# RecSys Discussion

Sam Reeves

4/21/2021

### The Company

Here we are taking a look at Wireless Emporium, the "original mobile accessory superstore". They are an online retailer selling, you guessed it, mobile accessories.

On the front page, you will notice an "accessory finder": you put in your brand and model of phone and a chosen category (I guess most people want items thrust at them), and hit the big red GO button. I imagine this information is enough to effectively categorize the majority of potential customers.

### The Rec System

I was led to this website for this assignment while browsing through some lists on GitHub. Apparently, WE uses a product called Strands for their recommender system. According to this page, Strands claims that WE experienced a 33.6% increase in orders and a twofold increase in overall sales conversion when they started using this tailor-made recommender.

Although this is a closed-source solution, Strands does publish some information about their approach.

They state "a 3-pronged approach to recommend the right accessory for the right model to the right customer – at the right time". Seems to me that they rely heavily on the data a website visitor would put in upon landing on the main page.

They list the three prongs:

- Using Strands' flexible catalog integration and mapping system, WE quickly and easily added product attribute fields so that all relevant characteristics of every mobile device could be easily recognized
- For those 100,000 products, Strands normalized all WE attribute data, so that the right accessories would automatically match the right core products
- Using the Strands BackOffice plug-and-play "Adjust Logic" and "Look & Feel" wizards, WE easily set up and then deployed "Your Recommendations" and "People who Bought this Also Bought That" onto the Wireless Emporium website. After integration, it tested performance with Strands A/B testing mechanisms to refine and improve performance over time, and chose winners to tighten merchandising performance and extract all possible value.

### Let's try it out.

I assume this is not going to work well for me simply because I don't actually want to buy anything. If Strands is as successful as they say they are, then I guess the algorithm won't suggest things to a person who isn't actually looking?

I put in my info: Google Pixel 3a, wallets and clutches…

My first suggested item is a "tactical" light desert camo carrying pouch for $6.99.

Let's change the category.

I change to covers and cases, and it still pushes the same "tactical" pouch.

I still don't want it. I click on a regular black plastic case, and I add a bejewled pink power bank to my cart. I go back to the wallets and clutches section, after it has shown me that it remembers the two items I have viewed.

Again, it shows the tacticool gear first. None of the items on the first page have shifted or changed.

### Scenario design

1. The target users are people who have phones and don't like to shop. Their key goals are finding something at a lower price than they might find at a store, or finding a niche item that they would find in a normal shop. WE can help them by specifically showing things that people want before they bother to search for them or by showing them items they didn't know existed. It may make sense for the company to consider these two groups of people separately and perform two distinct scenario designs.
2. According to my sources, WE relies heavily on the information placed in the drop down menus. It would make sense to trust people who have already visited the site, at the very least to look at items. Since Strands places emphasis on showing the right product at the right time, I find it odd that the ordering of items shown in a categorical search is not related to items I have viewed. No other items were suggested to me when I viewed my cart.
3. I think it may be computing intensive and thus expensive (or would affect the performance of a website guest's machine) to constantly compute a best order for listed items. Perhaps the best would be to bake the information from real visitors (those who bought and also those who didn't) into a file which can be referenced in the website source. Something similar to RARD II. This is a dataset from the Harvard Dataverse, composed from 94 million recommendations, which helps form the backend of a recommender system for academic articles.

The information on the increased sales exists. They should use it dynamically. Since the choices to be made here, on the part of the customer, are fairly straightforward, it should be an engine based off of:

- A fixed body of information from past website visitors
- High-granularity location information (some places have bad local stores)
- Time and day sensitivity
- Effects from items viewed or placed in the cart
- Maybe another question or two for the input data: Do you want something cheap and sturdy or something flashy and fresh?