)

It's on OneDrive. There's a link to a OneDrive folder. Okay, so my goal here is to give you a sense of the general scope of the things we will be discussing.

(1:30 - 2:09)

We will typically spend a bit of time on each of them as the semester progresses. So I will be repeating some of the things I'm going to say now as we delve into the details of each of the various pieces. But in a more general sense, what is a recommendation system? A recommendation system is a black box that provides, as well the name suggests, recommendations to users.

(2:10 - 2:27)

Nothing surprising here. A recommendation system can be anything that does this, really. So as we will see, a number of methods, techniques, and strategies can be deployed.

(2:28 - 2:52)

But in a general sense, they all do the same. They try to take the enormous amount of options that is at your disposal and filter it down into a limited set of recommendations to show to you. Now, what does it mean? Well, for example, you are booking and you provide hotel recommendations.

(2:53 - 3:11)

What does it mean? A user comes to you and selects a city, selects a date of arrival, a date of departure, and possibly a few options. Free cancellation, four stars, and a certain minimum rating. And you have a list.

(3:13 - 3:31)

Is every user going to see the same list? The answer is no. Why? Because different users have different characteristics. For example, I may like hotels with fancy breakfasts.

(3:31 - 3:44)

You may like hostels where there's a bar and you can have a chat with the other guests or other

people in the city. We may have different price ranges. We may be traveling for different reasons, and so on.

(3:45 - 4:07)

So for the purposes of the company, it's easier to try to, well, it's easier. It's better if the user finds a solution that is more pertinent to their needs immediately, rather than, you know, having to scroll around. Possibly, the hotels are also going to be ranked differently based on how much they pay the platform.

(4:08 - 4:38)

So there's a lot of criteria here. How many hotels could there possibly be in a city with rooms available within a certain period? Hundreds, maybe. But perhaps you're Amazon, a gigantic e-commerce platform that sells anything, mostly, to anyone at any place at any time.

(4:39 - 4:46)

Unless you're in Antarctica. Here, the scenario is quite different. You don't have 100 hotels.

(4:47 - 4:58)

You have 500 million possible products. Yes, the user will ask a query. So the user will open the webpage and write something.

(4:58 - 5:10)

I want a new charger for my phone. I want a new cover for my laptop. I want a new travel adapter because I'm going to Switzerland and I need to plug my devices.

(5:12 - 5:38)

Again, you will have to filter your results somehow and end up with an enormous list of possible results. How do you rank them? What is the first thing that you show? Is a predefined criteria or not? Again, a number of choices here. How do you use Spotify? Music recommendation.

(5:39 - 6:00)

Again, different scenario. In this case, I would imagine you're not going into the homepage and writing a specific request for the system. In this case, it is typically more common for the system to recommend something to you based on what you have done in the past.

(6:01 - 6:10)

Based on your playlists. Based on the songs you have listened to. And the algorithms that you need to do that are quite different.

(6:13 - 6:21)

And the list could go on and on and on and on. But the last name on this list is Netflix. Again, different scenario.

(6:21 - 6:39)

Movie recommendation. Again, you will have different options at your disposal. But if we take... I would guess there is going to be no two people in this classroom that by opening Netflix would see the same results.

(6:40 - 6:50)

Or would even see the same page structure. The very structure of the page is different. If you open it, you will see that there's a number of rows which are typically thematically consistent.

(6:50 - 6:58)

They're called carousels. Some of them are rather constant. For example, the new movies.

(6:58 - 7:08)

The ones that are going to be dropped from the platform. Like, last occasion to see this. And then you will see something that is more related to your interest.

(7:09 - 7:17)

I may see a sci-fi carousel. Someone else might see a sports team carousel. Or a comedy carousel on whatever.

(7:17 - 7:34)

The very structure of the page which contains other recommendations is personalized. Now, Netflix has been... How many of you know what was Netflix doing? Well, I'll tell you. It's easier.

(7:35 - 7:59)

They were selling... They were sending DVDs via mail before the streaming era. So you would get your red envelope with a DVD. Or, well, any color.

Depends on where you were. And they would send mailing lists with options. So the world has really changed a lot in the last 15 years.

(8:00 - 8:19)

In 2000, and I think it was 6, Netflix decided to launch the first major competition in the field. It was called the Netflix Prize. And the idea was that they would give you 100 million user interactions.

(8:20 - 8:58)

And the practitioners and researchers were asked to build a new recommendation model. If you managed to build a recommendation model able to improve the accuracy of the recommendation by 10% compared to their previous one, we will see how to measure the quality of recommendations later on, but the prize was $1 million. So that is the amount of money that a better recommendation, a 10% better recommendation, can move.

(8:59 - 9:09)

It was also extremely difficult. It took almost two years for a team to be able to do that. So there's a lot of money involved in this field.

(9:14 - 9:36)

Okay, but what data do we use? Well, if our goal is to provide recommendations for items, items can be any of the things I previously mentioned. Hotel, movies, restaurants, recipes, songs, books. Anything you want to suggest to someone.

(9:39 - 9:53)

The very last, well, the very minimum, you need to know which items exist. So you need to know something about the items themselves. If you are unlucky, you just know that they exist.

(9:53 - 10:03)

You just know that at some point in your catalog, there is item 127. You have no idea what it is, what it does, what's the domain. You know it exists.

(10:03 - 10:19)

You can do recommendations in this way, and we will see how. Or you have more details such as, I don't know, if it's a book, you might have some description of the book. You may have text, you may have features.

(10:20 - 10:52)

In which year was it published? Who wrote it? Is this the first version or not? How many copies it sold? If it's a movie, again, the genre, the director, the cast, and a number possibly of other features, and so on. Everything here is item data, so descriptors of the items. We will see how to use them, and why they are important, and also why they are not in a bit.

(10:54 - 11:25)

Of course, you are recommending things to users, so you could also know something about them. User features are typically related to the user specifically, so where do they live? What are their interests? Which gender? Well, you start running into privacy issues, so this is a delicate

type of information. The age range, for example.

(11:28 - 11:58)

So, if you happen to register in a platform, and that platform asks you to provide a selection of interests, this is what the platform was doing. So, the platform was trying to infer some user data from you to begin providing recommendations or filtering the results in some way. Now, the problem with user data is that users really don't like to give personal data, which is totally reasonable.

(11:59 - 12:19)

Also, user data is generally uninformative. Consider a situation in which you have, for example, the location and age range. How am I supposed to find a recommendation list that is adequate for every person in this classroom? I can't.

(12:21 - 12:35)

Oh, a funny story. Again, from Netflix, I think. Several years ago, at an event, they... I don't remember who was it, but anyway.

(12:35 - 12:51)

When you register in Netflix, you have the option to select the state you live in. Now, if you live in the United States, you have, well, two options. First, you have to say that you are in the United States, and then in which of the 50 states you live in.

(12:53 - 13:14)

Would you have guessed that I think 70% of their subscribers lived in Alabama? Of course not. It simply was the very first item in the list. So, yeah, you can collect user data.

(13:15 - 13:24)

It's probably rubbish. So, pros and cons. Oh, a last story.

(13:25 - 14:13)

If you register into a platform, and the platform asks you to use your Google profile, or if you are a boomer like me, Facebook profile, or any other sort of similar thing, what they are doing is that, yes, they're making it easier for you to log in, but they are also going to siphon some personal information from there to be able to build an initial profile. Whether they do this to a significant extent, or whether the information they can gather is very limited depends on the policies of the platform, blah, blah, blah. But frankly, who of you ever read a privacy statement? Okay.

(14:15 - 14:38)

Oh, I forgot to mention, the item data is usually much better than the user data, but it's also quite difficult to collect. So, if you want to categorize things, if you want to build a giant database with a description of every specific aspect of a movie, it takes a bunch of people a lot of time. So, both things are quite expensive to get.

(14:40 - 14:53)

Third option. You use the interactions between the users and the items. What does it mean? Well, you just record, log, what the users do.

(14:55 - 15:30)

For example, this user, you know nothing about this user, you just know that the user 1525 interacted with this bunch of items. What does it mean, interact? It means watching a movie, listening to a song one or one thousand times, if you're a compulsive listener, like I am, or perhaps purchased a product, possibly returned the very same product. So, you collect all these interactions within the platform over a long time frame, and you collect data in this way.

(15:32 - 15:47)

Advantages? One, well, you just let the platform do its thing, you don't need any additional data. We'll spend again a lot of time on discussing these later on. Disadvantages? A few.

(15:49 - 16:28)

First, there's, I would say, one billion, it's possibly a bit less, but around this number of products on Amazon, how many could you possibly buy after a few years? I don't know, if you buy a lot, a hundred maximum. So, what about everything else? So, type of data, this type of data is very easy to collect because you just let the platform do its thing, but it's also incredibly sparse. So, you're going to have an enormous amount of possible user-item combinations that never occurred.

(16:28 - 17:24)

It is frequent, if you work with real datasets, to have that perhaps only 10 to the minus 5 of the possible user-item combinations actually occurred. The sparsest dataset I ever worked with is Goala, which is an online check-in platform for points of interest, and its density was 10 to the minus 7, which means that every 10 million possible user-item combinations, only one occurred, which is extraordinarily sparse. So, yes, this type of data is easy to collect, and you may end up with billions and billions of interactions, but still those interactions will cover only a tiny fraction of all possible combinations, which means that the machine learning models and heuristic models we build must take this into account.

(17:25 - 18:09)

This is, I think, one of the most striking differences between this field and several other data

science-related, or let's say classification and machine learning-related fields. They have more data than we do, and this has a lot of implications. Upside.

It works really well. If you try to use what the user is doing to understand their preferences, it's typically much better than paying a bunch of people to try to infer what type of characteristics of an item are informative. So, there's a bunch of trade-offs.

(18:12 - 19:10)

This is just to give you a sense of the fact that although the type of data in terms of data structure is simple, all of these are going to be just a table, the semantics of the data, the way the users will interact with the system, and the information you can get from the data is going to affect which design choices you make. Oh, and then there's also the context. Say that you like science fiction.

Fine. And that occasionally on YouTube you watch these nostalgia trips on the seventh season of Star Trek Deep Space Nine, as I do, again. Say, however, that you also like the seminars that discuss the historical events from Alessandro Barbero, or that you also like pop music.

(19:11 - 19:37)

And so you have this type of varying tastes that combine your own interests. Which of these are you going to like at this time? Well, none of them, possibly, because you're attending my lecture. But how can a system know which one to suggest? If I'm going for a run, I want energetic music.

(19:38 - 20:08)

If I need to focus, I want the white noise with a hair dryer, or fans, or chitchat in the background. How do I understand when I should recommend one or the other? If you are watching a movie alone, you may watch a certain type of content. If you are with your kids there, or your nephews, or your fiancé, or whatever, you may choose something different, both of which, or all of which, you're going to like to various degrees.

(20:08 - 20:34)

The point is that you have a different intent. This is the context. What time it is? Where are you? With whom you are at this time? What are you doing? All of these are information that typically the model doesn't have, because, as you can imagine, they are delicate to gather and store, and it's a nightmare in terms of privacy, of course, but that inform the recommendation model.

(20:37 - 21:05)

For example, again, if I'm recommending to you something and you don't click on it, what does it mean? Does this mean it was an unsuccessful recommendation? Possibly, but possibly not. Maybe it was the type of splatter movie that you watch alone, and then, at the time, you have

your kids there, and so it's not suitable for them at this time. So, you got the wrong context.

(21:16 - 22:20)

And let's do another slide switch. So, which type of information is most used to determine whether two items are similar? The answer is, as everything, it depends. If you have a lot of interactions, you might use that one.

(22:21 - 22:35)

If you don't, you might use item data. I'll give you an example. Say that you are in a scenario such as news recommendation where the expiration date of the item is very close.

(22:35 - 23:02)

You're not going to recommend someone a news that is five days old, or, well, possibly you can, but the value is going to be low. So, either it is one of those very large journalistic endeavors from The Guardian or from this type of investigative journals, or it's going to be old news. So, in that case, you cannot really use user-item interactions because there's not going to be the time to collect them.

(23:03 - 23:23)

So, you need item descriptors. In other cases, movies, for example, it's extremely difficult to get good item descriptors. And so you try to use user interactions because, well, the movie can stay on your platform for months, and so you have the time to collect the data.

(23:23 - 23:35)

So, all of these things will play a role. But, again, we will discuss those aspects in a bit. Okay, so, let's start on the taxonomy.

(23:35 - 23:52)

Ooh, I just now realized I completely forgot to tell you a rather important thing. Tomorrow's lecture is suspended because we are off to a great start and I have to go to a conference in Rome and couldn't get anyone else to replace me. So, surprise.

(23:55 - 25:02)

Okay, now, back to the taxonomy. First, very broadly speaking, we can separate algorithms in two main categories, non-personalized and personalized. Any of you care to make a guess on what a non-personalized algorithm is? Or suggest one? No guesses? Hmm? Well, that is, sorry, what? That is a sorting algorithm.

(25:04 - 25:21)

For example, the most popular. So, you open your Netflix and you have, like, most popular movies or acclaimed by the critic. It's the list and everyone is going to see the same list.

(25:24 - 25:47)

Possibly highest rated. So, every scenario in which you have a list of recommendations that is defined based on a criteria which does not take you into account is non-personalized. These can still be good recommendations or great recommendations.

(25:48 - 26:15)

You may have curators. So, every now and then, you have some professionals who, for various types of companies, design editorially curated recommendation lists which are typically hidden among the various options where everyone is going to see the same list. So, this is a type of non-personalized recommendation.

(26:17 - 26:43)

Now, are non-personalized recommendations effective? It depends on the domain, again, and on what you're trying to do. Consider a scenario where your non-personalized recommendations are simply a list from the most popular movies to the list. So, you take a bunch of the highly popular movies or products on the platform.

(26:44 - 27:01)

Is this list going to be effective? Well, probability-wise, yes. The very definition of a popular movie is a movie a lot of people have liked. And so, if you recommend this to a new user, well, you're kind of running back to mama, yes.

(27:02 - 27:20)

But likelihood that the user is going to like it is rather high. You can do better, though. Personalized algorithms take your past behavior into account in a number of ways.

(27:22 - 27:57)

The idea is that you want to infer past preferences from anything that the user has done in the past and use them to decide what to do next. So, two people will see different recommendation lists based on what they did. The thing is a bit laggy.

(27:58 - 28:20)

Ha! So, now our goal is to personalize our recommendations. But then, how do we understand what a user likes? There's a bunch of methods, a bunch of techniques. The first two, which we will see, are content-based and collaborative filtering.

(28:22 - 28:46)

And here we are going back to the 90s because I believe that the very first content-based filtering method was introduced in 1993. So, it's a 30-ish year old discipline. What is a content-based filtering method? Well, it's any type of method that uses item descriptors.

(28:48 - 29:01)

So, the first type of information we saw in the previous slides. What does it mean? Well, if I liked, for example, the first Fellowship of the Ring movie. I don't actually remember the full title, but still.

(29:03 - 29:07)

Sorry, that was a title of one of them. The Lord of the Rings. So, the first movie in The Lord of the Rings.

(29:12 - 29:42)

The model looks for other movies that share the same descriptors, the same cast, the same possibly similar titles, possibly similar descriptions, the same director, the same producer. And we look for the most similar item in the catalog with those characteristics. It is likely going to be another of the Lord of the Rings movies.

(29:42 - 29:59)

Or if you like Star Wars, it's possibly going to be another of the Star Wars movies because they have very similar features. Seems reasonable, yes. Let's take another example.

(30:00 - 30:17)

Hotel recommendation. Okay, I want my included breakfast, I want my free cancellation, I want my four stars and blah blah. So, if I want to recommend you a certain hotel in a new location, I can use this information.

(30:17 - 30:31)

I can go and look for the hotel with the closest characteristics and rank based on that. Again, it's reasonable. Last example, say you just bought an expensive smartphone on Amazon.

(30:32 - 30:44)

Next day, you open the platform again. You don't input any query and look at the recommendations. And you see a lot of other expensive smartphones.

(30:45 - 31:15)

Does it make sense? Eh, not really. So, the issue might be here that in some domains, it doesn't really make sense to recommend things that are too similar to what the user has done or has

bought. In this case, it would have made more sense to suggest things that complement the purchase.

(31:15 - 31:29)

A new cover, a new charger, a new, I don't know, tempered glass screen or whatever. A new SD card. So, this is something that must be taken into account.

(31:32 - 32:05)

Another problem with content-based recommendations is that if you continue to recommend the very same thing or things that are extremely similar, you're going to get the user bored. The recommendations are going to be monotonous, repetitive. And so, you're going to lock the users in what is called a filter bubble in which the type of result they get from your system is more or less the same or extremely similar.

(32:06 - 32:18)

Now, you might like it. It's reasonable, again. But you might also not because what was the purpose of a recommendation system in the first place? To help the user explore.

(32:18 - 32:57)

And if you end up in a situation where your system is providing trivial and monotonous recommendations where you're not really doing a good job, yes, it makes sense for you to provide things that the user likes, but you must also make sure that you add new things. If the user gets bored, they will leave the platform. And what is going to happen if they do? The management is going to come to your office and ask which engineer built the system and get mad, which means mad at you, and it is something we don't want.

(33:01 - 33:33)

Oh, aside from jokes, filter bubbles can also have significant impacts. Think, for example, about news recommendation. Let's say that you watch your news feed from Google or any of these personalized news feed platforms, and that at some point you start seeing every day or every couple of days an article on how China is going to dominate the world in the future due to its economic growth.

(33:34 - 34:36)

And how the European Union is losing competitiveness, and this type of economic news. Or let's say that every day you hear about the last outrageous thing a certain politician, which is typically the opposite of whatever political preference you have, said. Or what was the last tragic accident in a factory, or what was the last I don't know, the most recent criminal behavior of an immigrant of any type of provenance.

That is going to impact your view of the world. And so if after weeks and weeks and months you are only exposed to a certain type of information, you're going to have a certain perception of reality. Which is quite different from what happened in the past when you would only buy the newspaper of the company you felt closer to your political preferences.

(34:37 - 34:55)

Because recommendation is much more fine grained. And if you like only a certain type of content you end up in an enormous confirmation bias bubble in which you are only ever going to be exposed to news that agree with you. And this has been recognized as a growing problem.

(34:57 - 35:57)

So societal implications are substantial. Okay, but why should I care about societal implication? Maybe you don't, maybe you care about yourself. Fine, what about a filter bubble on job recommendations? What about realizing that for some reason that no one knows, simply because no one has paid attention when the system was built, a person with certain characteristics consistently receives worst job recommendations.

Worst job recommendations for positions with a lower salary. This might be reasonable if the competencies, skills are different. But sometimes these things happen even when it's not.

And when this happens, it's a problem. Maybe you don't realize it yet, or maybe you have no idea, but this can have an impact on the life of people. So take that into account.

(36:01 - 36:39)

Okay, now I think I've lost my bearing. One second. Okay, okay, now on lighter tones.

Let's go back to the taxonomy. The next item on the list is collaborative filtering. As the name suggests, it is based on collaborative information, which means user item interactions.

(36:40 - 36:57)

So whatever the user did with the system. In this case, we are not using any item or user descriptor. So we are not relying on any pre-built knowledge on what the item is.

(36:57 - 37:38)

We might not even know anything at all, just a number, just an ID. This type of recommenders was invented, again, 30 years ago. I think I have written down a date.

I think it was deployed by Amazon in 1998. Yes, so again, long running type of methods. The general idea of collaborative filtering is to look for other people that have interacted with items in a similar way.

(37:39 - 38:14)

Actually, there's two. The first idea is let's say I interacted with two or three movies. In a user-based collaborative filtering method, what the algorithm does is to look among the available users for other people who did the same.

(38:15 - 38:35)

So to look for a bunch of other people who also interacted with the things I interacted with. And once I found them, you look at what else they have done. You look to which other items they have interacted with and use those as recommendations.

(38:35 - 39:00)

So in a user-based recommendation algorithm, you go look for similar users and you import, acquire, and use their preferences. Again, in a bunch of ways, but this is the general idea. This is, I believe, the very first collaborative filtering algorithm.

(39:00 - 39:05)

Again, 1998. The idea seems reasonable. Yes.

(39:06 - 39:36)

But there's a few catches. What happens if you only interacted with one item? Okay, maybe we will have 100,000 other users with the same interaction. How do I choose? How do I choose which ones are similar to you? I only have one data point.

I can't really. Well, I can. I will calculate a recommendation list, but that recommendation is not going to be very good.

(39:40 - 40:29)

What if you're a new user? I'm screwed if you're a new user because I know nothing about you at all and so I have absolutely no way to know to which other people you are similar. There's a few failure cases here as well. Typically, user-based methods are quite unstable because, well, you have, I don't know, three interactions.

Fine. You do something else. Now you have four.

It's only one interaction, but you've changed the number of data points I had about you significantly. And so the recommendations you receive are also going to change significantly. If you interact with something else, we go from four to five.

(40:29 - 40:43)

Again, one interaction, but proportionally it's still a lot. As you navigate through the system and

as you do, you interact with it, the recommendations you receive are going to be quite unstable. This is annoying for a lot of people.

(40:45 - 41:11)

So, pros and cons. It's also terribly cumbersome in terms of scalability because it is very frequent for a platform to have many more users than items. Okay, yes, Amazon has a billion products, but anyone else has much less and it's much easier for them to have... For example, let's consider a video streaming platform.

(41:12 - 41:41)

Their catalogs, such as Netflix and Paramount Plus and Disney, whatever, so their catalogs are typically of a few thousand items. They frequently have hundreds of thousands or millions or tens of millions of users. So, if you want to build recommendations and you first need to look through all the user database to find who is similar to you, well, you can precompute some of that, yes, but it's a lot of computation.

(41:43 - 42:17)

Again, all of these things will be touched on as we go forward with the course, just to give you a sense of how many, just how many different dimensions and things to take into account there are. Okay, I think I'll spend a few moments on the next collaborative filtering algorithm and then we will adjourn. We have an item-based collaborative algorithm, we also have a... Sorry, we have a user-based one, we also have an item-based one.

(42:17 - 42:44)

The principle is the same, we just switch perspective. What we try to do is we consider two items similar if users have interacted them in a similar way. So, if a couple of items have received the same interactions they are going to be similar.

(42:45 - 43:03)

Again, consider one of those I don't know, one of the Star Wars movies. There's going to be a bunch of people who interacted with most of the movies from that series. Due to this, these items are going to be considered similar.

(43:03 - 43:34)

It's again the same perspective, we will see that in practice you are transposing a matrix but well... Why is in practice item-based recommendation different? Well, there's a number of factors at play. You have less items and so it's more scalable. It's less unstable because a user might have very few interactions but an item has thousands.

(43:35 - 44:11)

So if a bunch of new people enter the system and do things, well my new data points are going to end up in a model that is not changing very much. Maybe I'm adding just 1% of the interactions of that item instead of 30% that I had with the user. How do you choose which one do you want? Well, in practice you choose which works best and you choose the one that is consistent to your computational budget.

(44:12 - 44:47)

As you will see in the coding session, the training time is going to become a factor. In the coding sessions you will do experiments with real-ish data sets with at least some tens of thousands of items and users, so we're still going to we're still talking about a few seconds for heuristic methods of training time and possibly a few minutes for machine learning methods, but you're going to see the difference between a scalable recommender system algorithm and a method that is not. And so this dimension is too important.

(44:57 - 45:14)

I hate to interrupt in the middle of the taxonomy, but we are more or less out of time and I need to go to Rome. So... Okay, so I'll be here for a few minutes if you have questions as I imagine some of you will. If not, see you next week.

Goodbye.