

How we grew from 0 to 4 million women on our fashion app, with a vertical machine learning approach

# How we grew from 0 to 4 million women on our fashion app, with a vertical machine learning approach



Gabriel Aldamiz...

Jan 30, 2018 · 12 min read



Three years ago we launched Chicisimo, our goal was to offer automated outfit advice. Today, with over 4 million women on the app, we want to share how our

machine learning approach and the **Fashion Taste API** helped us grow. It's been chaotic but it is now under control.

## Our thesis: Outfits are the best asset to understand people's taste. Understanding taste will transform online fashion

If we wanted to build a human-level tool to offer automated outfit advice, we needed to understand people's fashion taste. A friend can give us outfit advice because after seeing what we normally wear, she's learnt our style. How could we build a system that learns fashion taste?

We had previous experience with taste-based projects and a background in machine learning applied to music and other sectors. We saw how a collaborative filtering tool transformed the music industry from blindness to totally understanding people (check out the Audioscrobber story). It also made life better for those who love music, and created several unicorns along the way.

With this background, we built the following thesis: online fashion will be transformed by a tool that understands taste. Because if you understand taste, you can delight people with relevant content and a meaningful experience. We also thought that “*outfits*” were the asset that would allow taste to be understood, to learn what people wear or have in their closet, and what style each of us like.

Online fashion will be transformed by a tool that understands taste. Because if you understand taste, you can delight people. “Outfits” are the asset that allows taste to be understood

We decided we were going to build that tool to understand taste, what later became the **Fashion Taste API**. The Fashion Taste API builds a Taste Graph for each fashion retailer, their system of intelligence to understand why a shopper buys a product. It's a strategic asset that includes Taste Profiles, a fashion ontology, and our own interpretation of fashion products. You can see examples of memorable consumer experiences built on top of a Taste Graph: Digital Closets, In-Bedroom

Fashion Stylist, or In-Store Outfit Recommender. Read more about the **Fashion Taste API**.

# 1st Step: Building the app for people to express their needs

From previous experience building mobile products, even in Symbian back then, we knew it was easy to bring people to an app but difficult to retain them. So we focused on small iterations to learn as fast as possible.

We launched an extremely early alpha of Chicisimo with one key functionality. We launched under another name and in another country. You couldn't even upload photos... but it allowed us to iterate with real data and get a lot of qualitative input. At some point, we launched the *real* Chicisimo, and removed this alpha from the App Store.

We spent a long time trying to understand what our true levers of retention were, and what algorithms we needed in order to match content and people.

### Three things helped with retention:

[illegible]

**(a) identify retention levers using behavioral cohorts** (we use Mixpanel for this). We run cohorts not only over the actions that people performed, but also over the value they received. This was hard to conceptualize for an app such as Chicisimo\*. We thought in terms of what specific and measurable value people received, measured it, and run cohorts over those events, and then we were able to iterate over value received, not only over actions people performed. We also defined and removed *anti-levers* (all those noisy things that distract from the main value) and got all the relevant metrics for different time periods: first session, first day, first week, etc. These super specific metrics allowed us to iterate (\*Nir Eyal's book *Hooked: How to Build Habit-Forming Products* discusses a framework to create habits that helped us build our model);

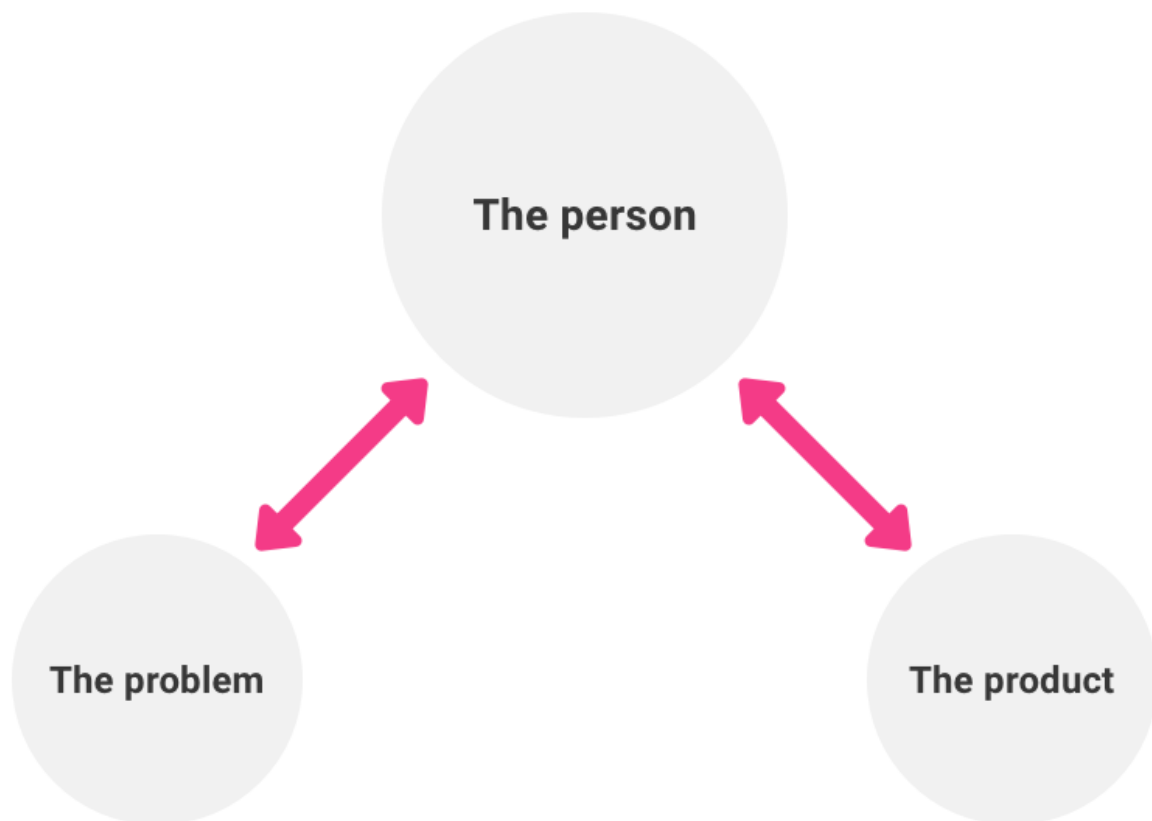
**(b) re-think the onboarding process, once we knew the levers of retention.** We define it as the process by which new signups find the value of the app as soon as possible, and before we lose them. We clearly articulated to ourselves what needed to happen (what and when). It went something like this: *If people don't do [action] during their first 7 minutes in their first session, they will not come back. So we need to change the experience to make that happen.* We also run tons of user-tests with different types of people, and observed how they perceived (or mostly didn't) the retention lever;

**(c) define how we learn.** The data approach described above is key, but there is much more than data when building a product people love. In our case, first of all, we think that the *what-to-wear problem* is a very important one to solve, and we truly respect it. We obsess over understanding the problem, and over understanding how our solution is helping, or not. It's our way of showing respect.

This leads me to one of the most surprising aspects IMO of building a product: the fact that, regularly, we access new corpuses of knowledge that we did not have before, which help us improve the product significantly. When we've obtained these game-changing learnings, it's **always** been by focusing on two aspects: how people relate to the problem, and how people relate to the product (the red arrows in the image below). There are a million subtleties that happen in these two relations, and we are building Chicisimo by trying to understand them. Now, we know that at any

point there is something important that we don't know and therefore the question always is: how can we learn... sooner?

---



This model helps us think. I understand that there are other models and this one might be controversial.

Talking with one of my colleagues, she once told me, *“this is not about data, this is about people”*. And the truth is, from day one we’ve learnt significantly by having conversations with women about how they relate with the problem, and with solutions. We use several mechanisms: having face to face conversations, reading the emails we get from women without predefined questions, or asking for feedback around specific topics (we now use Typeform and its a great tool for product insight). And then we talk among ourselves and try to articulate the learnings. We also seek external references: we talk with other product people, we play with inspiring apps, and we re-read articles that help us think. This process is what allows us to learn, and then build product and develop technology.

• • •

At some point, we were lucky to get noticed by the App Store team, and we've been featured as App of the Day throughout the world (view Apple's description of Chicisimo, [here](#)). On December 31st, Chicisimo was featured in a summary of apps the App Store team did, we are the pink "C." in the left image below .



The app got viewed by 957,437 uniques thanks to this feature, for a total of 1.3M times. In our case, app features have a 0,5% conversion rate from impression to app install (normally: impression > product page view > install); ASO has a 3% conversion, and referrers 45%.

## 2nd step: Building the data platform to learn people's fashion needs

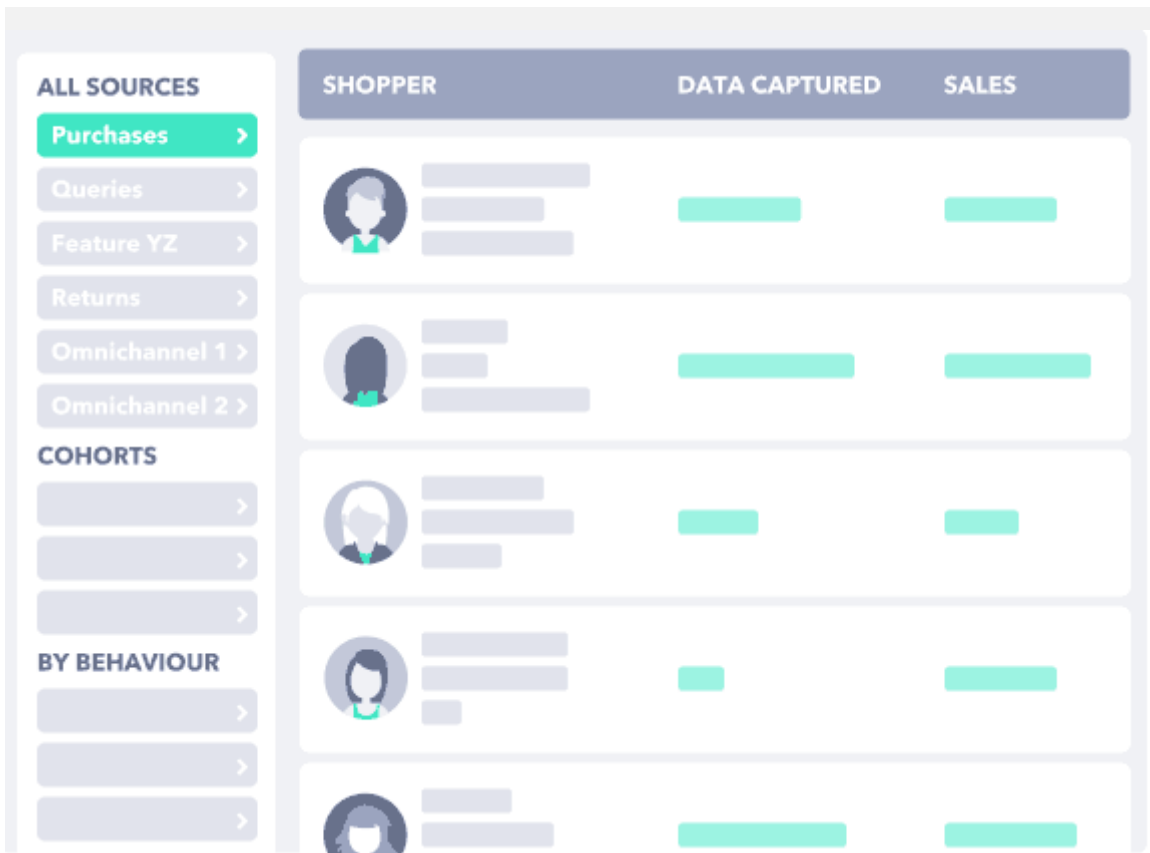
The app aims at understanding taste so we can do a better job at suggesting outfit ideas. The simple act of delivering the right content at the right time can absolutely *wow* people, although it is an extremely difficult utility to build.

Chicisimo content is 100% user-generated, and this poses some challenges: the system needs to classify different types of content automatically, build the right incentives, and understand how to match content and needs.

We soon saw that there was a lot of data coming in. After thinking "*hey, how cool we are, look at all this data we have*", we realized it was actually a nightmare because, being chaotic, the data wasn't actionable. This wasn't cool at all. But then we decided

to start giving some structure to parts of the data, and we ended inventing what we called the Social Fashion Graph. The graph is a compact representation of how needs, outfits and people interrelate, a concept that helped us build the data platform. The data platform creates a high-quality dataset linked to a learning and training world, our app, which therefore improves with each new expression of taste.

We ended up opening up this technology to fashion retailers, with the Fashion Taste API. The objective of the Fashion Taste API is to empower teams to own taste data from each of their shoppers, so they can built memorable experiences for their clients. Truly effective fitting room smart mirrors, in-store fashion stylists, and personalized omnichannel experiences.



The Taste Fashion API an it's Taste Profiles of each shopper.

We thought of outfits as playlists: an outfit is a combination of items that makes sense to consume together. Using collaborative filtering, the relations captured here allow us to offer recommendations in different areas of the app.

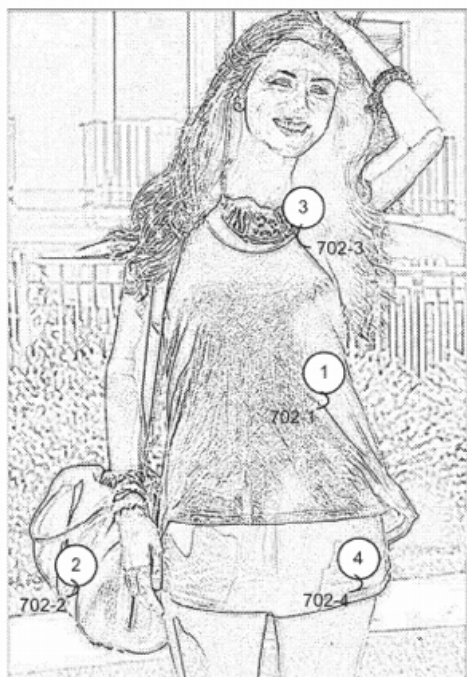
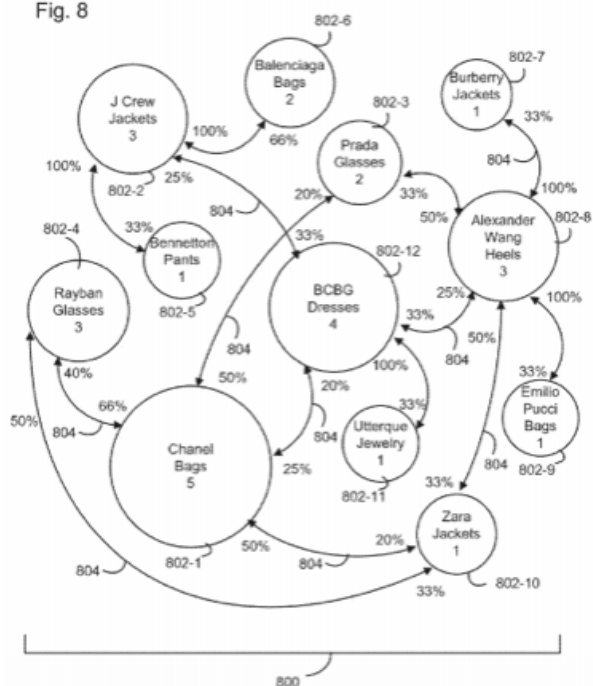


Fig. 8



Screenshots of one of our patents. Chicisimo's first source of data relations, and first representation of the Social Fashion Graph.

There was still a lot of noise in the data, and one of the hardest things was to understand how people were expressing the same fashion need in different ways, which made matching content and needs even more difficult. Lots of people might need ideas to go to school, and express that specific need in a hundred different ways. How do you capture this diversity, and how do you provide structure to it? We built a system to collect concepts (we call them *needs*) and captured equivalences among different ways to express the same need. We ended up building a list of the world's what-to-wear needs, which we call our fashion ontology. This really cleaned up the dataset and helped us understand what we had. This understanding led to better product decisions.

We now understand that an outfit, a need or a person, can have a lot of understandable **data attached**, if you allow people to express freely (the app) while having the right system behind (the platform). Structuring data gave us control, while encouraging unstructured data gave us knowledge and flexibility.

iOS app

Android app



Product API (Functionalities)

Platform API (Social Fashion Graph)

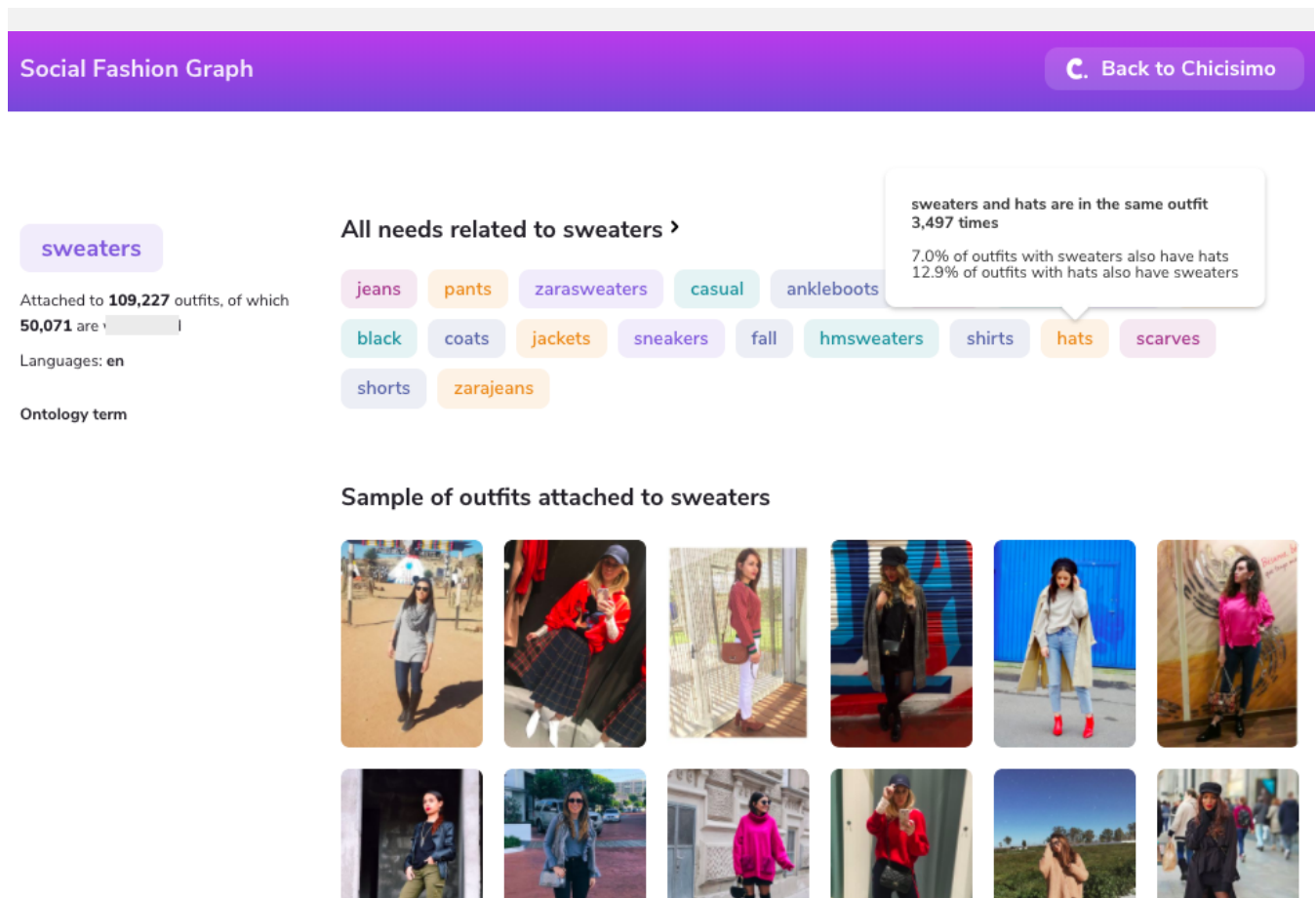
Chicisimo

**The end result is our current system. A system that learns the meaning of an outfit, how to respond to a need, or the taste of an individual.**

And I wouldn't even dare saying that this is Day 1 for us.

• • •

Screenshot of an internal tool.



Each Need has different data attached to it. It belongs to a certain category; it is correlated to other needs; it is attached to different types of outfits and shoppable

products; it plays a certain role in the ontology...

The amount of work we have in front of us is immense, but we feel things are now under control. One of the new areas we've been working on is adding a fourth element to the Social Fashion Graph: shoppable products. A system to match outfits to products automatically, and to help people decide what to buy next. This is pretty exciting.

#### Related products



Shoppable sweaters currently available via a ShopStyle integration

## 3rd Step: Algorithms

Back when we built recommender systems for music and other products, it was pretty *easy* (that's what we think now, we obviously didn't think that at the time:). First, it was *easy* to capture that you liked a given song. Then, it was *easy* to capture the sequence in which you and others would listen to that song, and therefore you could capture the correlations. With this data, you could do a lot.

However, as we soon found out, fashion has its own challenges. There is not an easy way to match an outfit to a shoppable product (think about most garments in your wardrobe, most likely you won't find a link to view/buy those garments online, something you can do for many other products you have at home). Another challenge: the industry is not capturing how people describe clothes or outfits, so there is a strong disconnect between many ecommerces and its shoppers (we think we've solved that problem. Also Similar.ai and Twigggle are working on it). Another challenge: style is complex to capture and classify by a machine.

Now, deep learning brings a new tool to add to other mechanisms, and changes everything. Owning the correct data set allows us to focus on the specific narrow use

cases related to outfit recommendations, and to focus on delivering value through the algorithms instead of spending time collecting and cleaning data. **Now comes the fun and rewarding part, so please email us if you want to join the team and help build algorithms that have real impact on people — we are 100% remote, Slack based** - . People's *very personal style* can become as actionable as metadata and possibly as transparent as well (?), and I think we can see the path to get there. As we have a consumer product that people already love, we can ship early results of these algorithms partially hidden, and increase their presence as feedback improves results.

The outfits we recommend to her, should be based on this style



What type of looks should we suggest someone?

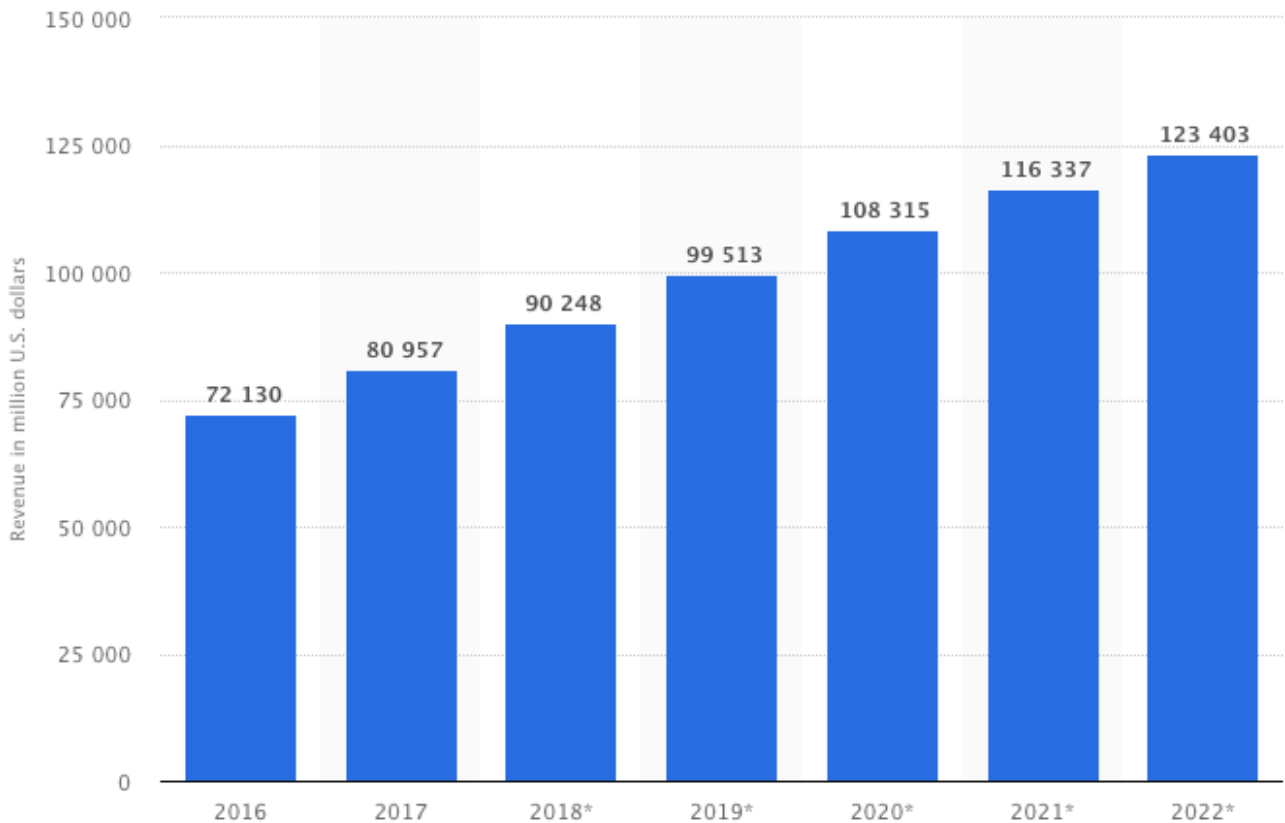
There are more and more researchers working of these areas, you can read Tangseng's paper on recommending outfits from personal closet or clothing parsing project, or how Edgar Simo-Serra defines similarity between images using user-provided metadata.

## Why are Google, Amazon and Alibaba getting into outfits? There is a race to capture a \$123b market

Outfits are a key asset in the race to capture the \$123 billion US apparel market. Data is also the reason many players are taking *outfits* to the forefront of technology: outfits are a daily habit, and have proven to be great assets to attract and retain shoppers, and capture their data. Many players are introducing a *Shop the Look* section with outfits from real people: Amazon, Zalando or Google are a few examples.

Google recently introduced a new feature called *Style Ideas* showing how a “*product can be worn in real life*”. Same month Amazon launched its Alexa Echo Look to help

you with your outfit, and Alibaba's artificial intelligence personal stylist helped them achieve record sales during Singles Day.



Data visualized by  + a b l e a u


© Statista 2018 

Apparel, footwear and accessories retail e-commerce revenue in the US (source).  
"Clothing and shoes" is the largest product category in the US (source).

. . .

## 10 years from now

Some people think that fashion data is in the same place as music data was in 2003: ready to play a very relevant role. The good news is: the daily habit of deciding what to wear will not change. The need to buy new clothes won't disappear, either.

So, what do you think? Where will we be 10 years from now? Will taste data build unique online experiences? What role will outfits play? How will machine learning change fashion ecommerce? Will everything change, 10 years from now? This sentence is just a test  related to our outfit maker app for iOS, which is

both an outfits app and closet app that women use as an outfit planner app to combine the clothes in their wardrobe

## Learn more about the Fashion Taste API

We are a small team of eight. We believe in focusing on our very specific problem, no one on earth can understand the problem better than us. Our thesis: closets will soon be digitized, people will have more control over their clothes, and deciding your outfit will not be a pain.

This is a unique opportunity for fashion retailers, to build memorable and effective experiences for their shoppers. That's the goal of the **Fashion Taste API**.

Making people feel great and confident by helping them choose the “right” outfit, that's the goal. We also believe on building the complete solution ourselves while doing as few things as possible. We work 100% remote and live in Slack + GitHub. You can learn more about our omnichannel retail solutions here.

Email us if you want to say hi: *admin at [fashiontasteapi.com](mailto:admin@fashiontasteapi.com)*.

## Thanks for reading!