Team Project - Clothing Recommendation

Maria Stella Vardanega, Risheng Guo, Ting-Hsuan Chen, Jiazhi Jia, Kratik Gupta

1. Business Problem

In recent years, with emphasis during the pandemic, e-commerce has increased dramatically. During the lockdown, consumers who wanted to go shopping were forced to do it online in most countries. It is difficult for users to not only find something they like on online platforms but also to find the correct sizing and fit. It's more likely that they will order the wrong size and have to return the item. Therefore, the company would have to handle returns, which will increase costs and decrease revenue. Moreover, as more people get vaccinated and more places remove their quarantine policies, the demand for attending social events is increasing dramatically.

Through the project, we hope to analyze the different attributes of a person and associate them to a particular item and size in order to subsequently give the consumers recommendations on size that are potentially more accurate than the consumer attempting to make an educated guess. In order to also help consumers find items that they might also like, we want to build a recommendation system that recommends clothing items according to interests and items they are already interested in. We believe our recommendation can assist users to better find what they need and thus make them order more, which will generate higher revenue.

2. Dataset

The dataset is from Kaggle provided by Rent The Runway:
https://www.kaggle.com/rmisra/clothing-fit-dataset-for-size-recommendation?select=modcloth_final_data.json

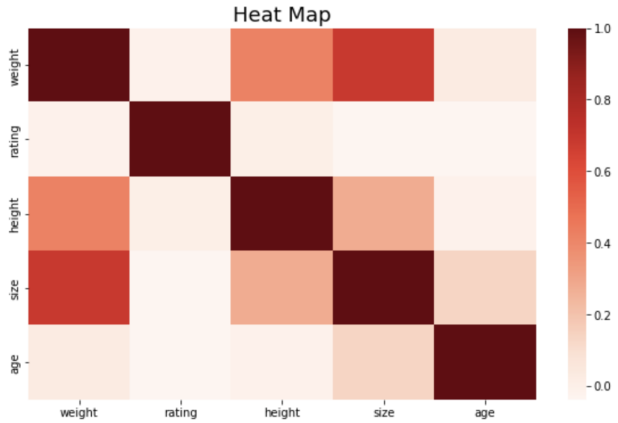
Our analysis is focusing on the following features (columns) in the dataset:

- Fit - fit feedback
- User_id - a unique id for each user
- bust size - user's bust size
- item id - a unique id for each product
- weight - user's weight
- rating - product rating given by user
- rented for - user's using purpose for the product
- review_text - user's review
- body type - user's body type
- review summary - the summary of user's review

- category - product's category
- height - user's height
- size - product's size
- age - user's age
- review_date - review published date

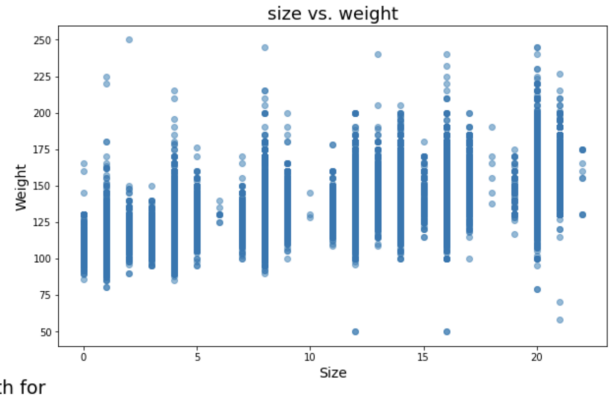
We filled all the null values by mode and mean, and converted all data to the relevant data type. After cleaning, the dataset contains 192544 rows and 15 columns. Then we performed exploratory data analysis on the cleaned dataset and we also did sentiment analysis on the 'review_summary' column.

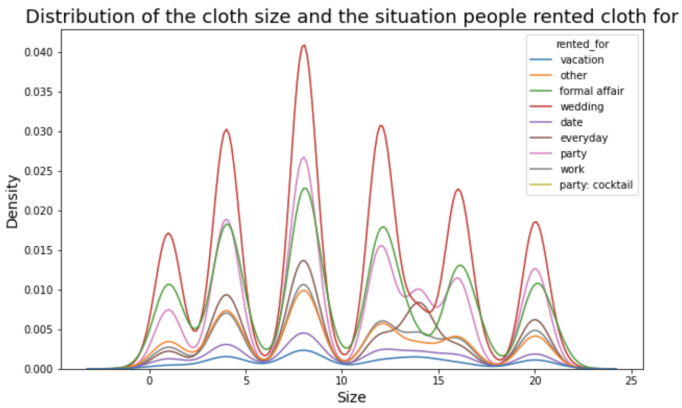
3. Exploratory Data Analysis



When exploring the data, we first created a heatmap to help study the correlation between each factor. We observed that the size of clothes is correlated with the user's weight and height. Especially weight, it has a significant positive correlation with the size of the clothes. Thus, we want to compare the user's weight and the clothing size they chose.

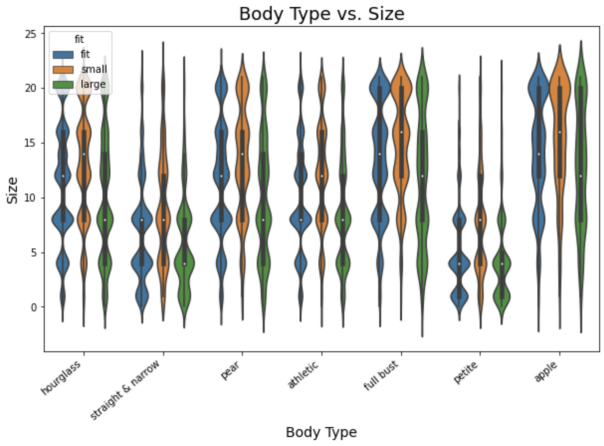
Illustrated by the graph on the right, the x-axis represents the size and the y-axis represents the weight, we see there is a clear positive correlation, which makes sense since people who have more weight usually buy larger clothes.

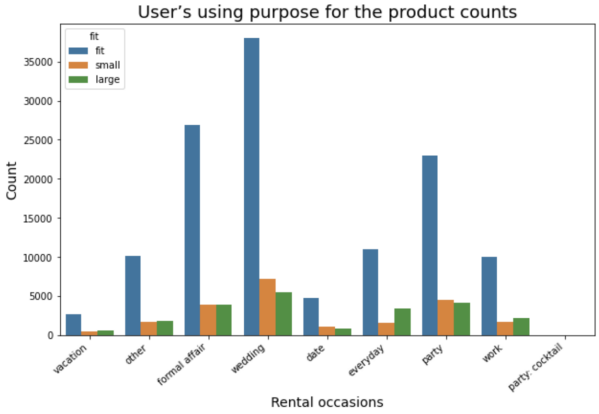




The graph on the left shows the distribution of the cloth size and the situation people rented cloth for. We can notice that most sizes fall into the range between 3 and 18 for all purposes.

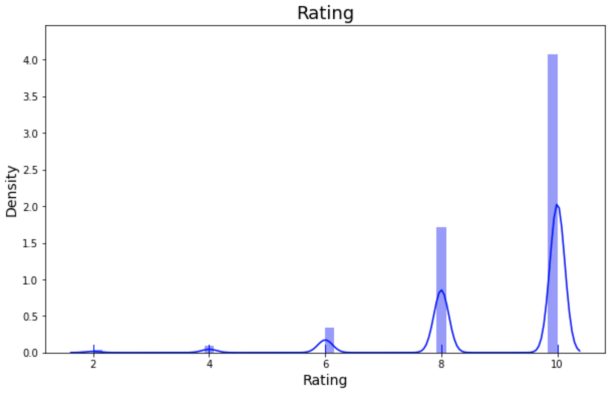
We are also curious about whether people's body types affect the cloth size they chose. Based on the plot, we can conclude that most people in this dataset have a body type of full bust and apple. Except for people with straight & narrow and petite body types who will choose a smaller size, there is not much difference in the clothing size between other body types.





We now know people's preferences when choosing their size. But do they get what they want? From the graph on the right, we can see that most people rent clothes for weddings and formal affairs. Across all the renting purposes, most people found the size that fits their body. So it makes data quite biased on fit, but it can be an actual representative of the sample.

The graph on the right shows the distribution of people's ratings on their purchases. It is clear that there are many more high ratings than low ratings. 10, which is the highest rating, also appears the most in the rating column. Thus, the last two graphs can prove the results we get at the beginning are valid.



4. The Methods

4.1 Text analysis on reviews

In this part, we want to figure out the reason why people are more likely to give a relatively high (or pretty low) score. First, we utilized the afinn package in python to get the sentiment scores for the ‘summary_reviews’ column and made a plot illustrating the distribution of scores. Then, we used TF-IDF vectorizer to fit and transform the customer's review with scores greater than 5 and scores smaller than 0 and plotted the distribution of every bigram review. Lastly, we also made word clouds to show what customers are saying in their renting experiences.

4.2 Cluster analysis

Our goal is to segment our sizing data into like-groups. In this part, we will first try out two clustering methods, which are Hierarchical Clustering and K-Means Clustering.

Before we conduct the hierarchical clustering analysis, we first apply one-hot encoding to our categorical variables. We use the get_dummies function to create the dummy ‘bust_size’ and ‘body_type’ variables for them to be included in the clustering. Subsequently, we scaled the features because the values in the columns were on different scales, for example, height and weight. Lastly, we conducted the Dimension Reduction by using PCA, dropping the least important variables while at the same time retaining the key information. For Hierarchical Clustering, We first tried out Euclidean and Cosine distance metrics then tried four linkage methods, including "single", "complete", "average", and "ward" linkage.

Before conducting the K-Means Clustering, we first dummified "rented_for" and "category" variables. Subsequently, we standardized our data. After rescaling our features, they became much easier to compare and each feature has a similar effect on our clustering algorithm. Then, Dimension Reduction was conducted by using UMAP. In order to ensure the reviews are taken into account while clustering the item_id, we used the text vectorizer to vectorize our reviews. For K-Means Clustering, we calculated and recorded the inertia and the average silhouette score.

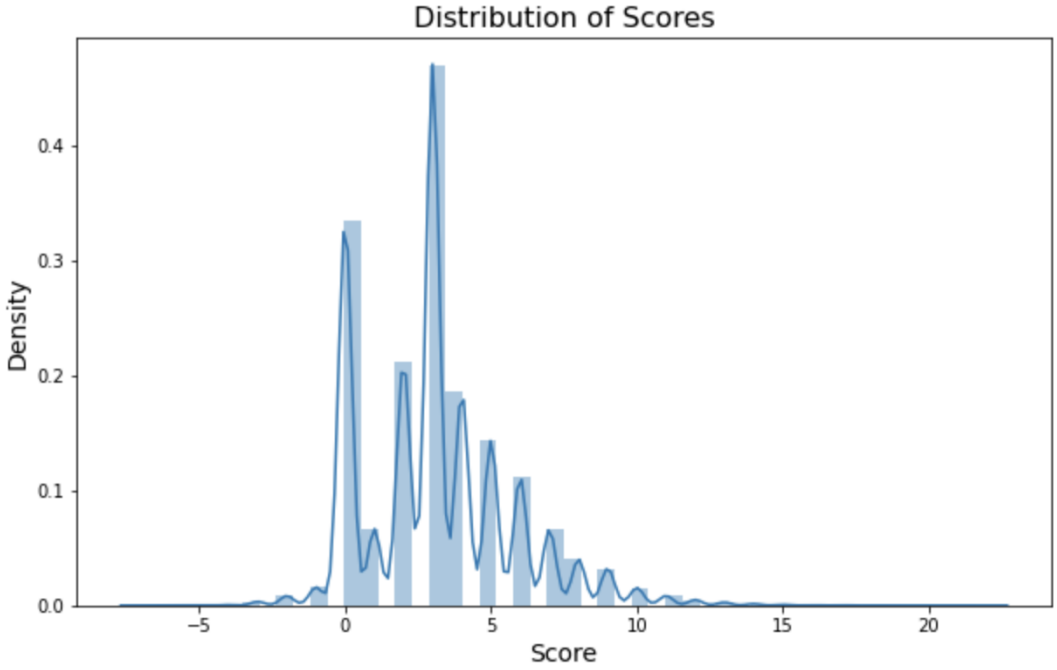
4.3 Clothing Item and Size recommendation

In this part, we first used K-Nearest Neighbors to recommend the items. Then, we utilized the Random Forest Regressor to find size recommendations.

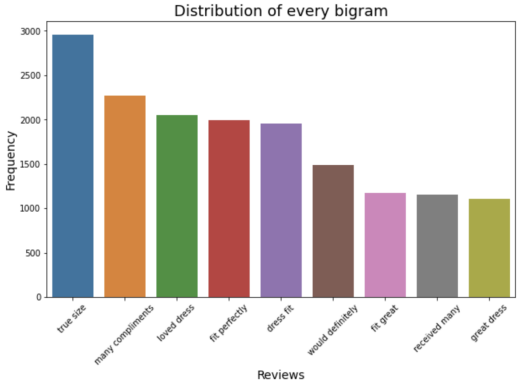
5. Analytical findings

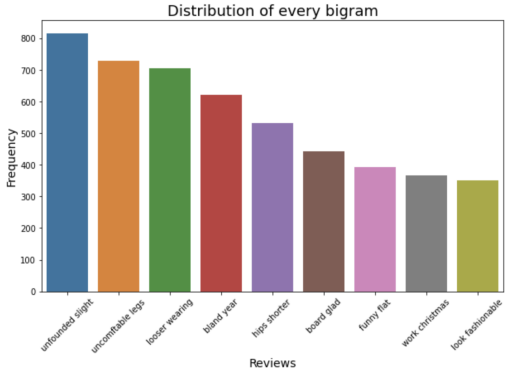
5.1 Analytical Finding - Text Analysis

After using Afinn, we found out that most people tend to give a moderate review (afinn score between 0 to 5) to the clothes they rent. But in some cases, customers would like to give harsh comments on their renting experience. There are also some situations where people praise the clothes. Therefore our goal is to find out why some people would criticize or praise after receiving their clothes and improve the quality of service.



By using TF-IDF vectorizer to fit and transform the review text with a high (or low) score, we can get the frequency distribution of every bigram. We think that the most frequent words among the review text can represent the motivations of customers to leave their comments.

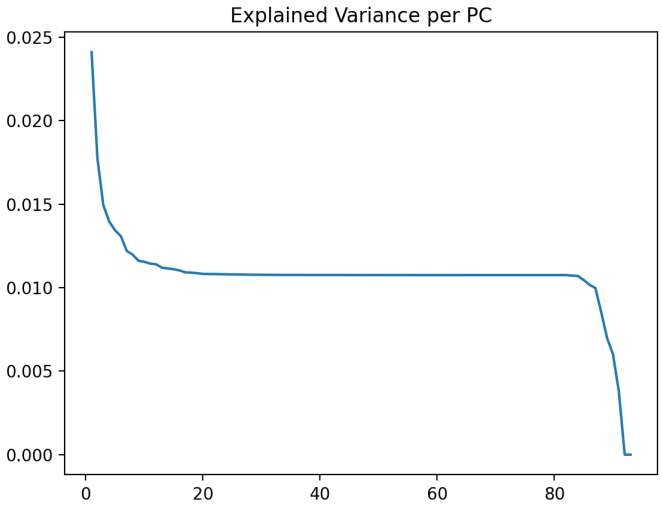




From the graph above we can see that the most important thing for a customer is whether that clothes fit them well. So it's necessary to conduct a prediction on the size of clothes before new customers place orders.

5.2 Analytical Findings - Cluster Analysis

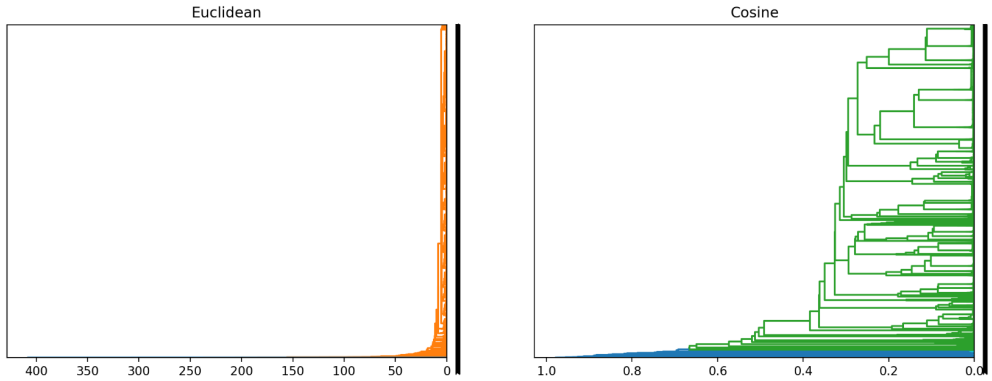
After using PCA to conduct Dimension reduction. We can see from below that the graph illustrates the explained variance per PCA component. We can see from the graph that with approximately the first 80 components we can explain the majority of the variance.



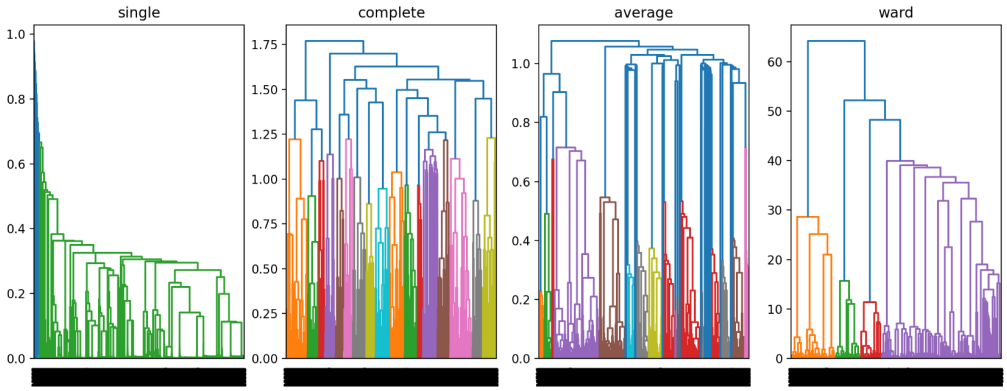
Because the dataset was very large (over 150,000 observations), a sample of 25,000 rows was taken. This number was determined by testing how long it would take to run code with various dataset sizes. 25,000 seemed to yield the best cost versus benefit.

5.2.1 Hierarchical clustering

Firstly, dendrograms were drawn according to whether the distance was measured using the Euclidean or Cosine metric. This yielded the figure below.



As can be seen from the figure above the Cosine metric finds more distance between the clusters and therefore seems to be the better option. Thus, this metric was used. Subsequently, different linkage methods were tested (single, complete, average, and ward) and their respective dendrogram was drawn, shown below.



Once again, we can see that the complete method seems to find better separation between the clusters compared to the other linkage methods. Therefore, the clusters found using the Cosine distance metric and Complete linkage method were profiled. The data was grouped by clusters and the rounded average age, size, height, weight , and body type were found for each cluster. The results are displayed in the table below.

Cluster	Size	Age	Weight (lb)	Height (in)	Body Type
1	14	38	135	64	Hourglass
2	16	33	140	64	Hourglass
3	12	40	172	68	Full Bust
4	4	32	126	66	Hourglass
5	10	34	133	65	Hourglass

6	9	32	140	67	Hourglass
7	20	34	165	61	Full Bust
8	12	36	130	62	Hourglass
9	8	36	117	64	Athletic

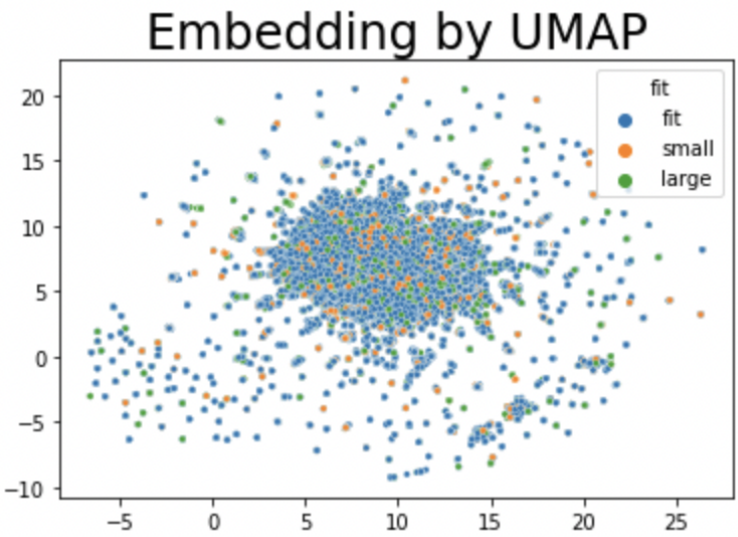
5.2.2 KMeans Clustering

Text Vectorization

To ensure that reviews are taken into account while clustering the item_id, we decided to vectorize the review_summary column. Using NLP with help of Spacy library and "en_core_web_md" for vectorizing as our reviews were in english language. We got an array with vectorized text which we utilized in our further analysis.

UMAP for Dimensionality Reduction

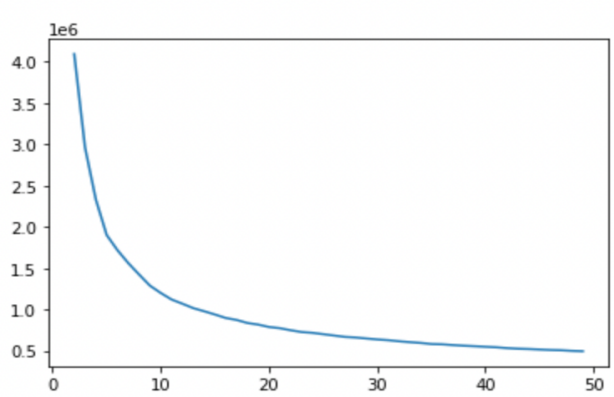
To process the data faster and to reduce features with high correlation we used Uniform Manifold Approximation and Projection (UMAP) for Dimension Reduction. Using the dummified data and text vectors we used UMAP to reduce our number of features from 474 to a default of 2.

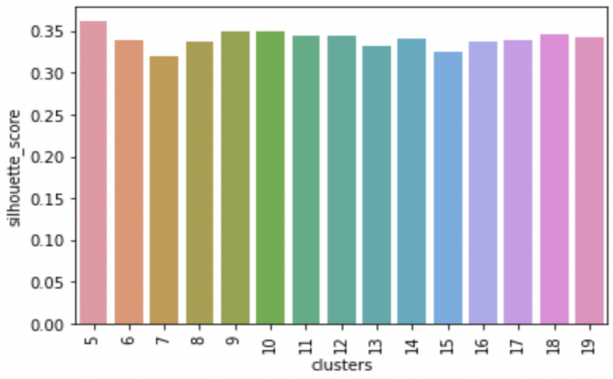


KMeans Clustering

For KMeans, the first objective was to find the optimal value of K, for that we ran multiple numbers of clusters and plotted the inertia and silhouette score of them. To ensure the process is fast we sampled the data for plotting clusters and silhouette scores. We used values for k in range

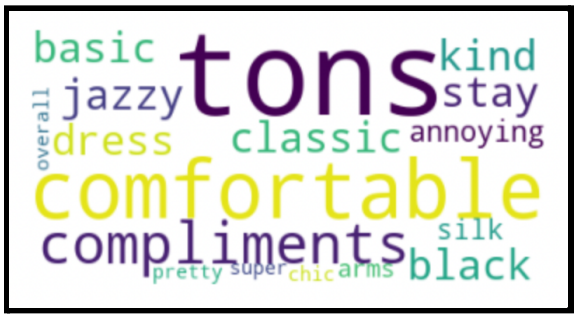
2 to 50 and found by elbow method that 5 clusters will yield best possible results with silhouette score of approximately 0.36.

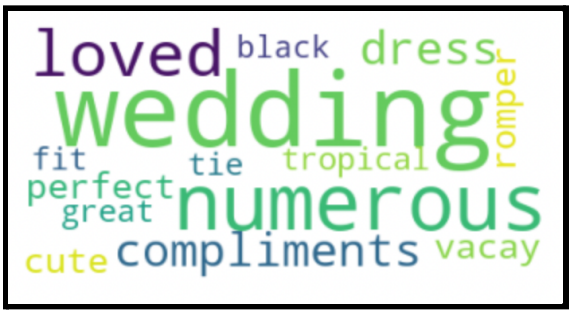




Word Cloud for Different Clusters

We used keyBERT to make a word ranking system from the review_summary column of every cluster. In such a way that most frequent words are not the only words represented by the word cloud. We got more optimal word clouds. As the word clouds shows, there are similar patterns in the reviews of different clusters.





5.3 Analytical Findings - Clothing Item and Size recommendation

Using K-Nearest Neighbors to find Item Recommendations

We used the K-Nearest Neighbours algorithm to get the recommendations on item_id. This is meant to be implemented with Relational Database. Where we will have more information about what item_id is representing. Assuming each item_id is linked to a particular dress we can provide recommendations if a user is interested in a particular item_id, using this model we can provide him/her suggestions regarding which item they might be interested in. E.g. for item_id 1795593 based on certain parameters our model will recommend 5 "item_id"s which are highly related to the one user is interested in.

```
# getting distances and suggestions based on id
distances, suggestions = knn.kneighbors(data_txt.iloc[12345, :].values.reshape(1,-1), n_neighbors = 6)
```

Python

```
data[data.index == suggestions[0,0]]
```

Python

	fit	user_id	bust_size	item_id	weight	rating	rented_for	review_text	body_type	review_summary	category	height	size	age	review_date
12345	fit	361493	34c	1795593	107.0	10.0	vacation	From the moment I saw this dress, I was in lov...	hourglass	In a word: Perfection!	maxi	61.0	4	31.0	2017-06-20

```
data[data.index == suggestions[0,4]]
```

Python

	fit	user_id	bust_size	item_id	weight	rating	rented_for	review_text	body_type	review_summary	category	height	size	age	review_date
44347	fit	953990	34a	1974220	108.0	10.0	wedding	Definitely need fashion tape to keep dress fro...	hourglass	Dress was beautiful! Received many compliments...	gown	61.0	3	30.0	2017-01-22

Using Random Forest Regressor to find Size Recommendations

We used a Random Forest Regressor to predict the sizes that best fit a user according to various measurements and attributes of this user. A function was also created that would request information from the user to then predict which size would best fit. The needed information is the age, height, and weight of the user.

```
[ ]  1 def size_predict():
     2     age = float(input("age = "))
     3     weight = float(input("weight = "))
     4     height = float(input("height = "))
     5     return rfc.predict(np.array([height,weight,age]).reshape(1,3))
```

6. Conclusions and Recommendations

When we first analyzed the reviews given by the users, we found that the majority of the reviews were of moderate sentiment. Therefore, they were neither extremely happy nor extremely unhappy about their clothing. However, when taking a closer look at the reasons why users gave negative reviews we discovered that fit played the most important role in the customer satisfaction; therefore, we wanted to focus on the size recommendation.

After running the hierarchical clustering analysis we decided to use the complete method with cosine distance as it gave the clearest clusters. From there, we proceeded to separate the data into 9 separate clusters. After profiling we can see that the major differences between the clusters can be attributed to the age, weight, and height of the consumer. Thus, these measurements were the ones used for the size prediction.

Lastly, we wanted to expand on the consumer satisfaction potential and make a recommendation system for similar clothing items the consumers might like. We decided to cluster according to the Item id. This process can be expanded by including another dataset that matches the item id to a photo of the item or various other attributes of the item.

After analyzing our results we saw that Rent The Runway has various possibilities to increase customer satisfaction, these may include:

- Improving the sizing recommendation system for their customers by collecting more data (e.g. their sizes in other stores, various body measurements).
- Possibly associating certain clothing item attributes with their fit (e.g. they fit smaller, normal, or larger than average clothing of that sizes) to better inform their customers.
- Clearly display the clothing item fit on the website to make sure consumers are aware before renting the product.