Practical Recommender Systems

Kim Falk

MEAP

MANNING

MEAP Edition
Manning Early Access Program
Practical Recommender Systems
Version 16

For more information on this and other Manning titles go to
www.manning.com

welcome

Thank you for purchasing the MEAP edition of *Practical Recommender Systems*.

Most web applications are only a short distance from utilizing machine learning and recommender algorithms to improve their customer experience. The book will demonstrate how to implement such algorithms, with emphasis on real-world systems.

To get the most out of this book, you'll need to be comfortable with developing web applications, from deep down in databases all the way up to HTML and JavaScript.

We'll start out by introducing the basics of recommender systems – What are they made of? How do they work? How do can they guess what movie I might want to watch this afternoon?

We'll create our working environment with a movie recommendation website, generate data to work with while you learn algorithms, and explore personas and different ways to understand data. All this will be programmed Python, using Django, PostgreSQL, a bit of mathematics, and a lot of common sense.

This foundation work is necessary before we begin to explain recommendation algorithms that you will be able to implement and test on your own sandbox website. Discussion of algorithms will focus on implementation and tricks needed to make them perform. I'll provide ample examples of how these algorithms are in use existing sites and inspire the reader on how to apply this to their site.

We hope you that you enjoy *Practical Recommender Systems* and that it will occupy an important place on your digital (and physical!) bookshelf.

—Kim Falk

brief contents

1
What is a Recommender

1.1 Real-life recommendations

I lived for years in Italy, in Rome. Rome is a beautiful place, with lots of food markets—not the central ones found in guide books, that are full of knock-off Gucci bags, but the ones that are outside the regular tour bus route, where farmers come and sell their products. Every Saturday we would go to see a greengrocer named Marino. We were good customers, real foodies, so he knew that if he recommended good things to us, we would buy them—even if we had strict plans to buy only what was on our list. The watermelon season was great, the many types of tomatoes offered a fountain of various flavors, and the mozzarella tasted unforgettable. Marino would also recommend for us *not* to buy something if it was not top quality, and we trusted him to give us good advice. This is an example of *recommendations.* Marino recommended the same things repeatedly which is okay with food, however that is not the case for most other types of products like books or movies.

When I was younger, before Spotify and other streaming services took over the music market, I liked to buy CDs. I would go to a music shop that catered mostly to DJs, and I would walk around and gather a stack of CDs, then find a spot at the counter with a pair of headphones, and start listening. With the CDs as context, I had long conversations with the man behind the counter. He would check which of the CDs I liked (and didn't) and recommend others based on that. I prized the fact that he remembered my preferences well enough between visits and didn't recommend the same titles to me repeatedly. This is also an example of *recommendations.*

Getting home from work, I always look in our mailbox to see if we got mail. Usually, the mailbox is full of advertisements from supermarkets, offering things that are on sale. Typically, the ads show pictures of fresh fruit on one page and dishwasher powder on the next—all things that supermarkets like to recommend that you buy because they claim it is a good offer. *Those are not recommendations; they are advertisements*.

Once a week, the local newspaper is among the mail. The newspaper features a top-10 list of the most watched movies at the theater that week. *That is a non-personalized recommendation.*

On television, a lot of thought goes into placing commercials within the right television content. Those are *targeted commercials*. Because they expect certain type of people watching.

In February 2015, Copenhagen Airport announced that it placed 600 monitors around the airport to show commercials based on the viewer's estimated age and gender, along with information regarding the destinations at the nearby gates[1]. The age and gender are inferred using cameras and an algorithm. A woman traveling to Brussels wants to see nice watches or an ad for a finance magazine, for example. A family going on vacation might be more interested in ads for sunblock or car rentals. *These are relevant commercials or highly targeted commercials*.

People usually perceive commercials on television or at the airport as a nuisance, but if we go online, the limits to what we consider invasive become a bit different. There could be many reasons for this, which is a whole topic in itself. The internet is still the Wild West, but although I think that the advertising at the Copenhagen Airport is quite invasive, I also find it irritating when I see advertisements on the net that are directed at a target group that I'm not part of. To target its commercials, websites needs to know a bit about who you are.

In this book, you'll learn about recommendations, how to collect information about the recipients of the recommendations, how to store the data, and how to use it. You can calculate recommendations in various ways, and you'll see the most used techniques. A recommender system is not just a fancy algorithm, it is also about understanding the data and your users. There is a long going discussion whether it is more important to have a super good algorithm or to have more data. Both has flipsides, super algorithms will require super hardware and lots of it. More data will create other challenges, like how to access it fast enough. Going through this book you will learn about the tradeoffs and get tools to take better decisions.

The examples above were meant to illustrate that commercials and recommendations can look similar to the user. Behind the screen the intent of the content is different, a recommendation is calculated based on what the active user likes, and what others have liked in the past, and is often requested by the receiver; a commercial is given for the benefit of the sender and is usually pushed on the receiver. The difference between the two can become blurry; in this book, we'll call everything calculated from data a *recommendation*.

---

[1] **www.egmont.com/int/Press/news-and-press-releases/Airmagine-revolutionises-advertising-in-Copenhagen-Airport/**

1.1.1 Recommender systems are at home on the internet

Recommenders are most at home on the internet because this is where we can not only address individual users but also collect behavioral data.

A website showing top-10 lists of the most-sold bread-making machines provides *non-personalized* recommendations. If a website for home sales or concert tickets shows you recommendations based on your demographics or your current location, the recommendations are *semi-personalized*. *Personalized* recommendations can be found on Amazon, where identified customers see "recommendations for you." The idea of the personalized recommendation also arises from the idea that people are not all interested in the popular items, but also items that are not sold the most, items that are in the long tail

1.1.2 The long tail

The term *the long tail* was first coined by Chris Anderson[2]. He identified a new business model which is frequently seen on the internet. If you don't have the limitations of a bricks-and-mortar shop, which is the limited amount of storage space, and more importantly you have a limited space to show products to users, but also limited reach since people have to come to your shop. Andersons insight was that without these limitations you didn't have to sell only popular products, as was the usual commerce business model. This idea was long considered a losing strategy as you would need to store a lot of products that might never be sold. But if you have a webstore then you don't need to pay an expensive rent since your storage can be in places where rent is cheap, or if you sell digital content, it basically doesn't take up any space at all. The idea of long tail economy is that you can also get rich by selling lot of products, but only a few of each, and to lots of different people.

I am all for diversity, so I think it is great, with a huge catalogue then the question of how users finds what they want is difficult to answer. And this is where recommender systems have their entrance. Because they help people find those diverse things that they would otherwise not know existed.

On the web, Amazon and Netflix are considered among the biggest giants both in content but also in recommendations; they're used in numerous examples throughout this book. In the following section, you'll take a closer look at Netflix as an example of a recommender system.

1.1.3 The Netflix recommender system

As you likely know, Netflix is a streaming site. Its *domain* is that of films and TV series, and it has a continuous flow of content available. The *purpose* of Netflix recommendations is to keep you interested in its content and to keep you paying the subscription month after month.

---

[2] https://en.wikipedia.org/wiki/The_Long_Tail_(book)

Netflix wants to show you content that will keep you there for as long as possible. The service runs on many platforms, so the *context* of the recommendations can differ.

Figure 1.1 shows a screenshot of Netflix from my laptop. I also access Netflix from my TV, my tablet, or even my phone. What I want to watch on each platform varies; for example, I would never watch an epic fantasy film on my phone, but I love them on TV.

Let's begin this walk-through by looking at that startup page. I have an account, and when I log into Netflix, I get the personalized start page shown in figure 1.1. The front page is constructed as a panel containing a vertical list of rows with subjects such as Top Picks, Drama, and Popular on Netflix.

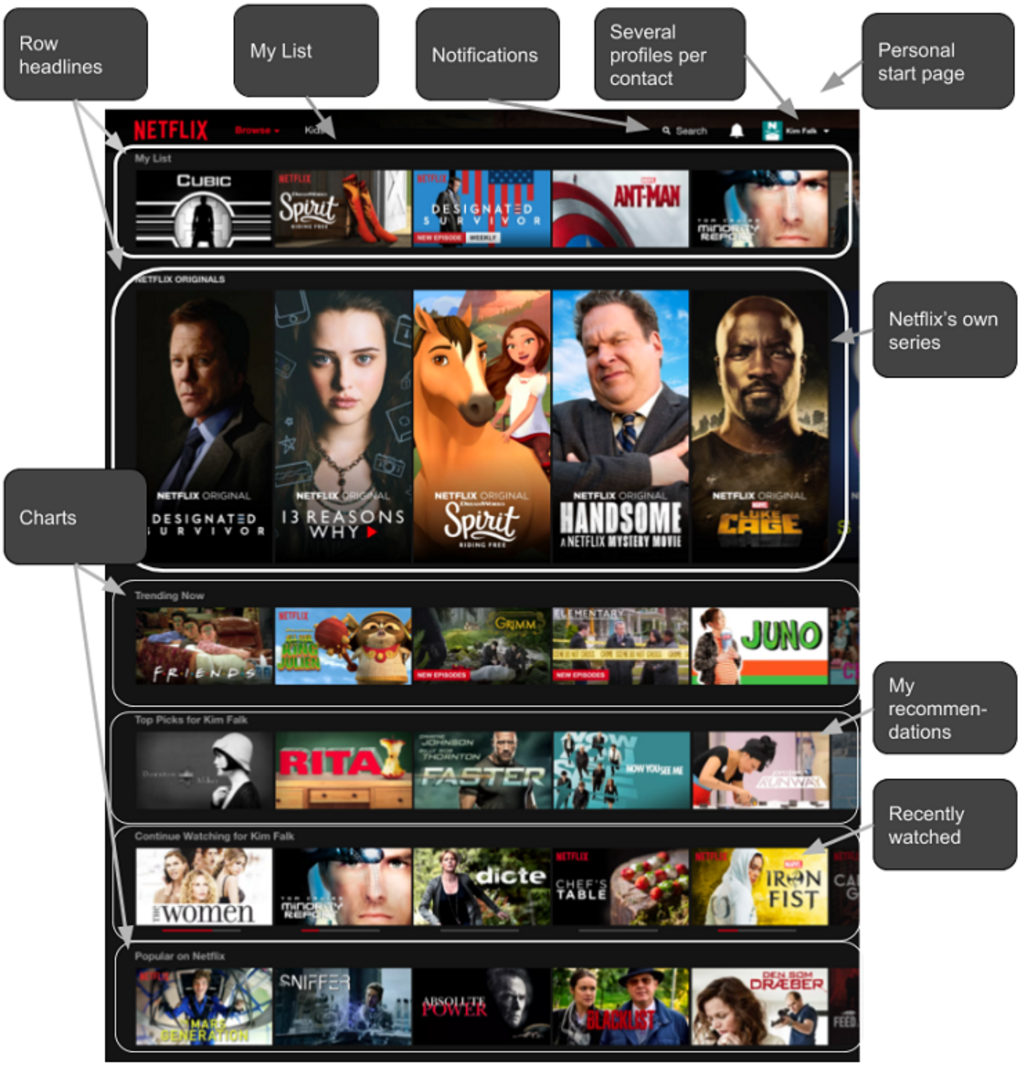

Figure 1.1 The Netflix start page, before they changed the layout.

The top row is dedicated to what what's on my list. Netflix loves the list, because it indicates not only what I've watched and what I'm watching now, but also what I (at least at some point) have shown an interest in seeing. Netflix wants you to notice the following row, as that contains the *Netflix Originals* – the series which are produced by Netflix, these are important for Netflix for two reasons, they have spent a lot of money on producing them, and are in most cases only found on Netflix, and second, Netflix has to pay content owners when users what's their content, but if that owner is Netflix then they save money. This also illustrates a point to consider, even if everything is personalized on the page, then the fact that the Netflix Originals are second row, probably isn't a result of me watching them, but rather a pursuit of an internal business goal.

CHARTS AND TRENDS

Next, we see the *Trending Now* list. Trending is a very loose term, which can mean many things, but has something to do with something that has been very popular within a very short period. The bottom row *Popular on Netflix* also has something to do with popularity, but over a longer period, maybe a week. Trends and Charts will be discussed in detail in chapter 5

RECOMMENDATIONS

The fourth row is the list of Top Picks for me and my profile. This list contains what most people would call recommendations, it shows what the Netflix recommender system predicts I'd like to watch right now. It looks almost right; I'm not into bloody, gory movies. I'd rather not see any dissections of bodies at all (which seems to be a clear sign that I've lost my connection with today's young people). Not all of the suggestions are to my liking, but I assume that it's not just my taste that Netflix uses to build this list. The rest of my household also watches content using my profile at times. Profiles are Netflix's way of letting the current user indicate who is watching.

Before introducing profiles, Netflix aimed its recommendations at a household rather than one person; it would try always to show something for mom, dad, and children[3]. But Netflix has since dropped that, and at least now my list doesn't include any children's show.

[3] http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html

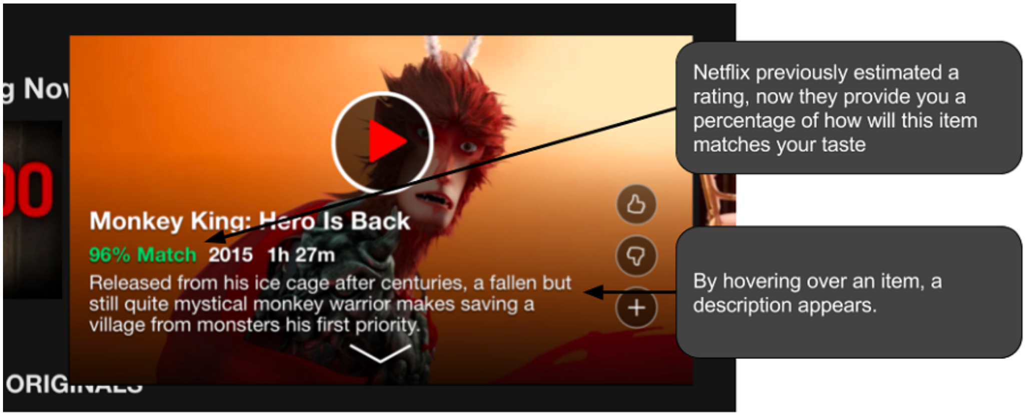

Figure 1.2 A Netflix Top Pick with predicted match

But even if Netflix is using personal profiles, I think that it is imperative to consider who is watching—not just the person with the profile, but also who else. I've heard rumors that other companies are working on solutions enabling you to tell the system that other people are watching too, to allow the service to deliver recommendations fitting all members of the audience, but so far, I haven't seen any in play. Microsoft Kinect has the functionality to recognize people in front of the TV by using face/body recognition. Kinect even takes it a step further by identifying not only household members, but other people from its full catalog of users, allowing Kinect to recognize users when they're visiting other homes.

ROWS AND SECTIONS

Back to the Top Picks of Netflix, you can find more details on the content by hovering your mouse over one of the suggestions. A tooltip appears with a description (see figure 1.2) and a predicted rating, which is what the recommender system estimates I would rate this content. You might expect that the recommendations in the "Top Picks for me" row, all have a high rating, like the one in figure 1.1, but looking through the recommendations, you can find examples of items with a low predicted rating, as shown in figure 1.3.

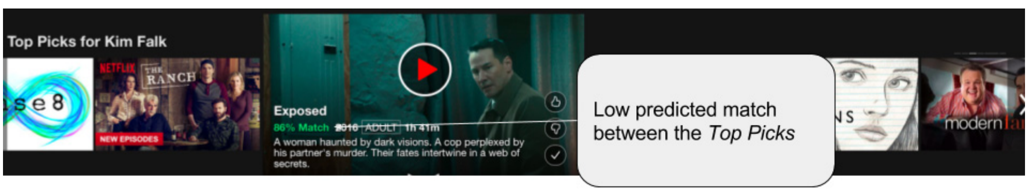

Figure 1.3 Netflix: Low predicted rating on a Top Pick

The ways of the Netflix recommender are many, so there are many possible explanations as to why Netflix chooses to recommend an item that it predicts that I wouldn't rate highly. One reason could be that Netflix is aiming for diversity over accuracy. Another reason could be that even if I wouldn't rate it maximal stars, the movie might still be something that I'm in the mood to watch now. This is also the first hint that Netflix doesn't put much value on ratings.

The titles of each row are different; some are of the type *Because you watched Suits*. These lines try to recommend things that are similar to *Suits*. Other rows are genres such as *Comedies*, which, curiously enough, contains comedies. You could say that the row titles are also a list of recommendations; you could call these *category recommendations*.

This could be the end of the story, but then you'd miss the most important part of the Netflix personalization.

RANKING

Because each of the headlines describes a set of content, this content is then ordered according to a particular recommendation system and presented to you in order of relevancy or rank, starting from the left, as illustrated in figure 1.4.

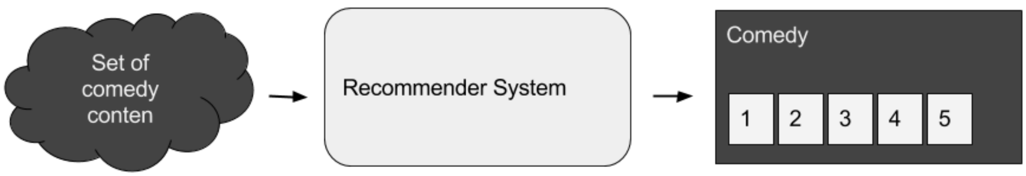

Figure 1.4: Each row is ordered by relevance.

Even in My List, which contains the content I've selected myself, the content is ordered according to the recommender system's estimate of its relevance for me now. I added the screenshot in figure 1.1 yesterday. Today my list has a new order, as shown in figure 1.5.

The Netflix recommender system will also try to recommend content that's relevant at that specific time or in a particular context. For example, Sunday mornings might be more for cartoons and comedies, whereas evenings might be for more “serious” watching of series like *Suits*.

Another row that might be surprising is Popular on Netflix, which shows content that’s popular right now. But Netflix doesn’t say that the most popular item is the one all the way to the left. Netflix finds the set of most popular items and then orders them according to what you would consider most relevant now.

BOOSTING

A point to ponder is why Netflix adds the show *Suits* in My List, considering that I’m already watching it. Looking at figure 1.5, you can see that among the notifications, Netflix indicates that a new season of Suits is out; this could explain why this show shows up. *Boosting* is a way for companies to put a finger on the scale when suggestions are calculated; Netflix wants me to notice *Suits* because it is new content and therefore has a freshness value. Netflix boosts content based on freshness; *freshness* can mean that it’s new, or it has been mentioned in the news, or somebody recently wrote an excellent review. Boosting is covered in more detail in chapter 6, as it’s something that many site holders will request as soon as the system is up and running.

Please note, that there is a machine learning algorithm family called boosting[4]; what I am referring to here is something different.

SOCIAL MEDIA CONNECTION

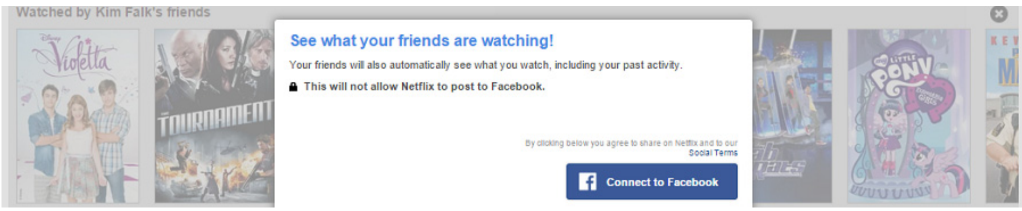

Figure 1.6 Netflix wants to know what my friends are watching

For a short period of time, Netflix also tried to use social media data[5]. If you scrolled down in your netflix page you would find something like what is shown in figure 1.6.

Netflix encouraged you to enable Facebook Connect, thereby allowing Netflix access to your friends list as well as other information. One of the advantages for Netflix is that it can find your friends and make social recommendations, based on what your friends like. Connecting with Facebook could also make watching films a much more social experience,

---

[4] https://en.wikipedia.org/wiki/Boosting_%28machine_learning%29
[5] https://www.cnet.com/how-to/get-to-know-netflix-and-its-new-facebook-integration/

which is something that many media companies are exploring. In this day and age, people don't sit down to watch films passively; they watch a movie while sitting with a second device (such as a tablet or a smartphone). What you're doing on the second device can have a large influence on what you watch next. Imagine that after you watch something on Netflix, a notification pops up on your phone that one of your friends liked a film, and presto, Netflix recommends that as the next thing to watch.

This social feature was however removed again somewhere between 2015-2016 with the argument that people were not happy with sharing their films with Facebook network. Which in the words of Netflix CPO Neil Hunt *is unfortunate because I think there's a lot of value in supplementing the algorithmic suggestions with personal suggestions*[6]*.*

TASTE PROFILE

With a page that's built almost entirely based on suggestions, it's a good idea to provide as much input as possible on your tastes.

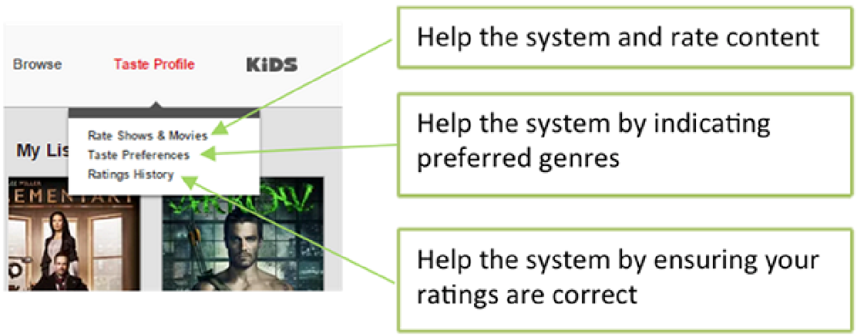

Figure 1.7 Example from how the Netflix taste profile looked in 2015

If Netflix doesn't have a clear sense of your taste, it can be hard for you to find what you want to watch. In 2016 Netflix had options which allowed users help build their taste profiles. The Taste Profile menu, shown in figure 1.7, enabled you to Rate Shows and Movies, to select genres by saying how often you feel like watching, for example, Adrenaline Rush content as illustrated in figure 1.8, or to check whether your ratings match your current opinions.

[6] http://www.businessinsider.com/netflix-users-dont-want-social-features-2016-2?r=US&IR=T&IR=T

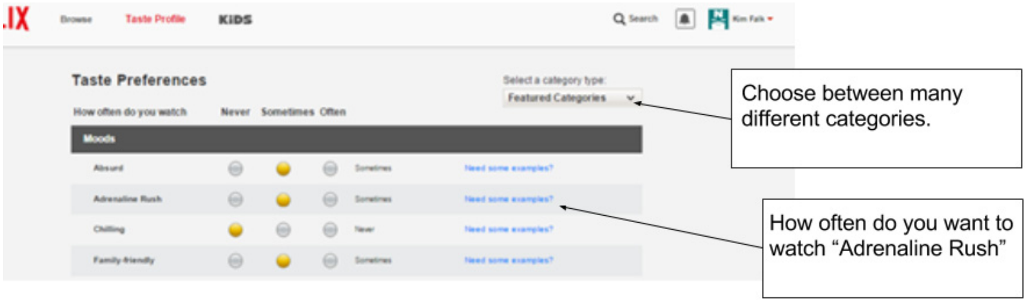

Figure 1.8 Netflix Taste Preferences

The manually inputted taste preferences enable Netflix to provide better suggestions. Asking the user for help with the taste profile is a method often used to allow the system to give suggestions for new users. But, as with so many things, there's often a difference between what users say they like and what they actually like. So, although users might provide more usage data, that explicit input is used less and less. And in fact Netflix has now removed the feature.

1.1.4 Recommender system definition

To be sure we're all on the same page, let's put down some definitions, as shown in table 1.1.

Table 1.1 Recommender system definitions

Term	Netflix example	Definition
Prediction	Netflix guesses what you will rate an item.	A prediction is an estimate of how much the user would rate/like an item.
Relevancy	All rows on the Netflix page (for example, Top Picks and Popular on Facebook) are ordered according to relevance.	An ordering of items according to what is most relevant to the user right now. Relevance is a function of the context, demographics, and (predicted) rating.
Recommendation	Top Picks for me.	The top N most relevant items.
Personalization	The row headlines in Netflix are an example of personalization.	Integrating relevancy into the presentation.
Taste profile	See figure 1.8.	A list of characterizing terms coupled with values.

With these definitions in place, we can finally define a recommender system.

DEFINITION: Recommender system

A recommender system can calculate and provide relevant content to the user, based on knowledge of the user, content, and interactions between the user and the item.

With a definition in place, you might think that you have it all figured out. But let's go through an example of how a recommendation could be calculated and how it would work. Figure 1.9 shows how Netflix might produce my Top Picks row.

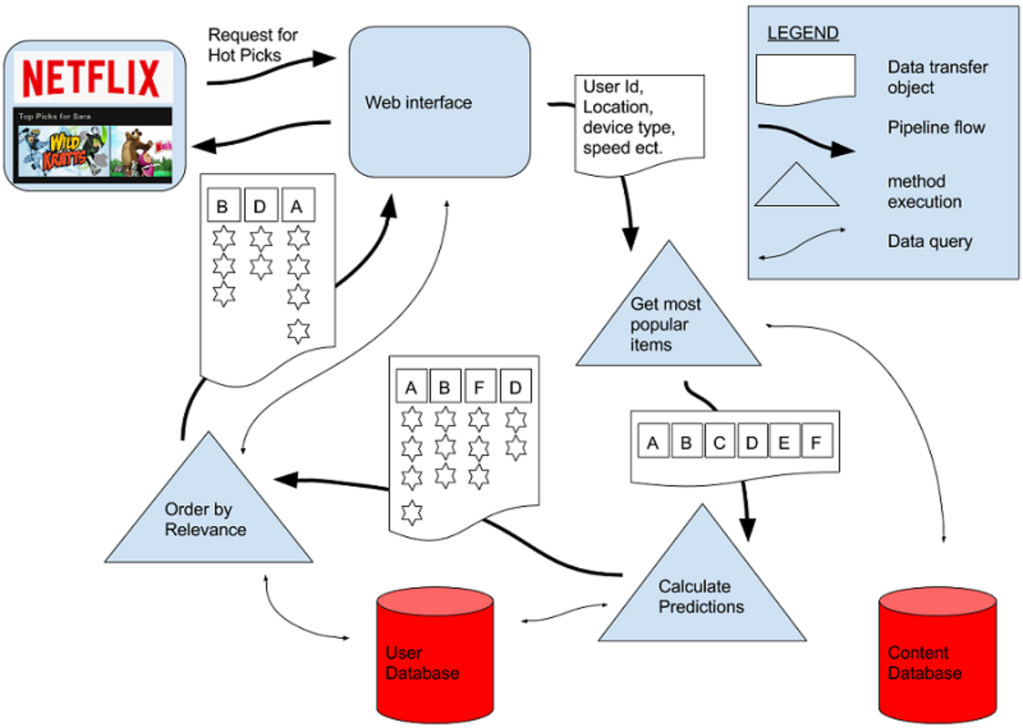

Figure 1.9: How Netflix Top Picks might be calculated

Here are the steps:

1. A request for the Top Picks list is received.
2. The server calls the recommendation system, which consists of a pipeline of methods. The first step is called `retrieve candidate items`. It Retrieves the items that is most similar to the current user's taste. from the catalog database.
3. The top five items (normally it could be 100 items or more) are piped into the next pipeline step, `calculate prediction`.

4. Prediction is calculated using the user taste retrieved from the user database. It's likely that the prediction calculation will remove one or more items from the list, due to their predicted rating being too small. In figure 1.9, two items are removed (items C and E).
5. The `calculate prediction` step outputs the significant items, now with a predicted rating added to them. The result is piped into an `order by relevance` process.
6. The `order by relevance` step orders the items according to the user taste, context, and demographics. It might even try to add as much diversity to the result as possible.
7. The items are now ordered by relevance. Again, item F was removed because the relevance calculations showed that one of them would not be relevant for the end user.
8. The pipeline returns the list.
9. The server returns the result.

Looking at figure 1.9, it becomes evident that there are many aspects to consider when working with recommender systems. The preceding pipeline is also missing the parts of collecting the data and building the models. Most recommender systems try to use the data shown in figure 1.10 in one way or another.

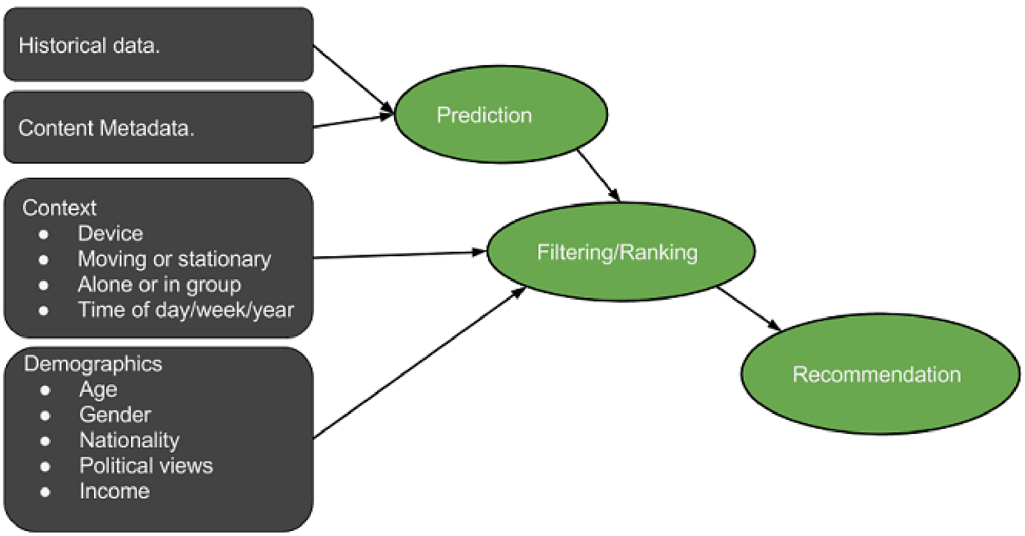

Figure 1.10 Illustrates that data can potentially be used as input data for a recommender system

This figure also illustrates another fact to take into consideration: the rating prediction is only a part of a recommendation system. Other things can also play an important role in what your system should actually display to the user. A big part of this book is about predicting ratings, it is very important, even if I made it sound like something negible here.

1.2 Taxonomy of recommender systems

Before starting to implement a recommendation system, it's a good idea to dwell a bit on what kind of recommender system you want to roll out of the garage. A good way to start is by looking at similar systems for inspiration. In the following, you will learn a framework for studying and defining a recommender system.

In the previous section, the tour d'Netflix provided an overview of what a recommender system can do. This section explains a taxonomy to use to analyze recommenders. I first learned about it in Professor Joseph A. Konstan and Michael D. Ekstrand's Coursera course "Introduction to Recommender Systems"[7], and have found good use for it ever since. It uses the following dimensions to describe a system: Domain, Purpose, Context, Personalization Level, Whose Opinions, Privacy and Trustworthiness, Interfaces, and Algorithms[8]

1.2.1 Domain

The *domain* is the type of content recommended. In the Netflix example, the domain is movies and TV series, but it can be anything: sequences of content such as playlists, best ways to take e-learning courses to achieve a goal, job listings, books, cars, groceries, holidays, destinations, or even people to date. The domain is significant because it provides hints on what you would do with the recommendations.

The domain is also important because it will indicate how bad it is to be wrong, if you are doing a music recommender then it is not really that bad if you recommend music which is not spot on. While if you are recommending foster parents to children in need, then the cost of failure is quite high.

The domain will also dictate if you can recommend the same thing more than once.

1.2.2 Purpose

What is the aim of the site, both for the end user and for the provider? For end-users, the use of Netflix recommendations is to find relevant content that they want to watch at that specific time. It might sound silly, but imagine that you didn't have any ordering or filtering. How would you ever find anything in the Netflix catalog when it has more than 10,000 items?

The purpose for the provider (in this case, Netflix) is ultimately to make customers pay for the subscription month after month by providing content they want to watch, right at their fingertips.

Netflix considers the amount of content viewed as a deciding factor in how they're doing. Often measuring something else instead of your direct goal is called using a proxy goal. Using a proxy goal is something you should be very careful about because it can inadvertently end

---

[7] https://www.coursera.org/learn/recommender-systems-introduction/
[8] The taxonomy first appeared in the book Word of Mouse by John Riel and Joseph A. Konstan.

up measuring other effects than what you wanted, more time spend on the Netflix platform could also mean very frustrated customers that searches and searches, or because the site keeps stalling while playing something[9]. Behind the scenes, there might also be considerations to balance things in such a way that Netflix pays the least money possible for what you are watching. Netflix probably pays less to offer 10-year-old episodes of *Friends* than a newer series or, even better, a Netflix original series; then they don't have to pay a license to anybody.

A purpose could also be to give information or to help or educate the user. In most cases, however, the purpose is probably to sell more.

What type of customers do you want to serve: consumers who arrive once and expect good recommendations, or loyal visitors who create profiles and return on a regular basis? Will the site be based on automatic consumption (for example, the Spotify radio station, which keeps playing music based on a song or an artist)?

1.2.3 Context

The *context* is the environment in which the consumer receives a recommendation. For example, Netflix delivers content on many platforms. The device the customer is using is the context. The context is also about the current location of the receiver, what time it is, and what the receiver is doing. Does the user have time to study the suggestions or is a quick decision needed? The context could also be the weather around the user or even the users mood! Consider a search for a café on Google Maps. Is the user sitting at a computer in the office and looking for a good coffee bar or is he standing on the street just as it starts to rain? In the first scenario, the best response would identify good-quality cafes in a bigger radius, whereas in the second scenario, recommendations would ideally contain only the nearest place to drink coffee while the rain passes. Foursquare is an example of an app where you can find cafes. We will look at Foursquare in chapter 12.

1.2.4 Personalization level

Recommendations can come at many levels of personalization, from using basic statistics to looking at individual user data. Figure 1.11 illustrates various levels of personalization.

---

[9] I recommend reading *Weapons of Math Destruction* by Cathy O'Neil if you want to know more about how wrong it can go when you use proxy goals.

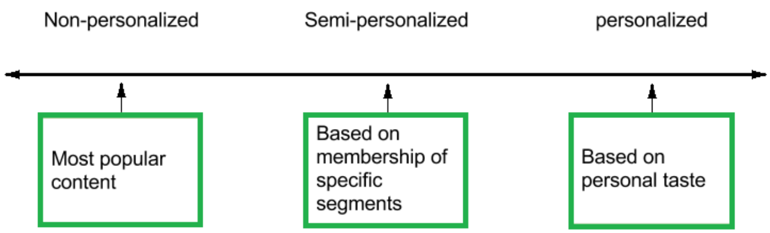

Figure 1.11 Personalization levels

NON-PERSONALIZED

A list of the most popular items is considered a non-personalized recommendation: the chances are that the current user might like the same items as most others do. Non-personalized recommendations also include showing things ordered by date, such as showing the newest items first. Everybody who interacts with the recommender system receives the same list of recommendations. Non-personalized recommendations also include when a café suggests drinks Friday afternoon, cappuccinos in the morning, but brunch on weekend mornings.

SEMI/SEGMENT-PERSONALIZED

The next level of personalized recommendations divides users into groups. You can segment groups of users in many ways: by age, by nationality, or by distinct patterns such as business people or students, car drivers or bicycle riders.

A system selling concert tickets, for example, would recommend shows based on the user's country or city. If a user is listening to music on a smartphone, the system might try to deduce whether that person is exercising, by using the GPS and seeing whether the device is moving. If it is stationary and at home, the consumer is probably sitting on a sofa and the appropriate music might be different. The recommender system doesn't know anything personal about you as a person, only as a member of a group. Other people who fit into the same group will get the same recommendations.

PERSONALIZED

A personalized recommendation is based on data about the current user that indicates how the user has interacted with the system before. This generates recommendations specifically for this user. Most recommender systems also use segments and popularity when creating personalized recommendations. Amazon's *Recommended For You* list is personalized. Netflix is an extreme example of personalized recommendations.

Usually, a site applies various types of recommendations. So far, only a few examples such as Netflix offer everything personalized. On Amazon, you will also find Most Sold Items, which is nonpersonalized, as well as the *Customers Who Bought This Also Bought This* list, which provides seeded recommendations. These are recommendations based on a *seed*, which is the current item.

1.2.5 Whose opinions

Expert recommenders are manual systems where experts recommend good wines, books or similar. These systems are used in areas where it's generally accepted that you need to be an expert to understand what is good. I would say that the days of expert websites are mostly over, and so the Whose Opinions parameter isn't used much nowadays. Almost all sites use the opinions of the masses.

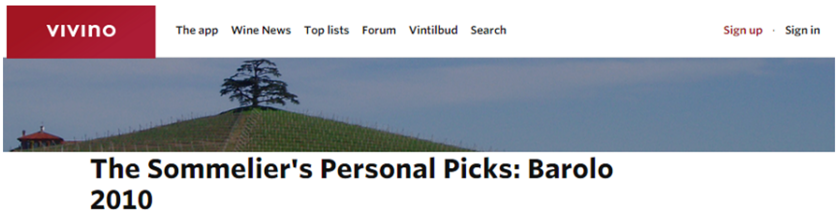

Figure 1.12 Vivino.com provides expert wine recommendations (the recommendations were cut away to save space).

They say that there's no rule without exception, and a few expert sites are still left. An example is the sommelier's recommendation on the wine site called vivino.com, shown in figure 1.12. In the first quarter of 2017, Vivino announced that they will add a recommender system to their app[10] to help users find new wines to taste based on their rating history.

1.2.6 Privacy and trustworthiness

How well does the system protect users' privacy? How is the collected information used? In Europe, it's common to pay money into a pension, which is handled by a bank. Often these banks offer different kinds of retirement savings plans. A system that recommends these should have strict rules for privacy. Imagine filling in an application for a retirement savings plan and describing that you have back problems, and a minute later receiving a phone call from a chiropractor with great offers to handle your exact problem. Or even worse, you buy a special bed for people with back problems, and an hour later, you receive an email that your health insurance premium has gone up.

Many people consider recommendations as a form of manipulation because they present choices that customers are more likely to pick than if they were offered just a random

[10] http://www.digitaltrends.com/home/vivino-market-wine-recommendations/

selection. And most shops are trying to sell more, so the fact that stores that use recommendations sell more makes people think that they are being manipulated. But if that means watching a film that would entertain rather than bore, then I would say it is okay. So, manipulation is more about the *motive* for showing a particular item rather than the *act* of showing it. If you're recommended inappropriate and non-optimal medicine because the vendor buys the website owner better dinners, then that is manipulation, which is to be frowned upon.

When the recommendation system starts performing, and an increase in business can be measured, many will find it tempting to inject vendor preferences, overstocked items, or maybe preferences for which brand of pills customers buy. However, beware: if customers start feeling manipulated, they'll stop trusting the recommendations and eventually find them somewhere else.

> ***The moment that recommendations have the power to influence decisions, they become a target for spammers, scammers, and other people with less-than-noble motives for influencing our decisions.***
>
> ***—Daniel Tunkelang***[11]

Trustworthiness indicates how much the consumer will trust recommendations instead of considering them as commercials or attempts at manipulation. In the Netflix example, I talked about how predictions can be discouraging for users if the estimated prediction is far off from the user's actually rating. This is about trustworthiness. If the user takes the suggestions seriously, the system is trustworthy.

1.2.7 Interface

The *interface* of a recommender system depicts the kind of input and output it produces. Let's have a look at each

INPUT

Netflix once enabled users to enter taste preferences by rating content and adding preferences on genres and topics, this can be used as input to a recommender system.

The example described with Netflix is an explicit input. Where you the consumer manually add information about what you like. Another form of input is implicit, meaning that the system tries to deduce taste by looking at how the user interacts with the system. Feedback will be handled in more detail in chapter 4.

[11] www.linkedin.com/pulse/taste-trust-daniel-tunkelang

OUTPUT

Netflix outputs recommendations in many ways. Netflix estimates predictions, provides personalized suggestions, and shows popular items (which normally is in the form of a top-10 list, but Netflix even personalizes that).

Types of output could be predictions, recommendations, or filtering. If the recommendations are a natural part of the page, it's called an *organic presentation*. The rows shown on Netflix are an example of organic recommendations; Netflix doesn't indicate that these are recommendations; they're just an integral part of the site.

Although the examples illustrated in figure 1.13 are *nonorganic,* Hot Network Questions explicitly states what it is (non-personalized recommendations). Amazon shows nonorganic personalized recommendations in its Recommended for You list, and the *New York Times* has nonorganic recommendations showing the most emailed articles.

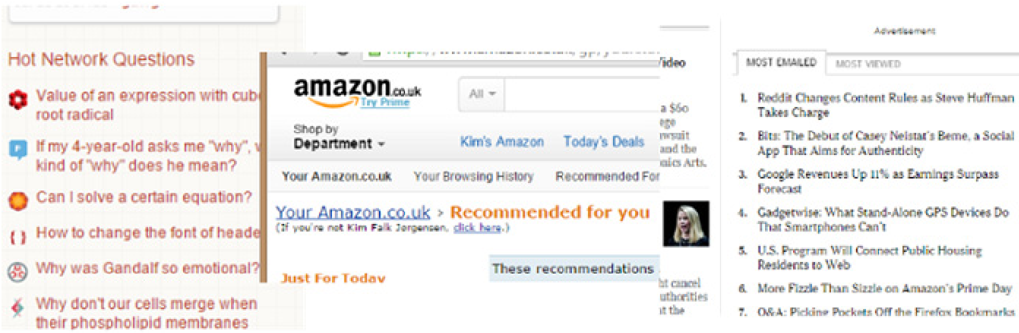

Figure 1.13 Examples of nonorganic, non-personalized recommendations:Hot Network Questions from Cross Validated, Recommended for You at Amazon and Most Emailed from the New York Times.

Some systems explain the recommendations. Recommenders with that ability are called *white-box* recommenders, and recommenders that don't are called *black-box* recommenders. Examples of each are shown in figure 1.14. This is important to consider when choosing an algorithm because not all of them provide a clear path back to the reasons for a prediction.

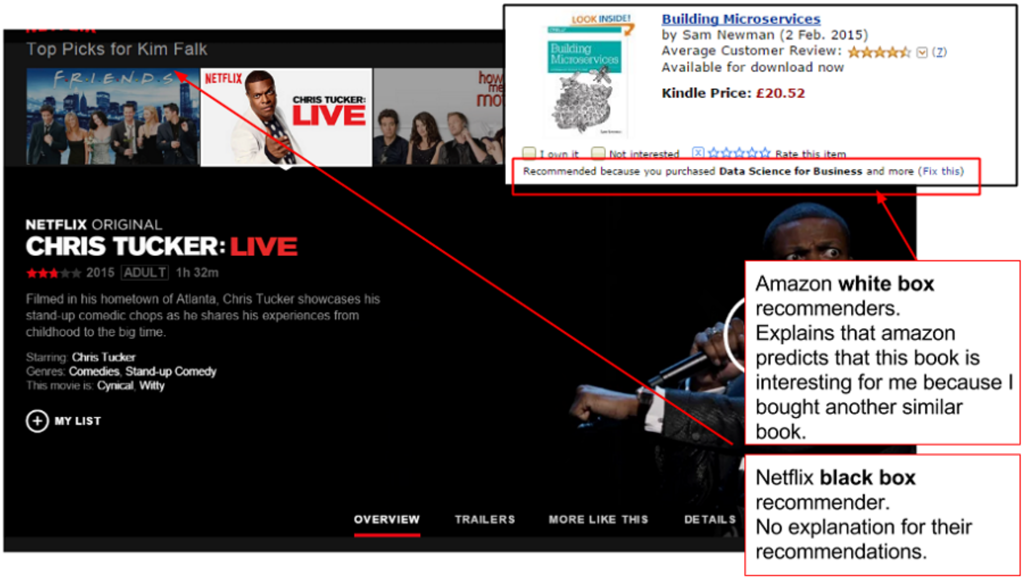

Figure 1.14 Black box and white box recommendations

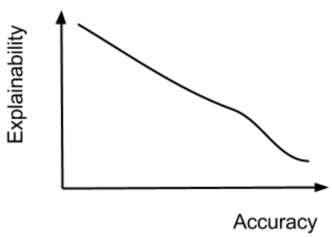

Figure 1.15 Explainability vs. quality of recommendations

Deciding whether you want to produce a white-box or black-box recommender is important because it can put constraints on which algorithms you can use. The more your system needs to explain, the simpler the algorithm will have to be. Often you can consider the decision as shown in figure 1.15. The better the quality of the recommendations is, the more complex and, therefore, the harder to show explanations. The problem is also known as *model accuracy - model interpretation trade-off*.

I have worked on a project requiring both explainability and quality to be optimum. To solve this, we had to build another algorithm on top of our recommender system to allow for the good-quality recommendations while also having a system that connected the evidence with the result.

Recommender systems have become extremely common in recent years, so there are many examples out there to look at. Often they are implemented in relation to movies, music, books, news, research articles, and products in general. But recommender systems also have a place in many other regions such as financial services, life insurance, online data, job searches, and in fact everywhere there are choices to be made. This book primarily uses websites as examples, but there is no reason for not working in other scenarios.

1.2.8 Algorithms

Various algorithms will be presented in this book. There are essentially two groups, and they depend on the type of data you use to make your recommendations. Algorithms that use usage data are called *collaborative filtering*. Algorithms that use content metadata and user profiles to calculate recommendations are called *content-based filtering*. The is a third type which can be a mix of the two types, called hybrid recommenders.

COLLABORATIVE FILTERING

Figure 1.16 shows a diagram illustrating one way of doing *collaborative filtering*. The outer set is the full catalog. The middle set is a group of users who have consumed similar items. A recommender system recommends items from the purple set, assuming that if users liked the same things like the current user, then the current user will also like other items this group has consumed. The group is identified by having an overlap between what the individual users have liked and what current user liked. Then the gap of content, which the current user is missing, will be recommended (the part of the middle circle that isn't covered by the circle representing the current user's likes).

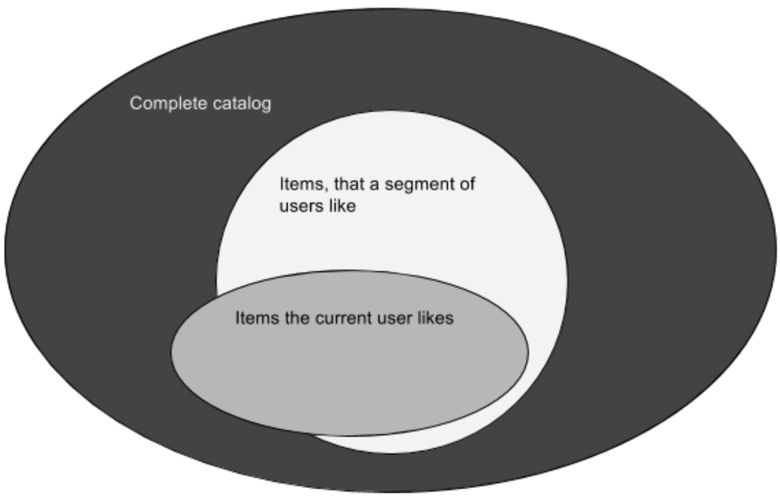

Figure 1.16 Collaborative filtering diagram

There are many ways to calculate collaborative filtering recommendations; you'll see a simple version in chapter 8 and again in chapter 11, where we talk about matrix factorization algorithms.

CONTENT-BASED FILTERING

Content-based filtering is about using the metadata you have on the items in your catalogue. In context of Netflix which is movies we would use descriptions of the movies. Depending on the specific algorithm used, the system can either calculate recommendations using either be done by taking the items the users has liked and find similar content to ,alternatively between the items and user profiles. Or simply between items. The system calculates a profile for each user, which contains categories of the content. If Netflix used content-based filtering, it would create a user profile comprising genres such as thrillers, comedies, drama, and new films, and give values to them all. Then a film gets recommended if it has similar values as the user. Here's an example:

User Thomas likes *Guardians of the Galaxy*, *Interstellar*, and *Game of Thrones*. Each film is rated according to a five-point system. The following could be a way of looking at the three films:

	Sci-Fi	Adventure
Interstellar	3	3
Game of Thrones	1	5
Guardians of the Galaxy	5	4

Based on this information, you build a profile of Thomas indicating Sci-Fi: 3, Adventure: 4. To find other films to recommend, you look through the catalog to find films similar to Thomas' profile.

HYBRID RECOMMENDER

Both collaborative filtering and content-based filtering have strengths and weaknesses, collaborative filtering needs to have a lot of feedback from the users to work probably, while the content-based filtering needs good descriptions of the items, to work. Often recommendations are produced as a mix of the output from the two types of algorithms we talk about above, plus other types of input, which could be as distance from the place, time of day.

1.3 Machine learning and the Netflix Prize

A recommender system is about predicting what content a user needs right now. There are many ways to predict this. Building recommender systems has become a multidisciplinary sport that takes advantage of computer science fields such as machine learning, data mining and information retrieval, and even human-computer interaction. Machine-learning and data-mining methods enable the computer to learn to predictions by studying examples of what it should predict. Consequently, recommendations can be constructed by using these prediction functions. Many recommender systems are centered around various machine-learning algorithms to predict user ratings of items, or to learn how to correctly rank items for a user.

One reason that the field of machine learning is growing is that people are trying to solve the recommender system problem. The aim is to implement algorithms that will enable computers to suggest our secret wishes, even before we know them ourselves.

Many will claim that the catalyst for this interest in applying machine learning to recommender systems was the famous Netflix Prize. The Netflix Prize was a competition hosted by Netflix that offered one million dollars to anyone who could come up with an algorithm that improve their recommendations by 10%. The competition began in 2006, and it took almost three years for somebody to win it. Many people, universities, and companies tried to do it. In the end, it was a hybrid algorithm that won. A hybrid algorithm runs several algorithms and then returns a combined result from all of them. (we will talk about hybrids in chapter 11).

Netflix never used the winning algorithm. The biggest reason was that it was such a complicated algorithm that the performance hit on the system could not justify the improvements.

Sadly, we don't have Netflix to play with while learning about recommender systems. Instead, I have implemented a small demo site called MovieGEEKs to show off the things described in the book. The site would require a lot of tweaking before it would be production ready. Understanding recommender systems is its key purpose.

1.4 The Movie GEEKs website

This book is about how to implement recommender systems. It will provide you with the tools to do that, no matter which platform you want to use for your recommendation system.

However, to do anything interesting with a recommender system you need data and to get a feel for how it is working its not enough just to look at numbers. This book focuses on websites, but that doesn’t mean that everything written here doesn’t apply to any other type of system. The following is a short introduction to the framework in which we’ll do our dance.

We will use a website built using a Django Website. And I encourage you to download and use the site as you read through the book, as it will help you understand what is going on. The fact that it is a Django site or something else is not so important; I will point you to where to look.

Django website and framework

If the words *Django webframework* sounds strange to you, please have a look at the Django documentation at https://www.djangoproject.com/start/overview/.

The website is downloaded once and will contain all the functionality described in this book. The scenario we will pretend we are doing is as follows:

Imagine that you have a customer who wants to take his DVD selling online. I imagine an old DVD rental shop that’s in Bath, in the United Kingdom, with an owner who wishes to try movie selling on the internet.

(The store sadly doesn’t exist any longer.[12] The shop was anything but electronic; it was all managed with small paper cards, and although you might think that sounds impossible, it all seemed to work fine. In real life, I don’t think the owner would ever have taken his business online, but one of the unique things about this place is that you’d always get superb recommendations; the owner would do a monthly review—expert opinion recommendations—and the people who worked there knew everything there is to know about films.) I like to think of recommender systems as an attempt to give personal service to people on the net. The following is a brief description of what the fictive owner wants.

1.4.1 Design and specification

To get started on the project, you’ll need to put down some overall points for the design.

The main page of the site should show visitors the following:

- A tiled area of movies
- An overview of each film, without leaving the page
- Recommendations as personal as possible
- A menu list containing the genres

Each movie should have its own page containing details as follows:

- Movie poster
- Description
- Rating

Each category should have a page containing the following:

- Same structure as the front page
- Recommendations specific to the category

[12] A sad day for movies as Bath rental shop On the Video Front shuts: www.bathchronicle.co.uk/sad-day-movies-Bath-rental-shop-Video-shuts/story-17883596-detail/story.html

1.4.2 Architecture

You'll use Python and the Django web framework to implement this site. Django enables you to split a project into different applications. Figure 1.17 shows a high-level architecture and provides an illustration of which applications will build the site.

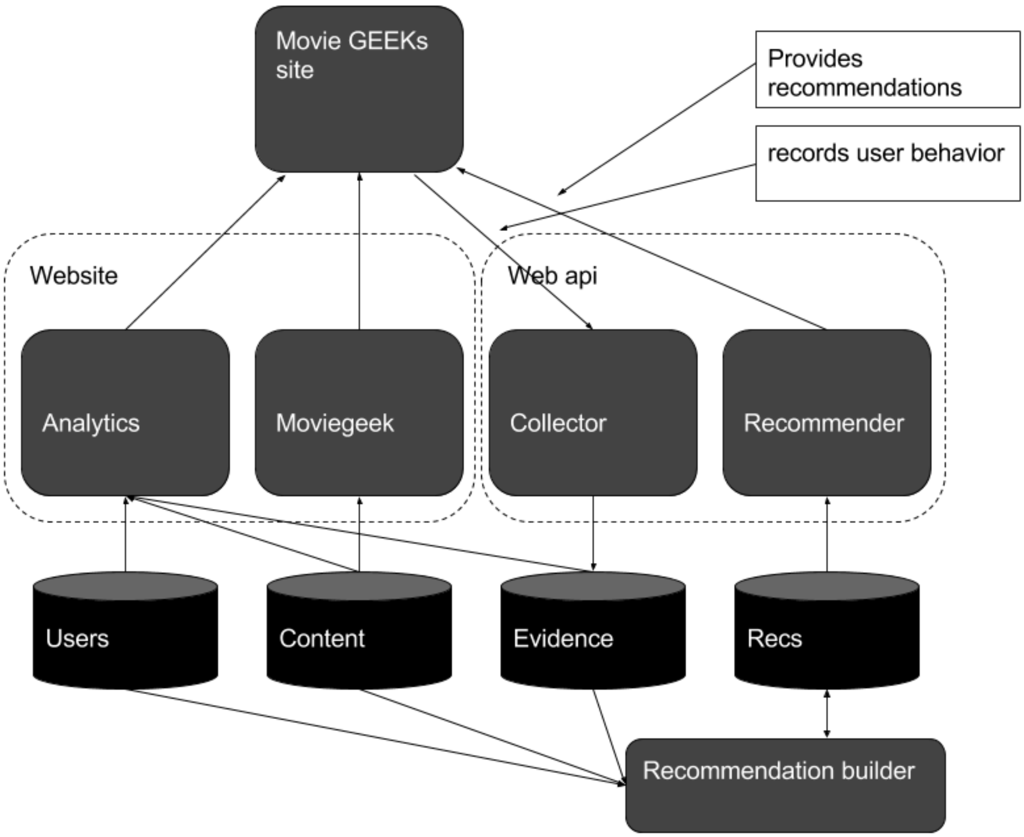

Figure 1.17 MovieGEEKs architecture

Let's do a quick walk-through:

- *Moviegeek*—This is the main part of the site. Here the client logic (HTML, CSS, JavaScript) is placed along with the Python code responsible for retrieving the movie data.
- Analytics —The ship's bridge, where everything can be monitored. This part will use data from all the databases. The analytics part is described in chapter 4.
- Collector—This handles the tracking of the user behavior, and stores it in the evidence

database. The evidence logger is described in chapter 2.

- Recs—This is the heart of this story, and is what will add the edge to this site. It will deliver the recommendations to the MovieGEEKs site. The recs part is described in chapter 5 and the rest of the book.
- Recommendation builder—This will pre-calculate recommendations, which the recs will use to provide elaborate recommendations to the user. You will meet the recommendation builder for the first time in chapter 7.

Each of these components/applications will have exciting data models and features that hopefully will blow future visitors away. But not to spoil everything now; please find details in each of the respective chapters.

The website is called Movie GEEKs and is a movie site mainly for the single reason that a dataset is available that contains a long list of content—movies, users, and ratings. Even more important, the content includes URLs that translate into movie posters, which makes working with it much more fun.

Let’s look at a screenshot. Figure 1.18 shows the front page, which shows the landing page. When the user clicks a movie, a pop-up appears, giving more information and a link to even more details.

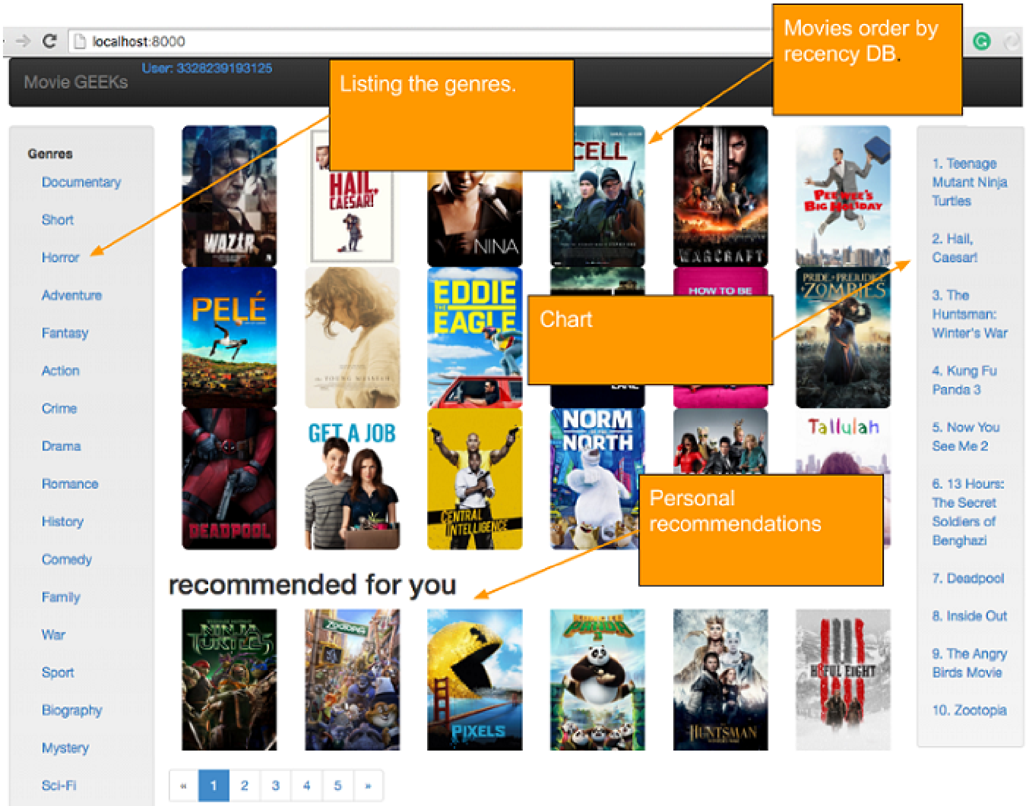

Figure 1.18 The loading page of the MovieGEEKs site

That's it! Simple but it will do the trick. Go ahead and download it now. Please refer to the readme on git hub for installation instructions.

https://github.com/practical-recommender-systems/moviegeek

The MovieGEEK site uses a dataset called MovieTweetings, it is a dataset consisting of ratings on movies that were contained in well-structured tweets on Twitter[13].

1.5 Building a recommender system

Before moving on, let's have a quick look at how you would go about building a recommender system. Assuming you already have a platform in the shape of a website or some app where

[13] https://github.com/sidooms/MovieTweetings

you want to add a recommender system, it would go something like the cycle shown in Figure 1.19

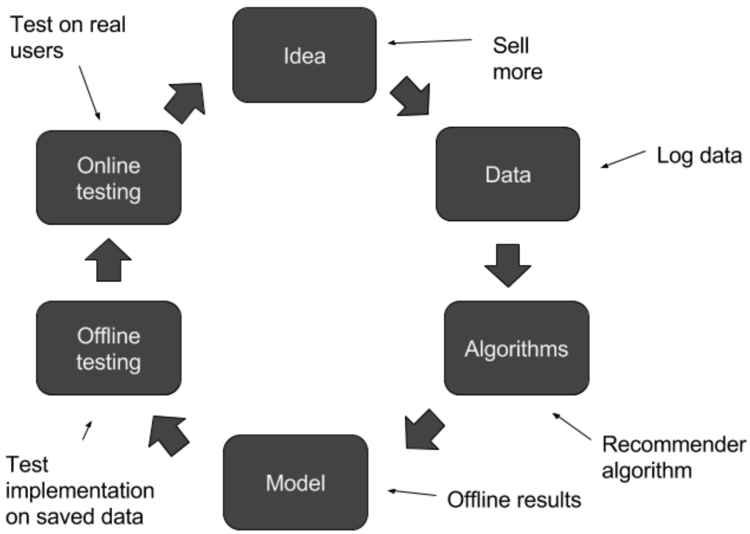

Figure 1.19 The data driven approach

Start out with an idea that you want to sell more by adding recommenders. You then (or hopefully you already have been) collect behavioral data, and use that data to build an algorithm, which creates some kind of model when it is run. The model can also be considered as a function, which will, given a user id, calculate recommendations.

With this model, you'd try it out on some historical data simply to see if you can use your model to predict some behavior of a user. For example, if you have data showing what users bought last month, then you can try to create the model using the three first weeks of data and see how well the model recommends things that the users bought in the last week of your month of data. If it is better at predicting what the users has bought compared to some baseline recommender system, which can be as simple as a method that returns the most popular items. If it is doing well, you can expose it to some of your users and see if you can track improvement. If yes, then it can go into production; otherwise, it's back to the idea.

1.6 Summary

The first chapter is finished, and you should now have an idea of what a recommender system is, and an understanding of what's needed as input and what can be produced. To summarize, you should now understand the following:

- Netflix uses recommendations to personalize its site, and help users selected things the like.Recommender system is a common term for many different components and methods. A prediction is different from a recommendation. A prediction is about predicting what rating a user would give some content, while a recommendation is a list of items that is relevant to the user.

- A recommendation context is what is happening around the user when a recommendation arrives. items that might not be predicted to have the highest ratings can be recommended if they suit the context. The taxonomy described in the chapter is handy when you are looking at other recommender systems or trying to design your own. It is good to go through this taxonomy before starting to implement your own recommender system.

Knowing the basis of a recommender system gives you the foundation for chapter 2, which shows how to collect data from users.