SWIPE TO UNLOCK

A Primer on Technology and Business Strategy

Neel Mehta
Parth Detroja
Aditya Agashe

Swipe to Unlock

Published by Belle Applications, Inc.
swipetounlock.com

2nd edition, 2st revision (September 2018)

ISBN-13: 978-1976182198
ISBN-10: 1976182190

To my friends and family, for supporting me no matter how crazy my dreams get — Neel

To my friends and family for their never-ending support, and Professor Deborah Streeter for believing in my vision, which helped make this book possible — Parth

To my family and friends, thanks for supporting my passion for business and helping me push past my fears to embrace entrepreneurship — Adi

When you grow up, you tend to get told that the world is the way it is and your life is just to live your life inside the world, try not to bash into the walls too much, try to have a nice family, have fun, save a little money. That's a very limited life. Life can be much broader, once you discover one simple fact, and that is that everything around you that you call life was made up by people that were no smarter than you. And you can change it, you can influence it, you can build your own things that other people will use… That's maybe the most important thing. It's to shake off this erroneous notion that life is there and you're just gonna live in it, versus embrace it, change it, improve it, make your mark upon it… Once you learn that, you'll never be the same again.

— Steve Jobs
(who, by the way, never wrote any code for Apple)

Contents

Chapter 1:
Introduction

Understanding technology is essential in today's world, no matter your career. Doctors are using artificial intelligence to diagnose patients.[1] Finance, payments, and banking have all been moving online.[2] Businesspeople are adapting to a world where, as of 2018, the world's five biggest companies are all tech companies.[3] Even farmers are using drones to grow better crops.[4] Whether you're a doctor, banker, farmer, or most anything else, you need to know how technology works.

If you don't know how to code, you might be a little worried right now.

But here's the thing: you don't need to know how to code to understand technology. We believe that all the most important technology topics — from how the internet works to how apps like Snapchat make money — can be explained in plain, simple English. You'll find explanations of these topics, and many more, in *Swipe to Unlock*.

In this book, we've distilled all the big ideas we've learned from working around the tech industry, from tiny startups to massive corporations. Inside *Swipe to Unlock*, you'll find breakdowns of the hottest technologies, translations of the

most mysterious tech buzzwords, and analyses of the business strategies that power the tech industry. We think it's everything you need to know about technology and the business strategy behind it, and we think absolutely anyone can learn it.

But enough about us. How about you?

Who this book is for

The goal of *Swipe to Unlock* isn't to teach you how to code. It's to teach you how to think like a technologist. Whenever you come across a technology topic at work or on the news, you'll be able to think through how the technology works, why it was made that way, how it makes money, and how it might help or harm people. While specific apps and companies are always coming and going, the core concepts you'll pick up in *Swipe to Unlock* should prove useful for a long time to come.

If you want a business, marketing, product management, or other non-engineering role at a tech company, you need to understand the fundamentals of technology. In *Swipe to Unlock*, we'll teach you how to think like a software engineer. You'll learn how software is made, how the internet works, and what buzzwords like "big data," "machine learning," and "APIs" mean. (Don't worry if you've never heard of these terms — we'll break them down throughout the book!)

If you're already a software developer but want to move more toward product management roles, we'll teach you the business strategy insights you need to know. We'll pose a variety of real-world business questions and work through the answers. *Why does Amazon offer free shipping with Prime if it loses them money? Why did Facebook acquire Instagram? How do software companies make money when most apps are free?* You'll also learn how to break down complex technical concepts to laypeople. Questions like "explain cloud computing to a five-year-old" are surprisingly common in tech interviews — so we'll show you some useful analogies, like how cloud computing is similar to calling an Uber.

If you work outside the tech industry, you'll learn about the digital tools you need to stay ahead of the curve. We'll teach you what jargon like "predictive analytics" and "the cloud" means, and we'll introduce you to companies that have been using these technologies to boost sales, cut costs, and improve efficiency.

Even if you don't need to know any technology for your career, you still use it every day — you probably have a very advanced piece of technology in your pocket (or your hand) right now. *Swipe to Unlock* will make you a better-educated consumer by answering some of the questions you might have had while following the news or using your favorite apps. *What caused the fake news epidemic on social media? What's this "net neutrality" debate that everyone's talking about? How much of my personal data does Google own?* We'll talk about these questions and many others.

No matter why you're reading this book, we think you'll find plenty of valuable insights and ideas, and you'll learn how to think — and speak! — like a technologist.

But before we start, let's dive a bit deeper into the topics you'll learn about.

What's inside

We'll kick off *Swipe to Unlock* by looking under the hood of the technology you use every day, from your phone's operating system (like Android or iOS) to your favorite apps. In the middle chapters, we'll break down important concepts like "the cloud," "big data," and hacking. Then we'll zoom out and look at some of the business, legal, and political implications of technology. We'll finish up with a look at emerging trends in technology and some predictions for the future.

All three of us hate reading textbooks, so we didn't write *Swipe to Unlock* like a textbook. Instead, every single section is in the form of a question, where we step through real-world examples. *How does Spotify recommend songs to you? How do self-driving cars work? How could a hacker hold your computer for "ransom"?* We'll break down many cases like these.

Along the way, you'll read about some surprising, and maybe even funny, stories. *How did Target figure out that a teenager was pregnant before her own father did? How did a single typo take down 20% of the internet?* We'll even meet some interesting characters, like a Wall Street trader who blasted right through the Allegheny Mountains in his quest to make a perfectly straight internet cable.

A final note: we couldn't possibly cover everything in full detail in *Swipe to Unlock*, so we'll give you dozens — and, in some cases, hundreds — of citations per chapter. You can use these links to dive deeper into any topic that interests you.

For job hunters

If you're looking for a non-developer role at a tech company, we're glad you're reading *Swipe to Unlock* — we think it'll be useful! A big reason we wrote this book was because there just weren't any comprehensive resources for non-experts to learn the basics of technology.

Swipe to Unlock aims to teach you the technology and business concepts you'll need to know, and it'll help you understand tech companies' favorite jargon. We think the technical topics will be especially useful if you're coming from a non-technical background, and the business concepts will especially help if you're coming from an engineering background.

Bear in mind, though, that *Swipe to Unlock* isn't designed to give you resume or networking advice, and it won't teach you how to code or prepare for interviews. For that kind of information, check out the Resources section of our website at swipetounlock.com/resources, where we share links to some useful books and articles.

The questions we explore in *Swipe to Unlock* aren't designed to resemble actual interview questions; instead, they'll give you the technical and business insight to craft answers that'll set you apart from the pack. Let us explain with a couple of examples.

One example: we'll talk about how Google's algorithm decides which ads to show to users. Google probably wouldn't ask you to explain this algorithm at an interview, but they could ask you how you would increase ad revenue from a particular group of

users. Knowing how the targeting algorithm works is crucial to creating a strong answer.

A second example from the business world: we'll discuss why Microsoft acquired LinkedIn. Microsoft interviewers probably won't ask you to explain the reasons for the purchase, but they could ask you how you'd improve Microsoft Office for businesses. In your answer, you could draw on our discussion of the purchase, where we discuss how Microsoft and LinkedIn data could work together. You could mention one interesting idea we'll cover: analyzing a company's current employees via LinkedIn to see what kinds of talent they need to add. We think answers like these can help you stand out from the crowd.

Third, we'll give you some great anecdotes to take to the interview. For instance, we'll introduce you to Robinhood, an app that lets you trade stocks without paying brokers any commission. Robinhood has a clever way of making money: it earns interest on the unspent money sitting in users' accounts. At an interview, what if someone asks you to come up with a new money-making strategy for Venmo, an app that lets you send money to friends for free? Well, you could pull out the Robinhood story and propose that Venmo earn interest on money sitting in Venmo users' accounts. We think our stories will help you find many creative ideas like this.

In short, *Swipe to Unlock* is an essential complement to books full of sample interview questions. Sample questions alone won't help you understand the tech industry, think strategically about software products, or become fluent in technology jargon, but we believe *Swipe to Unlock* will. Best of luck!

Who we are

Before we jump into the main content of *Swipe to Unlock*, let us introduce ourselves. The three of us met while working at Microsoft. We quickly realized that Silicon Valley does a poor job making the world of technology accessible to non-experts. We believe that everyone can — and should! — understand the fundamentals of technology, and we're passionate about helping make that happen. That's why we wrote *Swipe to Unlock*.

Here's a bit more about us.

Neel Mehta is a Product Manager at Google. He has previously worked at Microsoft and graduated *cum laude* from Harvard University.

Parth Detroja is a Product Manager at Facebook. He has previously worked at Microsoft, Amazon, and IBM. He graduated *summa cum laude* from Cornell University.

Aditya Agashe is a Product Manager at Microsoft. Previously, he was the founder and CEO of Belle Applications. He graduated *cum laude* from Cornell University.

Thank you and good luck!

Whatever your personal or career goals, we hope *Swipe to Unlock* helps you. If you'd like more tips along the way, learn more and join our mailing list at swipetounlock.com. We regularly release new interviews, articles, and posts to help non-developers understand the world of technology.

Thank you again for choosing to read *Swipe to Unlock*, and we hope you enjoy!

Neel Mehta
hathix.com
linkedin.com/in/neelmehta18

Parth Detroja
parthdetroja.com
linkedin.com/in/parthdetroja

Aditya Agashe
whoisadi.com
linkedin.com/in/adityaagashe

Chapter 2:
Software Development

Let's start our tour of the world of technology by taking a closer look at the apps you use every day.

Every piece of software, from the original Pac-Man game to the latest version of Snapchat, is made of instructions, or code, that people wrote down and that computers follow. While the code behind apps like Spotify or Yelp might seem like magical spells, many of the core ideas are simple. We won't get into the nitty-gritty of code in this chapter, but we'll take a high-level overview of three major building blocks of apps and websites.

First, software is built around algorithms, or well-defined procedures that computers use to solve a problem. For instance, the algorithm to find the area of a triangle is to multiply the base and the height, then cut the result in half. There are algorithms to predict the weather, recommend movies to watch, process credit card payments, and more. We'll take a look under the hood of some famous apps and websites to reveal the algorithms that make them tick.

Then we'll turn to a second building block: APIs, or application programming interfaces, which let apps borrow algorithms and data from other apps. We'll explore how APIs

work and why developers use them, and we'll learn how to analyze an app to figure out what APIs it uses.

Algorithms and APIs help developers build out software, but to perfect it, developers turn to a third tool: A/B tests. In an A/B test, you show different versions of a feature to different groups of users: for instance, half the users get a red button while the other half get a blue button. Then you look at certain statistics, or "metrics," to see which variation performed better, and you push the winning version to all users. We'll finish up this chapter with a look at how some apps and websites use this powerful, scientific technique to improve their software.

Once you understand these three building blocks, you can start peering under the hood of any app or website to learn how it works. Let's take a look.

How does Google search work?

Whenever you search on Google, the search engine combs through the over 30 trillion webpages on the internet and finds the top 10 results for your question.[1] 92% of the time, you'll click on a result on the first page (that is, among the top 10 results).[2] Finding the top 10 things out of 30 trillion is really hard — it's about as hard as trying to find a penny randomly dropped somewhere in New York City.[3] Yet Google does this expertly and in just half a second, on average.[4] But how?

Google doesn't actually visit every page on the internet every time you ask it something. Google actually stores information about webpages in databases (tables of information, like in Excel), and it uses algorithms that read those databases to decide what to show you. Let's dive into how these work.

First, Google needs to build a database of every webpage on the internet. Google uses programs called spiders to "crawl" over webpages until it's found all of them (or at least, what Google thinks is all of them). The spiders start on a few webpages and add those to Google's list of pages, called the "index." Then, the spiders follow all the outgoing links on those pages and find a new set of pages, which they add to the index. Next, they follow all the links on *those* pages, and so on, until Google can't find anything else.

This process is always ongoing; Google is always adding new pages to its index or updating pages when they change. The index is huge, weighing in at over 100 million gigabytes.[5] If you tried to fit that on one-terabyte external hard drives, you'd need

100,000 — which, if you stacked them up, would be around a mile high.[6]

Word search

When you search Google, it grabs your query (the text you typed into the search bar) and looks through its index to find the webpages that are most relevant.

How might Google do this? The simplest way would be to just look for occurrences of a particular keyword, kind of like hitting Ctrl+F or Cmd+F to search a giant Word document. Indeed, this is how search engines in the 90's used to work: they'd search for your query in their index and show the pages that had the most matches,[7] an attribute called keyword density.[8]

This turned out to be pretty easy to game. If you searched for the candy bar Snickers, you'd imagine that snickers.com would be the first hit. But if a search engine just counted the number of times the word "snickers" appeared on a webpage, anyone could make a random page that just said "snickers snickers snickers snickers" (and so on) and jump to the top of the search results. Clearly, that's not very useful.

PageRank

Instead of keyword density, Google's core innovation is an algorithm called PageRank, which its founders Larry Page and Sergey Brin created for their PhD thesis in 1998.[9] Page and Brin noticed that you can estimate a webpage's importance by looking at which other important pages link to them.[10] It's like how, at a party, you know someone is popular when they're

surrounded by *other* popular people. PageRank gives each webpage a score that's based on the PageRank scores of every other page that links to that page.[11] (The scores of *those* pages depend on the pages that link to them, and so on… this stops eventually thanks to some math tricks from linear algebra.[12])

For instance, if we made a brand-new webpage about Abraham Lincoln, it would initially have very low PageRank. If some obscure blog added a link to our page, our page would get a small boost to its PageRank. PageRank cares more about the quality of incoming links rather than the quantity,[13] so even if dozens of obscure blogs linked to our page, we wouldn't gain much. But if a New York Times article (which would probably have a high PageRank) linked to our page, our page would get a huge PageRank boost.

Once Google finds all the pages in its index that mention your search query, it ranks them using several criteria, including PageRank.[14] Google considers many smaller criteria as well: it considers how recently a webpage was updated, ignores websites that look spammy (like the "snickers snickers snickers snickers" site we mentioned earlier), considers your location (it could return the NFL if you search for "football" in the US, but the English Premier League if you search for "football" in England), and more.[15]

One particularly interesting technique is called "query rewriting," where Google looks for words with similar meanings to the word you searched for. For instance, if you searched for "sofa", Google could also return webpages that include terms like "couch," "chair," "seat," or "furniture."[16]

Gaming Google?

There are pitfalls to PageRank, however. Much like spammers abused keyword density (as with "snickers snickers snickers snickers"), spammers have now started making "link farms," or webpages that contain tons of unrelated links. Website owners can pay link farms to include a link to their webpages, which would artificially boost their PageRank.[17] However, Google has gotten pretty good at catching and ignoring link farms.[18]

There are some more mainstream ways to game Google, though. An entire industry, called search engine optimization (SEO), has sprung up to help website owners crack Google's search algorithm and make sure their webpages appear at the top of Google searches.[19] The most basic form of SEO is getting more pages to link to your page. SEO includes plenty more techniques, such as putting the right keywords in your page's title and headings or making all of your site's pages link to each other.[20]

Constant changes

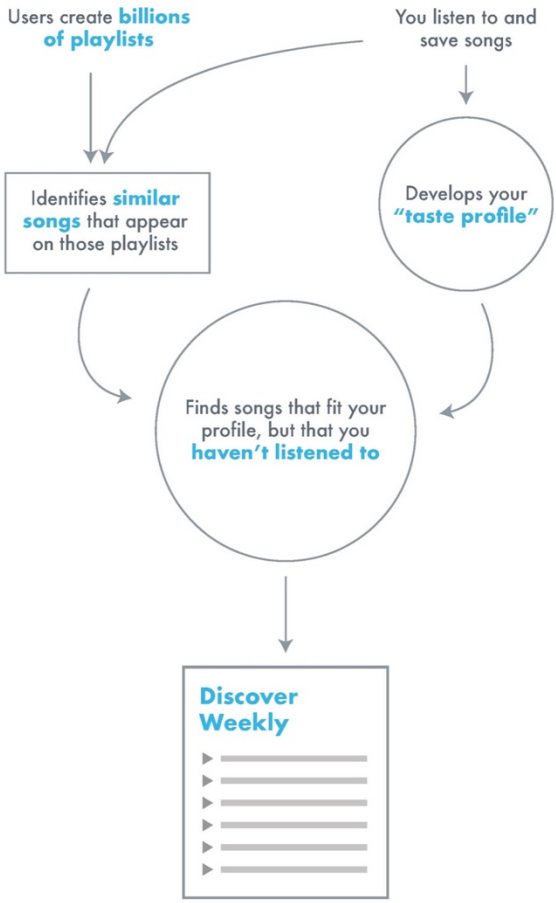
One final thing to note: while PageRank has always remained at the core of Google, the search engine is always changing. Google made 1,600 changes to its search algorithm in 2016 alone,[21] and in recent years it's added features like driving directions and "infoboxes," or cards that show quick information about your search term from Wikipedia.

Infoboxes are a new feature of Google's search results. They grab relevant information about your search from sources like Wikipedia. Source: Google search

How does Spotify recommend songs to you?

Every Monday morning, Spotify sends its listeners a playlist of 30 songs that are, almost magically, perfectly tuned to their tastes. This playlist, called Discover Weekly, has been a hit: within six months of launching in June 2015, it was streamed over 1.7 billion times.[22] But how does Spotify get to know its 75 million users so intimately?

Spotify does employ music experts who hand-curate public playlists,[23] but there's no way they could do that for all 75 million Spotify users. Instead, Spotify has turned to a computer algorithm they run every week.[24]



Spotify's algorithm for automatically recommending songs for you. Source: Quartz[25]

The Discover Weekly algorithm starts by looking at two basic pieces of information. First, it looks at all the songs you've listened to and liked enough to add to your library or playlists. It's even smart enough to know that if you skipped a song within the first 30 seconds, you probably didn't like it! Second, it looks at all the playlists that other people have made, with

the assumption that every playlist has some thematic connection; for instance, you could have a "running" playlist or a "Beatles jam" playlist.[26]

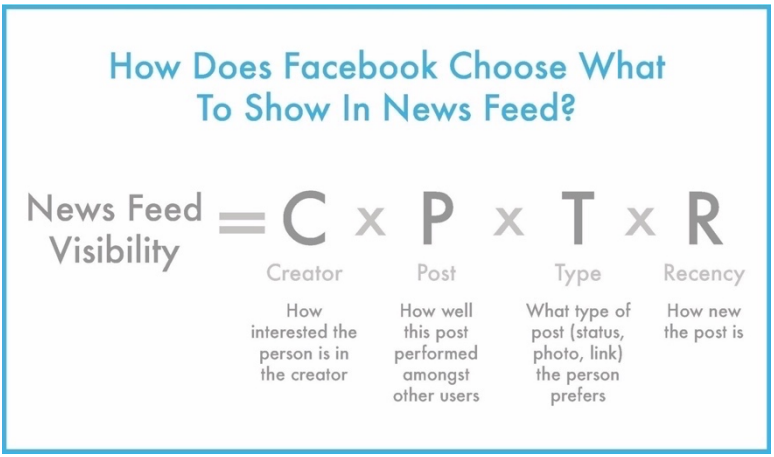
Once it has this data, Spotify uses two methods to find songs you might like. The first method involves comparing the two datasets to figure out which new songs are related to the ones you like. For instance, suppose someone made a playlist with eight songs on it, and seven of those are in your library. You probably like this type of song, so Discover Weekly might recommend the one song that isn't in your library.[27] This technique is called "collaborative filtering," and it's the same technique that Amazon uses to suggest items you might like based on your purchase history and the purchases of millions of other users.[28]

The second method that Spotify uses to make your playlist is your "taste profile." Based on just the songs you've listened to and liked, Spotify will determine which genres (e.g. indie rock or R&B) and micro-genres (e.g. chamber pop or New Americana) you like and recommend songs from those genres. It's just a different form of their strategy for recommending songs based on past listening patterns.[29]

Recommendation systems are getting more and more popular across the app landscape: besides Spotify, Netflix suggests movies, YouTube suggests videos, Facebook suggests friends, and so on. Once you understand how Spotify runs its recommendation system, the others all become easier to understand, because they largely use the same methods, like collaborative filtering.[30]

How does Facebook decide what shows up in your news feed?

Over a billion people look at their Facebook news feed every day, and Americans spend almost as much time on Facebook as they do interacting face-to-face with other people.[31] Because it draws so many eyeballs, the news feed has tremendous power. The news feed can influence our mood, put us in an ideological echo chambers,[32] or even influence who we're going to vote for.[33] In short, what appears in your news feed matters. So how does Facebook decide what to show in your news feed?



A simplified explanation of Facebook's news feed algorithm. Source: TechCrunch[34]

More specifically, how does Facebook take the hundreds (or thousands) of updates you get every day and sort them for you? Like Google, it uses an algorithm to figure out what's most important. There are about 100,000 personalized factors, but we'll focus on four key ones.[35]

The first factor is who posted it. Facebook will show you more posts from people you've interacted with more (e.g. people you've messaged more or tagged more), with the assumption that you're more likely to engage with their future posts.[36]

Second is the post's quality. The more people have engaged (e.g. liked or commented) with a post, the more interesting Facebook thinks it is, so the more likely it is to appear at the top of your news feed.[37]

Third, the type of post. Facebook figures out what kinds of posts (videos, articles, photos, etc.) you interact with a lot and shows you more of those.[38]

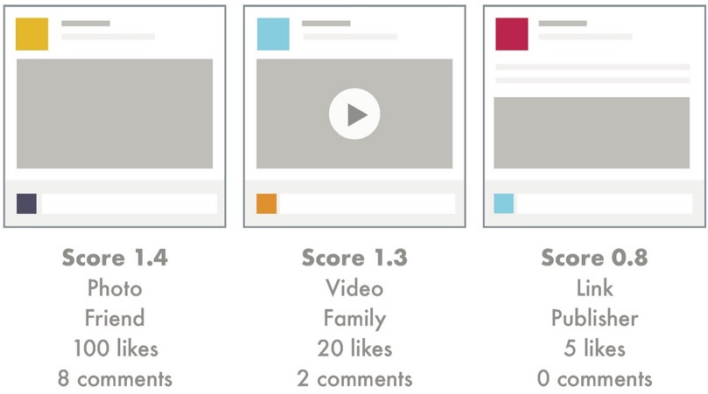
The fourth major factor is "recency": newer stories get ranked higher.[39]

There are plenty more factors, though. TIME found a few:

> Use a phone with a slow mobile connection and you may see less video. Writing 'congratulations' in a comment signals the post is probably about a big life event, so it will get a boost. Liking an article after you clicked it is a stronger positive signal than liking before, since it means you probably read the piece and enjoyed it.[40]

As you can tell, Facebook really tries to maximize the probability that you're going to like or comment on the posts in your news feed, a metric called engagement. After all, the more you like your news feed, the more you're going to scroll

down, and the more you scroll down, the more ads you'll see. Ads, of course, are how Facebook makes most of its money.[41]



An example of how Facebook ranks posts and determines what appears on your news feed. Source: TechCrunch[42]

Fighting fake news

Algorithms like Facebook's news feed algorithm are incredibly powerful, but the danger is that they're still easy for crafty hackers to game. With no human oversight, the algorithms could be turned against us.

A famous example is the fake news epidemic that swept Facebook during the 2016 American presidential election.[43] Recall that the news feed algorithm doesn't consider how true or reputable a post is; it only cares about maximizing engagement.[44] Fake newsmongers took advantage of this to attack politicians they didn't like, posting outrageous and demonstrably false news articles around Facebook. These articles naturally drew many clicks and comments, so

Facebook's news feed algorithm propelled them to the top of many people's news feeds.[45]

How could we fix weaknesses like this in the news feed algorithm? We might need to bring back a human touch to the algorithm — which is a bit ironic because algorithms are designed to reduce the amount of work humans have to do in the first place. For instance, Facebook and Google introduced features that let people flag fake news posts, which their updated algorithms can then hide.[46]

Even outside the battle against fake news, Facebook has been bringing in humans to improve the news feed algorithm, because counting likes can only tell you so much about how people feel about posts. In 2015, Facebook started hiring focus groups that scroll through their news feeds and give feedback to the people behind Facebook's news feed algorithm.[47] (Yes, that's right, you *can* be a professional Facebook tester, but we recommend staying in school regardless.)

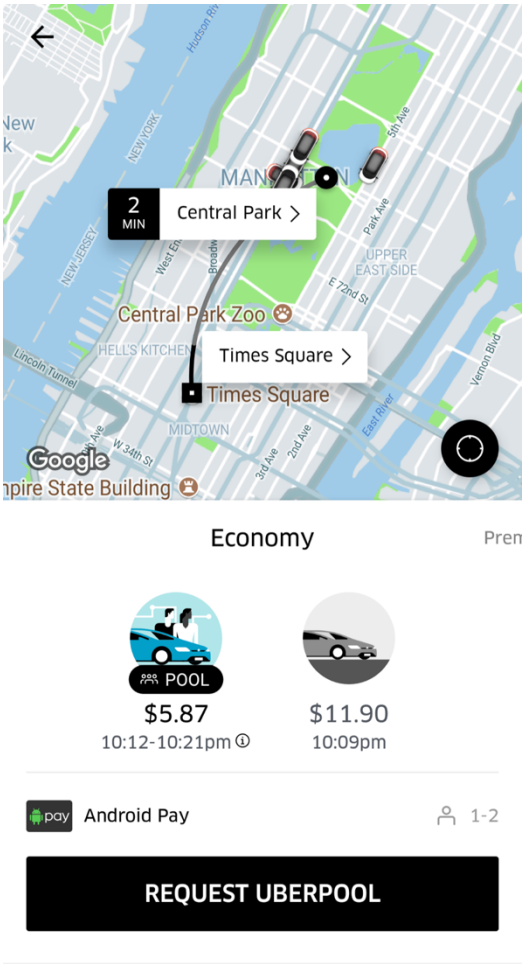
Algorithms aren't magic spells that run the world. They're just sets of rules (though complex ones) that other people wrote to make computers do a particular task. And, as Facebook shows, sometimes machines and people need to work together.

What technologies do Uber, Yelp, and Pokémon Go all have in common?

Ever tried making a digital map like on Google Maps? It's pretty hard. You need to keep track of every road, building, city, and shoreline on the planet. Maybe you'll need a fleet of cars to drive around the world and take pictures and measurements, as Google has done for Google Maps.[48] Oh, you'll also need to make a nice interface with features like panning and zooming. (Don't get us started on making an algorithm to find driving directions between two points.)

In short, making a digital map is really hard. But apps like Uber, Pokémon Go, and Yelp need to include a map to show where available cars are, help the user find wild Pokémon, or display nearby restaurants. Do these apps need to spend thousands of person-hours making their own maps?

Fortunately for them, the answer is no. And if you've ever used these apps, you can probably figure out what they do instead: they embed a Google Map in their app. Searching up restaurants? Yelp drops pins on a Google Map centered on your location. Want to Uber downtown? The app draws the route you'll take on a Google Map and calculates the approximate time it'll take to get there.[49]



Uber uses the Google Maps API to draw a map of your area and predict your car's travel time. Source: Uber on Android

Google lets you include a small snippet of code in your app to draw a Google Map. It also provides other snippets of code to let you draw on the maps, calculate driving directions between points on the map, and even find out the speed limit for a particular road. All these tools are cheap or even free.[50] These tools are big win for developers; they can use the technology

that has taken Google years to perfect with just a small amount of code. There's no need to reinvent the wheel!

These snippets of code that let you borrow another app's functionality or data are called APIs, or application programming interfaces. In short, APIs let apps talk to each other. Let's look at three main kinds of APIs.

The first kind of API lets one app ask another specialized app to solve a particular problem, like calculating driving directions, sending text messages, or translating sentences. It's like how you could call a plumber or carpenter to fix problems around your house instead of trying to do it yourself. One example of this kind of API: processing credit card payments is pretty difficult, so apps that charge you (like Uber[51]) can use PayPal's Braintree API, which lets anyone use PayPal's algorithm for processing credit card payments with just a few lines of code.[52] Similarly, it's a pain to write code that sends emails or text messages — so when apps like Venmo need to send you confirmation emails or texts, they can just use a specialized API.[53] And any app that lets you log in via Facebook or Google uses those websites' "Single Sign-On" APIs for verifying someone's identity.[54]

The second type of API lets one app ask another app to hand over some interesting information, such as sports scores, recent Tweets, or today's weather. It's like calling the front desk at a hotel to learn which museums and restaurants they recommend. ESPN offers an API that lets you get rosters for every major-league sports team and scores for every game.[55] New York's subway system lets you track where trains are and predict when the next train will arrive at a station.[56] The US

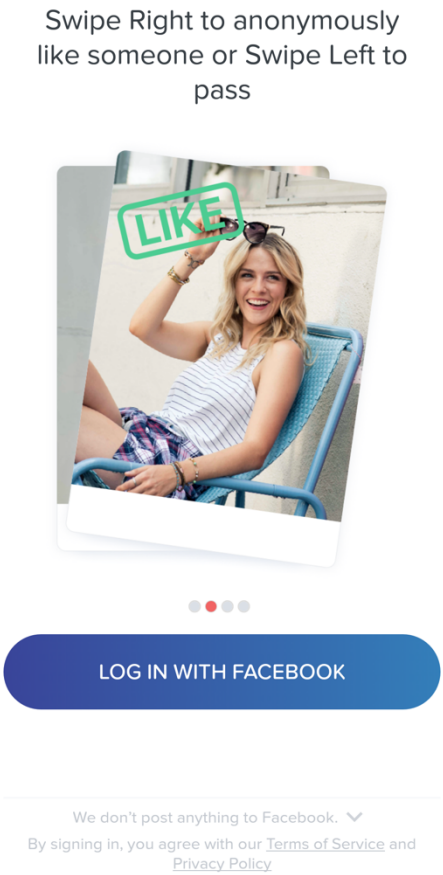
government offers APIs that let you access all kinds of data about American healthcare, energy, agriculture, and more.[57] And don't worry, there's even an API to get random images of cats.[58]

The final kind of APIs lets developers access features of the device itself. Snapchat taps into your phone's Camera API to zoom, focus, and snap photos. Google Maps itself uses your phone's Geolocation API to figure out where in the world you are. Your phone even has sensors called accelerometers and gyroscopes, which fitness apps (and Pokémon Go) use to determine which direction you're walking and how fast you're moving.[59]

It's worth noting that APIs aren't perfect. Using an API makes app developers' lives easier, but it also makes their apps dependent on the API.[60] If PayPal's Braintree payment API somehow stopped working, Uber would be unable to process payments — not great for their business. Still, a specialized company's API will probably be more reliable than any similar feature that a non-specialist company could make themselves.

All this brings us back to the question: what technologies do Uber, Yelp, and Pokémon Go have in common? They all use APIs, especially the Google Maps API, to avoid reinventing the wheel. Indeed, APIs are a core part of pretty much every app out there.

Why does Tinder make you log in with Facebook?



Tinder on Android. Note that the only way to sign in is with Facebook.

If you've ever used the dating app Tinder, you'll notice that they make you log in with your Facebook account before you can set up a profile. Once you connect your Facebook profile, Tinder imports information like your profile picture, your age, your list of friends, and the Facebook pages you like.[61] As you

might have guessed, this is done through an API that Facebook offers. With this API, any app can let users create accounts by linking their Facebook profiles.[62]

Why does Tinder use this API? For one, it ensures that there are no empty profiles (which no one would want to swipe on) since some basic information always gets imported from Facebook.[63] Second, requiring Facebook login helps them stop bots, since bots are less likely to have Facebook accounts.[64] Third, this helps Tinder make better matches: by gathering your friend list, Tinder can show you how many mutual friends you have with each potential match, and that sense of connection probably encourages people to swipe more. Finally, by getting the Facebook profiles of all their users, Tinder can get some deep insights into their user base, such as how old they are, where they live, or what they're interested in.[65] These insights can help Tinder tweak their app's design or advertising strategies.

Signing up with Facebook is also helpful for users. Making your profile with Tinder becomes much faster when most of your basic information and pictures are already imported from Facebook.[66] Seeing more completed profiles and fewer bots improves your experience, as well.[67] And signing up with Facebook means you don't have to remember yet another username and password.[68]

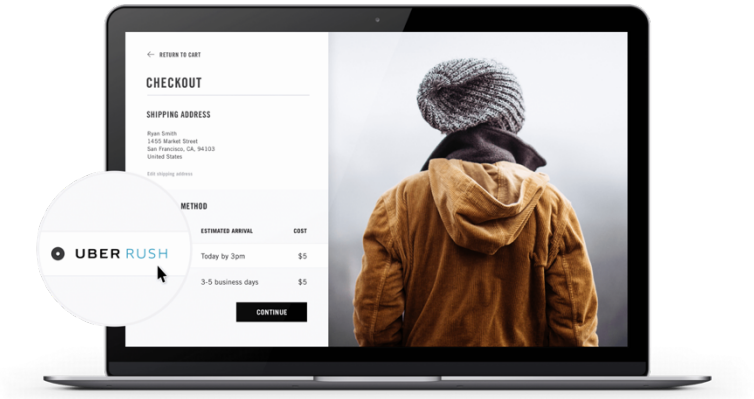
Why would Facebook publish the API that lets people sign in to other websites using their Facebook credentials? Well, when you use Facebook' sign-in API (often called "single sign-on," or SSO) to register for Tinder, Facebook realizes that you're a Tinder user. Facebook gets similar data points when you use

Facebook to log into other websites. Facebook can then use this data to target ads at you more effectively, for instance by showing more dating-related ads to Tinder users.[69]

As this example shows, publishing APIs is a great way for companies to gain data and usage, and using APIs helps apps save time and offer better features. The end goal is to help you, the user.

Why does Uber let other apps use Ubers to deliver their products?

In 2015, Uber published an API called UberRUSH that lets other app developers ship items short distances using Ubers.[70] For instance, a local florist could use the UberRUSH API to deliver flowers using Ubers instead of hiring their own fleet of drivers.[71] In other words, the API lets other companies rent Uber's huge driver network and logistics infrastructure.[72] Why did Uber bother going through the hassle of making and publishing this API?



An example of a website that uses the UberRUSH API to deliver items using Ubers. Source: Uber[73]

The most obvious reason is money. Uber can charge a bit for companies to use UberRUSH.[74] There are some more interesting reasons, though.

The second reason is to help Uber expand its market. Uber is already a leader in transporting people on demand,[75] but before UberRUSH, Uber wasn't competing in the market of

transporting *things* on demand. Uber wanted to be sure they won that market before someone else, like the fast food-delivery service Postmates, beat them to it.[76] In fact, UberRUSH might even let Uber start competing with traditional shipping companies like FedEx and UPS in the same-day shipping space.[77] It's a savvy business move.

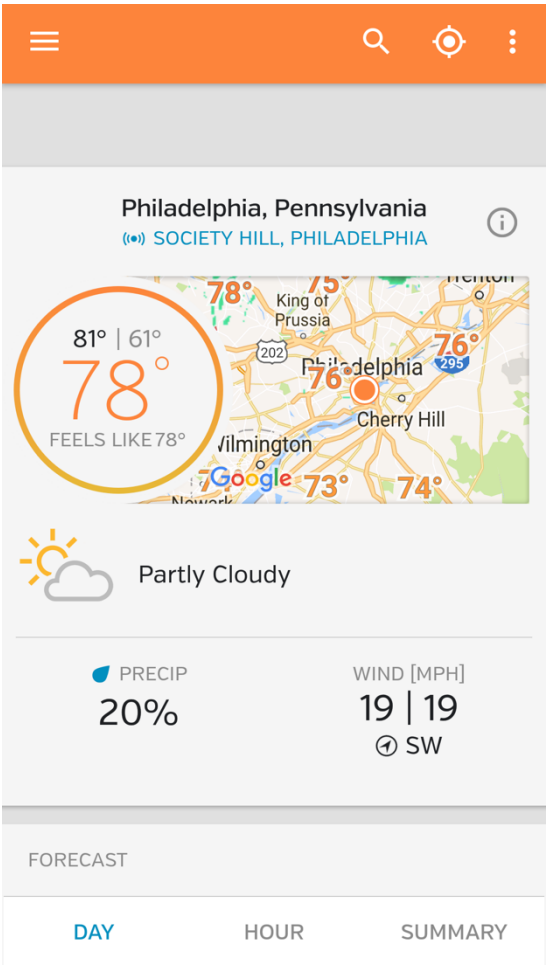
The third reason is users and data. By expanding into the "ship things short distances on demand" market, Uber can attract new customers and drivers and run more trips. Uber's key strength (or "core competency," as businesspeople call it[78]) is its huge driver network,[79] so attracting new drivers is a big win. More trips also means more ride data, which can help Uber improve its services — for instance, by helping Uber figure out where to place cars to minimize riders' wait times.[80]

So why does Uber publish an API that lets other app developers deliver stuff with Ubers? It's not just to be nice. The API lets Uber make money, expand into new markets, and gather more customers, drivers, and data. Meanwhile, the companies that use UberRUSH to ship things avoid the cost and headache of operating their own driver network.[81] It's a win-win.

How does your weather app work?

Pull out your phone and open the weather app. Let's use it as an example to show how you can understand almost any app by analyzing which APIs and algorithms it uses.

Your weather app will probably look something like this:



The Weather Underground app on Android.

Let's dive into the screenshot to figure out how the app works.

First, consider the APIs it uses. As you can tell from the map, the Weather Underground app embeds a Google Map (just like Uber, Yelp, and so many other apps) and then uses the Google Maps API to draw temperatures and weather patterns (green blobs for rain, white blobs for snow, etc.) on top.[82] Second, whenever you plug in a zip code, the app uses a zip code API to figure out which city you mean.[83] For instance, entering 60601 shows you the weather for Chicago, Illinois. Third, if you hit the target in the top right, the app will use your phone's Geolocation API to figure out your GPS location and show you the weather for that place. And finally, Weather Underground gets its weather data, such as the current temperature and probability of rain, from the National Weather Service, which offers all the weather data that the American government collects through a free API.[84]

Current weather information is great, but weather app users really want to know about future weather. So forecasters like the National Weather Service and Weather Underground use algorithms to predict future weather based on current conditions, recent weather trends, and other factors. The NWS, for instance, uses the Weather Forecasting Model, an algorithm that simulates the physics of the atmosphere to make weather predictions.[85]

These APIs and algorithms are the heart of the weather app, and they might just be enough to build a weather app yourself. If you use this same lens to look at your favorite apps — from

Pokémon Go to Snapchat to your to-do list app — you can start making sense of how these apps really work.

Why does every Washington Post article have two versions of the same headline?

Take a look at the following two screenshots of the same Washington Post piece. Notice anything different?

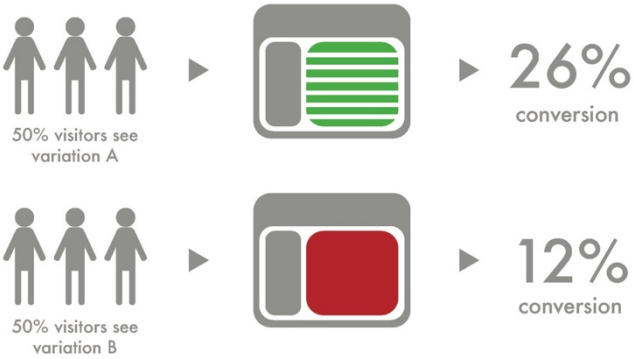
Notice a difference? The Washington Post shows people different versions of the same headline. It's part of an experiment to figure out which one gets more clicks. Source: The Washington Post[86]

That's right, the headlines are different! In 2016, the Washington Post rolled out a feature that lets article writers specify two different headlines for every article.[87] But why?

It's actually an experiment that the newspaper runs to maximize the number of clicks on its articles.[88] This experiment automatically shows one version of the headline to one group of visitors; let's say one-half of them, randomly chosen. It shows the other version to the remaining visitors. After letting the experiment run for a while, developers look at particular statistics, or metrics, like the number of clicks on the headline. Developers decide which version is better and show the winning version to everyone. This is a simple but powerful

way to improve an app's metrics. For instance, the first version of the above headline was clicked 3.3% of the time, while the bottom version was clicked 3.9%[89] — that's an 18% jump, just by changing a few words!

This technique is called A/B testing, which you might remember from the start of the chapter. A/B testing is a powerful, data-driven way to improve online products.[90] It's called "A/B testing" because you compare at least two versions of a feature, A and B.



A/B testing shows at least two variations of the same feature (A and B) and compares relevant metrics to decide which variation to push to all users. In this case, everyone would start seeing variation A, which was better at getting users to take a desired action (or "conversion"). Source: VWO[91]

Not sure which marketing catchphrase will get people to buy? Instead of endless debating, just run an A/B test! Not sure if a red or green "sign up" button will get more people to click? Run a test! (If you're curious, the red button got 34% more clicks in one experiment.[92]) Not sure which Tinder profile picture will get you the most swipes? Guess what: Tinder

automatically runs A/B tests to decide which of your photos, when shown as your main profile picture, gets you the most right swipes.[93]

That brings us back to the question: why does every Washington Post article have two versions of the same headline? It's part of the Washington Post's A/B testing framework, called Bandito. Bandito tries out different versions of the headline to see which one people click on more, and then shows the winning headline more often.[94] (Technically, it's a little more sophisticated than an A/B test — it's a so-called "multi-armed bandit" algorithm[95] — but the core idea is the same.)

A/B testing is very popular among news outlets. BuzzFeed uses A/B tests to find the most clickbaity headlines as well.[96] A BuzzFeed competitor called Upworthy actually tries up to 25 versions of the same headline in its quest to find the perfect one.[97] A/B testing is very important: according to Upworthy, the difference between a decent headline and a perfected one is 1,000 vs. 1,000,000 views.[98]

Many more apps and websites use A/B testing. Facebook, for instance, is always rolling out new features to "limited test audiences."[99] Snapchat lets advertisers A/B test their ads, to pick the ones that get tapped the most.[100] Even brick-and-mortar stores can do A/B testing: one startup, for instance, lets stores vary the background music they play to maximize shoppers' spending.[101] A/B testing really is everywhere.

Significance testing

There's one important caveat to keep in mind whenever you're doing statistical tests: you need to check whether your observed findings happened because of something meaningful or were just due to chance. For instance, if you flip a coin six times and get five heads, you can't be confident that the coin is unfairly weighted toward heads — it could just be dumb luck. But if you flipped the coin six hundred times and got five hundred heads, you might be on to something.

When companies perform A/B tests, experimenters report how one version changed a particular metric compared to the other version. They also report a statistic called a *p*-value, which shows the probability that the difference they observed was due to chance.[102] Usually, if $p < 0.05$ (i.e. there's a less than 5% chance that the difference was just random), they can assume the change was meaningful, or "statistically significant."[103] Otherwise, they can't be sure that their results weren't just dumb luck.

For example, say Amazon made the "Add to Cart" buttons a bit bigger for half their users and found a 5% increase in sales, with $p = 0.15$. While the bigger button seems like a great move, there's a 15% chance that the sales boost was caused by dumb luck, not the button. That 0.15 is greater than 0.05, so Amazon's testers won't roll out the bigger button. (A cutoff of 0.05 might seem too strict, but statisticians are very careful people.[104])

So if you ever get clickbaited by a headline like "18 Food Arguments So Strong That They End Friendships,"[105] don't feel too bad — you're up against a powerful blend of social

science, software development, and statistics. Like it or not, A/B testing is extremely effective.