Problem

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.

Solution

SizeRec aims to solve this problem by recommending a size for customers based on simple user input: sizes customers wear in other brands. SizeRec would be implemented as a button on product pages, i.e. right under the size selector. The user flow would be:

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 (See the Recommendation Methodology section for detailed info about recommendation mechanisms)

   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, e.g. Shopify or Magento. 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. If the testing period is successful, we will implement SizeRec as a universal JavaScript snippet, which retailers can inject onto their e-commerce website. This JavaScript implementation will eliminate platform dependency.

**Recommendation Methodology**

**Sizing Chart Interpolation**

We will collect size charts from 100 most popular fashion brands and store them in a database. Given sizes for brands A, B, and C, SizeRec can recommend a size for a brand D by performing a table lookup, which will return body measurements of the customer. Given the body measurements, SizeRec can look at the size chart of brand D to recommend a size. Although this is a very rudimentary recommendation methodology, it will suffice in increasing customer confidence in sizing.

**Next Steps**
After we have proven that SizeRec delivers value to retailers, we will allow them to collect additional user data, such as weight, height, and body type, alongside with sizes and brands. We can use this data to train a machine learning model that will recommend sizes in a more sophisticated way. The input will be the weight, height, body type, and the labels are the sizes the customer wears in other brands.

**Deliverables (Chris)**

I will be responsible for the whole backend portion of SizeRec, which includes the following:

- Data collection, organization, and storage
- Building the recommendation system
- (maybe later) Training a machine learning model that recommends size based on additional user data

**Data Collection**

This is a difficult task because there are hundreds of brands with various sizing charts in both text and image format. In addition, it is perhaps infeasible to collect the sizes of all clothing items at all stores. Indeed, in order to make data collection as efficient as possible, I will do market research on the most popular clothing items worn by men and women, and I will do market research on the 50 or 100 popular clothing brands. Although this may still amount to many different sizing charts to pull from the web, one possible solution is to use Amazon Mechanical Turk to make the process of data collection as quick and reliable as possible. Another solution may be to build a web scraper that
visits the sizing charts of various brands online, takes screenshots, and stores them in an online storage bucket.

Sizing data for any given item of clothing from different brands also varies. For instance, some brands will include chest size, neck size, and waist size for their men’s shirts while other brands will include other variables. In the collection of data, I will look for common variables across brands and only include those parameters in my database. Among these common variables, I will need to decide which variables are most important to fit. In other words, I will classify primary, secondary, and tertiary variables in order of importance to fit. To do this, I will conduct market research or build a classifier in Python.

Once I have decided on these factors, I will make various tables in my database corresponding to clothing item. For instance, I may have a Men’s T-shirt table, or a Women’s Jeans table. The database will be served using a PostGres SQL database.

**Recommendation System**

As mentioned in the Recommendation Methodology section, one solution to the sizing problem is to map clothing sizes of brands that individuals normally wear to a size of the brand in question. A naive implementation of this solution would be to simply average the dimensions of the clothing that individuals normally wear and find the closest fit to the brand in question. However, a more sophisticated solution would be to find intersections in the dimensions of clothing that the customer normally wears and then to map those intersections to a size in the brand in question. We can start with the primary variable, and if the intersection straddles two different sizes of the brand in question, we can move on to the secondary variable, and then to the tertiary variable. The
recommender could then give a confidence level of the “trueness” of the fit, depending on the intersections and how well we are able to find a size for the brand in question that matches.

**Machine Learning**

Another solution to the sizing problem requires ample data and could be implemented once there is a wide customer base. As mentioned in the recommendation methodology section, I could train a machine-learning algorithm, perhaps using the Keras library in Python. However, this solution is dependent on the availability of user data and must necessarily come after having gone to market sufficiently long enough to have acquired the data.