SizeRec: Implementing a Sizing Recommendation Software-as-a-Service Solution for Online Retailers

Christopher Wan
New Haven, Connecticut

Abstract

Every day, fashion retailers lose an enormous chunk of potential online revenue, because many customers do not know what size will fit them. This is especially the case for first-time customers. This problem is exacerbated by inconsistent sizing across various brands, so knowledge of a perfect fit in one brand does not translate into a perfect fit in the same size of another brand. Currently, retailers solve this problem by publishing size charts, which provide customers with recommended sizes for different body measurements. While helpful for experienced online shoppers, these sizing charts are completely useless for the majority of shoppers, who do not know their body measurements. However, even if these shoppers do end up making a purchase, many will return the clothing after finding that it does not fit. Once the clothing is returned, many retailers will no longer put the items back on the rack for sale. As a result, the retail industry loses billions of dollars per year in revenue from returns. We posit that a mechanism that increases customer confidence in sizing would significantly boost a retailer’s revenue and reduce the amount of returns. Thus, SizeRec is a software service that helps recommend sizes for customers based on simple user input: sizes customers wear in other brands. SizeRec collects sizing data from the most popular men’s and women’s fashion brands, then maps these sizes from these brands to a particular size for the current brand in question. SizeRec is implemented as a button on product pages, right before the customer adds the item to the cart. SizeRec is currently live and being used by a number of online retailers to increase their revenue.

Keywords: Recommendation algorithm, Software Engineering, SaaS products
1. Introduction and Background

With the proliferation of the Internet, we have seen an explosion in online retail. Indeed, according to Big Commerce, 51% of Americans prefer to shop online, and 67% of Millennials and 56% of Gen Xers prefer to shop online rather than in-store [1]. However, even with the large number of online shoppers, there are significant shortcomings from online shopping that leave much to be desired. For instance, 49% of individuals say that one of their least favorite aspects of online shopping is their inability to try a product before purchasing it. As a result, 42% of online shoppers have made a purchase they later regret. This is exacerbated by the problem of vanity sizing, the practice stores use to alter sizes in order to psychologically make their customers more happy about their purchase. The large number of online shoppers, combined with their dissatisfaction, presents a large market opportunity to address some of these concerns. Indeed, online shoppers are often dissatisfied with the fit of clothing in particular. Whereas customers can simply try on various outfits in traditional brick-and-mortar stores, these customers are often blind to the fit of clothing when shopping online. In trying to tackle this problem, we first researched our competitors, and how we can best enter the market.

1.1. Vanity Sizing

Vanity sizing is a problem that makes online shopping especially difficult [2]. In particular, vanity sizing involves taking a particular size of clothing, such as Small, and making its dimensions larger, giving the false notion that the customer is wearing a more petite article of clothing when the particular article actually has larger dimensions. Therefore, those who wear the clothing in a particular size will psychologically feel better that they are wearing a smaller size even though the dimensions of the clothing are large. This problem is particularly poignant for women, where a size 2 in some vanity-sized brands may correspond to a size 8 in other non-vanity-sized brands. Because of vanity sizing, many individuals, especially women, are hesitant to purchase clothing online.

1.2. Competitors

TrueFit: TrueFit is a data-driven personalization platform for footwear and apparel retailers that uses rich connected data and machine learning to enable personal experiences
for fashion retailers. It has organized a platform of apparel and footwear data through its partnerships with thousands of top brands, the world’s leading retailers, and millions of consumers. TrueFit integrates on a store’s product page, and it asks customers for a host of their individual data, including their height, weight, and sizes they wear in their other brands. As a multi-million dollar company, they are already backed by brands as large as Nordstrom and Adidas. Their pricing, however, is quite high, at over $ 2000 / month.

**VirtuSize**: VirtuSize does not have a sophisticated database of measurements from apparel retailers, but it does ask customers for their articles of clothing in other brands. The biggest problem with this is that this solution requires customers to know the measurements of their clothing, and this measurement process can be a hassle. A large benefit of VirtuSize is that once it has the customer data, it provides a very compelling visualization for how the current article of clothing would fit on the customer’s body.

We believe that TrueFit’s solution is more effective at increasing customer conversions because it requires the least amount of work for the customer. Whereas VirtuSize requires the customer to intimately know the measurements of his/her own clothes, TrueFit only requires the customer to know the size. However, TrueFit is aimed at the higher end of the market. After doing customer research, it is clear that many lower and middle-end stores have declined to work with TrueFit due to their prohibitively expensive pricing. While TrueFit does have big brands as partners, these big brands have more money to use a service like TrueFit.

Therefore, in developing SizeRec, we aim to create a solution similar to TrueFit, where we will only ask customers for the sizes of the clothing in their favorite brands. However, we aim to target the lower end of the market and position ourselves as a quick, painless solution with minimal overhead for online sizing inconsistencies.

1.3. *Shopify*

To target this lower end of the market, we identified Shopify as the best marketplace to initially sell. Shopify is an e-commerce platform that allows anyone to easily sell online [3]. Many boutique stores that have just been started will use Shopify to get their store up and
running. Indeed, while larger enterprises can afford their own engineers to develop a sophisticated stack for their website, smaller stores lack such an engineering team and will often turn to Shopify to get their stores running with minimal costs. Shopify also has an extensive App Store, where online stores can purchase apps to make their stores more compelling. From themes to giveaway applications, the Shopify App Store sells many software solutions for relatively cheap prices, ranging from Free to hundreds of dollars. According to Shopify, Shopify boasts over 300,000 merchants as of August 2016, and 80,000 Shopify stores have at least one app installed [4].

Shopify exposes extensive APIs to developers to create applications for their App Store. Given Shopify’s wide reach and support for new applications, we found it a natural choice to begin development on Shopify.

2. Specification and Deliverables

My goal is to build SizeRec, a software-as-a-service for online retailers to help their customers find the perfect size. The customer flow would be as follows:

1. Customer clicks on the SizeRec button
2. The button opens up a SizeRec modal, where users can enter sizes they wear in other brands
3. SizeRec recommends a size based on user input

To be an effective solution for retailers, the integration costs of SizeRec have to be minimal. Fortunately, the majority of e-commerce websites are created with e-commerce platforms, such as Shopify. These e-commerce platforms allow developers to create “apps”, which retailers can then install via an app store. SizeRec will first be implemented as a Shopify app.

I have worked with Minh Tri Pham on this project, and we have clearly delineated our responsibilities. In particular, I have been in charge of the following parts of this project:

- Server setup and deployment in Flask
- Data collection and organization
• Integration with Shopify

• Data analytics

• Store Flow

3. Methods and Implementation

3.1. Server setup and deployment

The web development programming paradigm involves Models, Views, and Controllers (MVC) [3]. In this programming paradigm, when a client hits an endpoint of a webpage, the endpoint corresponds to a controller of the server. The controller is responsible for translating user input and communicating this input with the models. The models correspondingly represent how data for the web application is stored and abstracted. Once the controller has grabbed the relevant data from the models, this data is presented to the client on the front-end as a view.

In this project, I used the Flask microframework and applied the MVC programming paradigm. Flask uses Python for all back-end logic in its controllers and models, and it uses HTML, CSS, and Javascript to render its views [6]. Flask also makes use of the Jinja2 templating engine to render data sent from the controllers to the view. For the models, I used SQLAlchemy, an open source SQL toolkit and object-relational mapper for Python.

I deployed SizeRec using Heroku, a cloud Platform-as-a-Service that supports several programming languages and is used as a web application deployment model. Heroku uses dynos, lightweight Linux containers that runs a single user-specified command. For my Heroku deployment, I used a free dyno plan, which provides for 550 hours of uptime per month. I also use the Heroku Postgres Database with the Hobby Dev plan, which provides 10,000 rows, continuous protection, direct SQL access, and a connection limit of 20.

3.2. Data collection and organization

SizeRec is a service that depends on thousands of data points from stores on their size charts. When a customer uses SizeRec, they input their sizes in their favorite brand of
clothing, and it is our job to use measurements from those brands and map them onto the current brand in question. Therefore, data collection was an integral part of this project.

To collect data, we divided items of clothing into a few broad categories, including:

- Men’s Tops
- Men’s Bottoms
- Women’s Tops
- Women’s Bottoms
- Shoes

Each of these categories has variables that are particularly important to fit. For instance, for Men’s Tops, the size of an article of clothing depends on the chest circumference, neck circumference, and waist circumference. Similarly, for Women’s Tops, the size depends on the bust size, waist size, and hip size. So, for each of these categories, I identified primary, secondary, and tertiary variables that affect the size of a particular article.

So far, we have collected data for Men’s Tops and Women’s Tops. To collect these data, I found 60 of the most popular men and women’s brands, and we visited their stores to get their size chart information. I require that the primary variable for each size chart be present, but the secondary and tertiary variables are nullable. I then wrote a Python script to automatically insert these brands into our database.

The structure of our clothing database thus looks as follows:

- **ClothingCategory** - The ClothingCategory model contains information about the category name (e.g., Men’s Tops or Women’s Bottoms), and the primary, secondary, and tertiary variables (e.g., waist, chest, hip, etc.).

- **SizeChart** - The SizeChart model contains information about the brand, the corresponding ClothingCategory, and the MeasureValues for the particular article of clothing. The SizeChart model also contains a column for the style of the clothing. For instance, many women’s brands have Plus sizes and Petite sizes for their Tops.
• MeasureValue - The MeasureValue model contains information about the size of a piece of clothing (S, M, etc.,) and the ranges of the measurements of the corresponding variables for a particular item of clothing. For instance, suppose a Banana Republic Men’s Top in Small has a chest circumference of 36-38 inches. The corresponding MeasureValue model will contain the lower bound and the upper bound (36 and 38, respectively). Each SizeChart will have an array of MeasureValue objects that provide information about the various sizes of the particular SizeChart.

3.3. Integration with Shopify

In integrating with Shopify, the goal was to allow each Shopify store administrator to log onto SizeRec using Shopify’s OAuth platform and add SizeRec to their individual stores accordingly. As mentioned in Section II, Shopify exposes an API to developers to allow them to cleanly create apps for the Shopify platform. In particular, I used the Shopify Python API to make calls to the native Shopify API. Using the Shopify Python API encompassed the following components:

OAuth: Creating login functionality on Shopify requires using their OAuth protocol. In particular, I set up a Shopify session using my API Key and shared secret [5]. I then specify permissions that I require from the Shopify admin in particular. These permissions include the ability to read their products and modify their template files. Shopify then generates a token that expires after 24 hours. This token is then used to log the Shopify admin into SizeRec, provided the admin inputs the correct login credentials. I refresh this token every single time the Shopify admin logs in to SizeRec.

Assets: Each Shopify store has a Product HTML page that SizeRec needs to modify in order to inject the SizeRec button. Modifying these pages must be done through the Shopify API. In particular, each Shopify store uses a theme that has a ‘product.liquid’ file that contains the template code for the page that contains the Add to Cart button, which is where the SizeRec button must be injected. Using the Shopify API, I grab the ‘product.liquid’ file, modify it to include our SizeRec scripts, and save the file.

Products: Every Shopify online retail store has a number of products they sell, ranging
from T-Shirts to Blouses. However, these products are not standardized across all Shopify stores. For instance, while store A may have a product labeled ‘T-Shirts’, this does not necessarily mean that store B may have the same product label. However, each of these products for a store has their own sizes according to each particular store in question. Thus, I make use of the Shopify API to pull all of the unique products of a Shopify store. I then serve all of these products to the store admin and allow the admin to input the sizing information for each of their products.

3.4. Store Flow

It is important to tackle the store admin experience after they have signed on with SizeRec. In particular, SizeRec has a number of ClothingCategories that correspond to items such as Women’s Tops and Men’s Tops. However, each individual Shopify store has their own product categories, such as T-Shirts and Blouses that must be mapped to SizeRec ClothingCategories.

When a store begins with SizeRec, they begin by choosing one of our ClothingCategories, and they then are brought to a page where they can map some of their product types to the particular ClothingCategory in question. For instance, the Men’s Top ClothingCategory could be mapped to Men’s T-Shirts, Men’s Sweaters, etc., for a particular store. After creating the mappings, the store admin is then taken to a page where then can enter the sizing information for their products, using the primary, secondary, and tertiary variables specified by SizeRec.

Once the store has confirmed their mappings and the corresponding sizing information, they can then choose the activate or deactivate the SizeRec button for their store products. Each Shopify admin has a dashboard that they can use to modify their size charts and the activation of their SizeRec buttons.

The various flow components of the Shopify admin are shown in Figures 1-3 on the following page.
Figure 1: A Store Admin maps their product types to SizeRec's clothing categories.

Figure 2: A Store Admin enters sizing information about their product types.
3.5. Data Analytics

In order to assess the effectiveness of SizeRec for a particular store, it is important to include analytics for customer usage. Currently, I have implemented number of daily clicks on the SizeRec button for a particular store in addition to the number of daily clicks on the SizeRec button for a particular store’s product.

To do this, in the SizeRec script injection, I include meta information about the current store and the current product for the store. On every click of the SizeRec button, both of these pieces of information are passed to our servers, where we log a Click event (a model) that contains information about the time of the click, the particular store it was clicked for, and the store’s product.

When the Shopify admin logs into their dashboard, I query all daily Click events for the store. All of this information is then displayed to the Shopify store admin graphically using Chart.js, a Javascript library that renders data in user-friendly graphs.
4. Discussion

4.1. Challenges

There were a number of challenges that I encountered in the development of SizeRec. To begin, it was difficult to get started with Flask because the MVC programming paradigm is very different from what traditional computer scientists are accustomed to. Figuring how to set up and hook the models, views, and controllers with each other required looking at many examples and going through many tutorials.

Another challenge was getting set up with the Shopify API. Although there is documentation for the Shopify API, this documentation is by no means extensive. In addition, the Shopify Python API wrapper, although developed by Shopify itself, was even more lacking in documentation than the native API. The support for the API was also limited, as developers failed to respond to queries via email, Twitter, or Shopify online forums. To figure out how to fully use the Shopify Python API, I had to pore through the public test scripts and see how Shopify was calling the API to run their tests. I then had to adapt these test scripts to the SizeRec codebase in order to implement OAuth, Asset modification, and Product querying.

Another challenge was data organization. Indeed, we began organizing our data without stratification. In other words, for a particular clothing category, such as Men’s Tops, we had a model that corresponded to Men’s Tops and contained all information about the brand, clothing category, and measurements. However, this became unscalable as we expanded to other product categories. The table grew very large, and I realized that querying would become inefficient. So, we abstracted our models to include ClothingCategories, SizeCharts, and MeasureValues. This approach removed the need to create an entirely new model for each clothing category.

4.2. Future Work

One avenue of future work involves creating new ClothingCategories for Shopify stores to integrate their own products with. As of right now, the majority Shopify stores include
products that span much farther than the Men’s Tops and Women’s Tops that we are currently supporting. However, it is important to note that the problem of Men’s and Women’s bottoms are much more difficult to tackle. For instance, a pair of Men’s jeans are sized by numbers (e.g., 32 x 30) that correspond to the waist circumference and the leg inseam, respectively. In reality, though, a pair of Men’s jeans sized 32 x 30 will not necessarily have a 32-inch waist circumference and a 30-inch leg inseam. Indeed, this once again underscore the prevalence of vanity sizing. Online size charts will not admit to vanity sizing, so research or partnership would need to be formed with individual stores in order to find out the actual measurements of these articles of clothing. So, future work would need to be done in contacting the research and development departments of various stores to acquire their actual measurements for bottoms.

In addition, although the Shopify market is vast, our goal is to expand outside of the Shopify market and target online retailers in general. This would require us to create functionality that integrates outside of the Shopify API. We would need to write a general Javascript snippet that can be injected onto any store product page. However, it is important to note that this would almost necessarily require effort from the store developers to paste our code into their product page. This would make our ease of integration much more difficult since, using the Shopify API, SizeRec can perform this injection without any intervention from store developers.

SizeRec also has basic data analytics built in, including the number of clicks per day per store and number of clicks per day per store per product type. However, much more rich analytics can be provided in the future. In particular, an online store will often know very little about a customer when the customer visits the page. SizeRec can help provide more information about the customer by tracking conversions from the SizeRec button and gathering customer information from the inputs to SizeRec. More specifically, because SizeRec asks for a customer’s favorite sizes in other brands, we can collect data for a store on which brands the store’s customers generally prefer to wear. This could potentially help the store in positioning itself as more of a competitor to these stores.

Furthermore, the SizeRec button currently has a particular design and style that is un-
changeable. However, each Shopify store, and each online retail store in general, has its own theme and feel. Therefore, we can work in the future to make the SizeRec button and modal more customizable to integrate with a particular store’s theme. This would make SizeRec more attractive in integrating with any store in question.

Finally, outside of technical details, SizeRec needs future development in terms of sales and marketing. As of now, we have a limited number of customers on free trial via Shopify. More work needs to be done to expand our sales channel in reaching out to other stores as potential customers.

5. Acknowledgments

A special thank you to Professor Ruzica Piskac of the Yale Computer Science Department for advising this project. She has given me suggestions, feedback, and direction through the entirety of this endeavor. Also, a thank you to Minh Tri Pham, who put in countless hours into SizeRec with me.

6. References


