AI-Powered Size Chart Generator for Apparel Sellers

Overview

With the rapid rise of e-commerce, particularly during the pandemic, one of the biggest challenges for online apparel sellers has been helping customers select the correct size. The issue often results in customers ordering incorrect sizes, leading to returns and increased costs for businesses. This project aims to solve this problem by developing an AI-powered system that generates accurate size charts for apparel sellers, helping customers choose the right fit and ultimately reducing returns and increasing sales.

The Problem

Customers often struggle to find the correct clothing size when shopping online due to variations in sizing across different brands and product types. This can lead to dissatisfaction, increased returns, and higher costs for apparel sellers. This project aims to analyze body measurements, purchase history, and return/exchange data to create a robust AI system that can generate accurate size charts and provide size recommendations.

The Solution

Our AI-powered size chart generator is designed to:

Utilize a comprehensive database of user body measurements (height, weight, bust/chest, waist, hips, etc.).
Analyze previous purchase history and return/exchange data to identify patterns in successful size selections.
Cluster users by body type and their corresponding successful size selections.
Generate size charts for different apparel categories (tops, bottoms, dresses) that accurately map users to sizes like S, M, L, and XL.
Provide confidence scores for each generated size chart, helping sellers and customers make informed decisions.
Allow for easy updates as new purchase and size data becomes available.

Data Source

We are using a dataset from Rent the Runway, which includes:

192,544 rows and 15 features
Body measurements, age, body type, and the sizes users ordered
Insights into what sizes worked for different body types
This dataset allows us to cluster users based on body measurements and generate size recommendations that are personalized for each user.

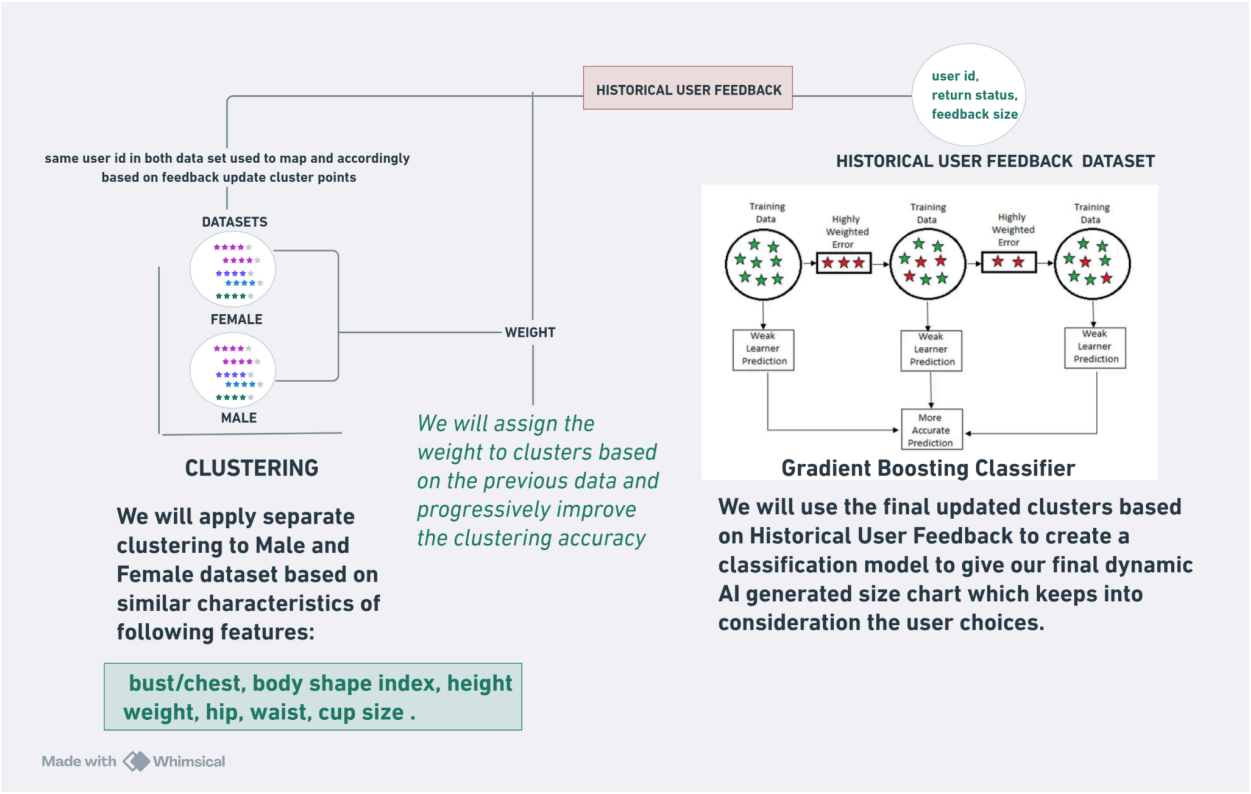
Model Features

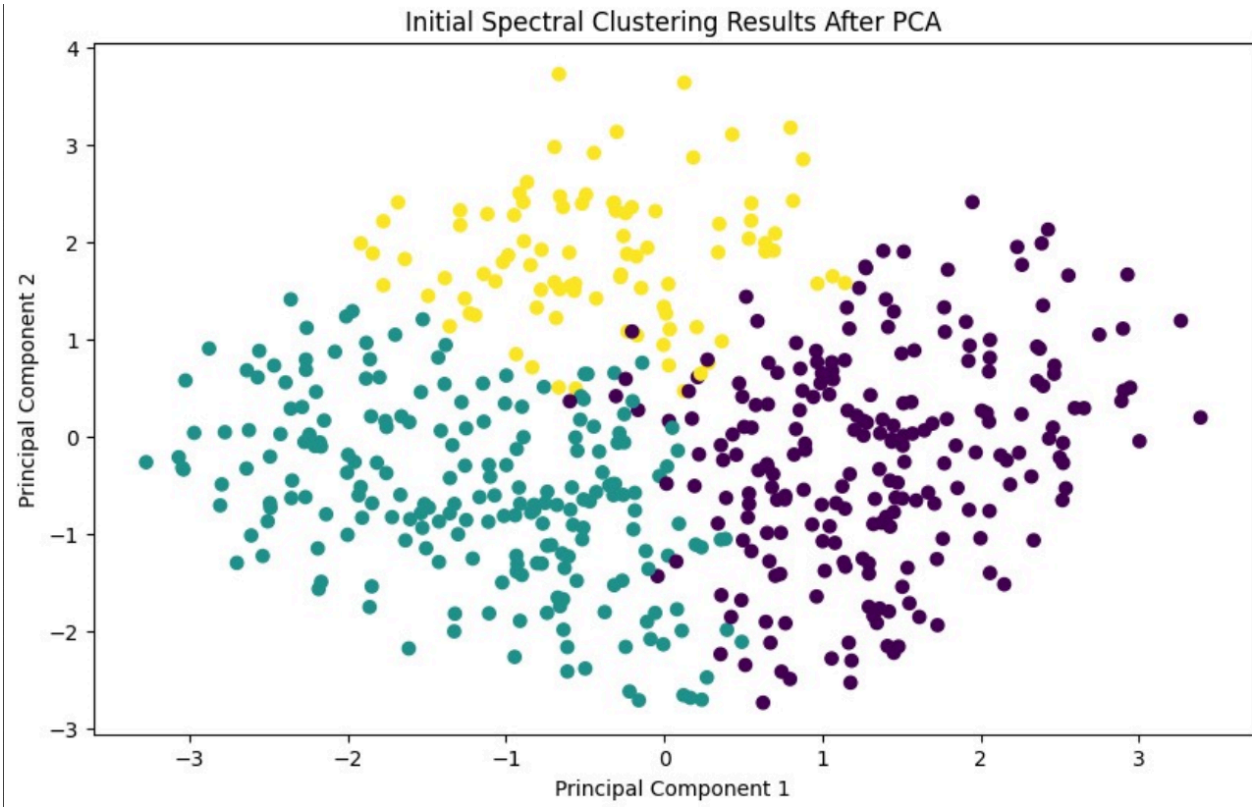
Body Measurement Integration: The model incorporates key body metrics such as height, weight, chest, waist, and hips.
Purchase History Analysis: By analyzing past purchases and returns, the model identifies patterns that lead to accurate size recommendations.

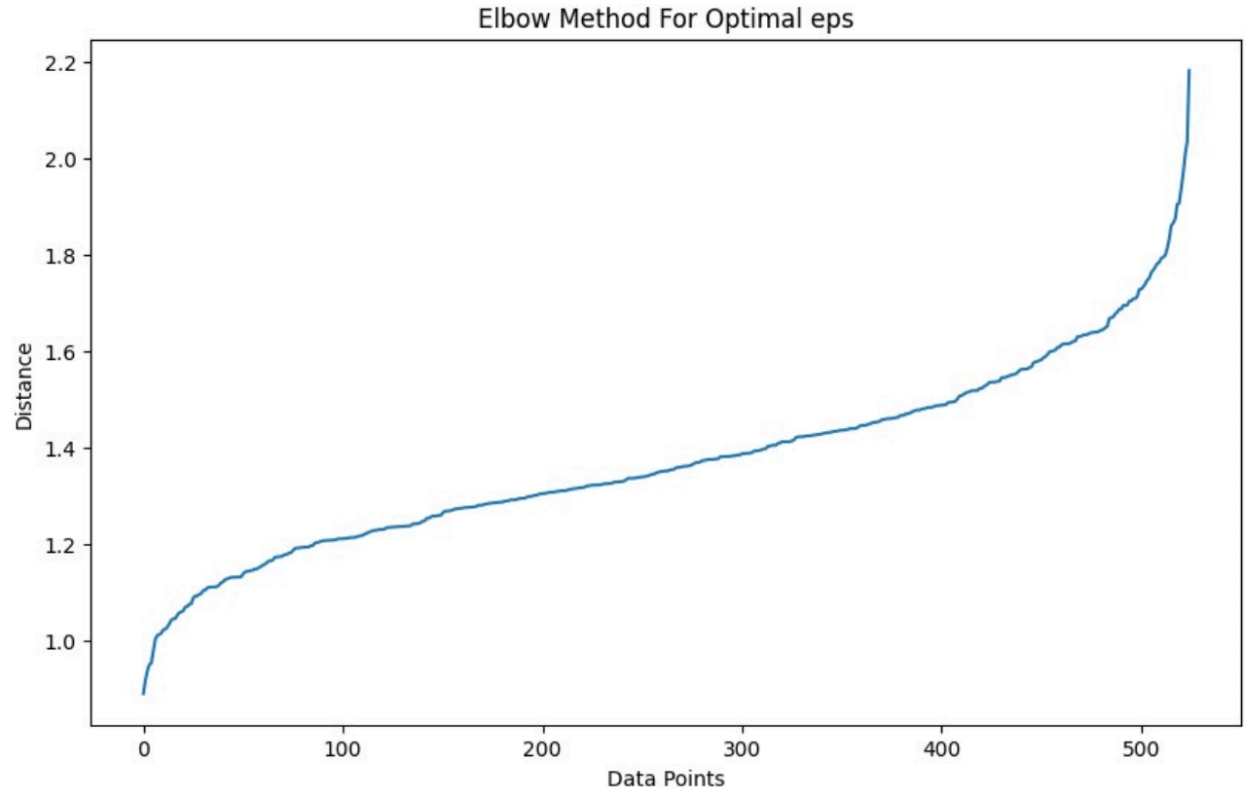
Dynamic Size Chart Generation: The system dynamically generates size charts for various apparel types, providing sellers with precise sizing information.
Confidence Scores: Each size chart comes with confidence scores, indicating the likelihood of accuracy for different body types.
Scalable & Adaptive: The model is scalable, capable of handling new brands or product lines, and adaptable to new data as it becomes available.
Judging Criteria
The model's effectiveness will be evaluated based on:

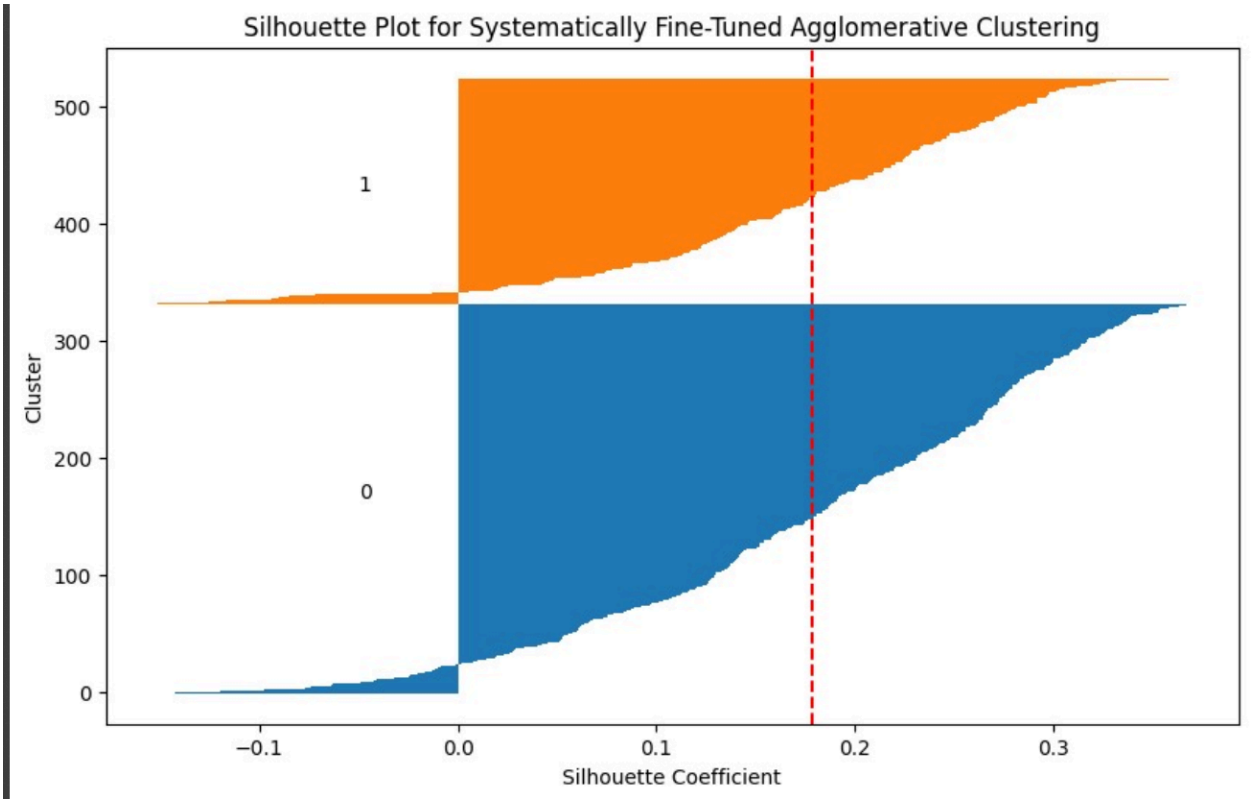
Accuracy: How closely the generated size charts match known accurate size charts from brands.
Category Handling: The ability to handle a variety of apparel categories.
Return Reduction: Effectiveness in reducing size-related returns, tested using simulated holdout data.
Scalability: Ability to adapt to new brands or product lines without significant modification.
Efficiency: Processing speed and resource efficiency in generating and updating size charts.
Future Work
As the system evolves, we aim to integrate:

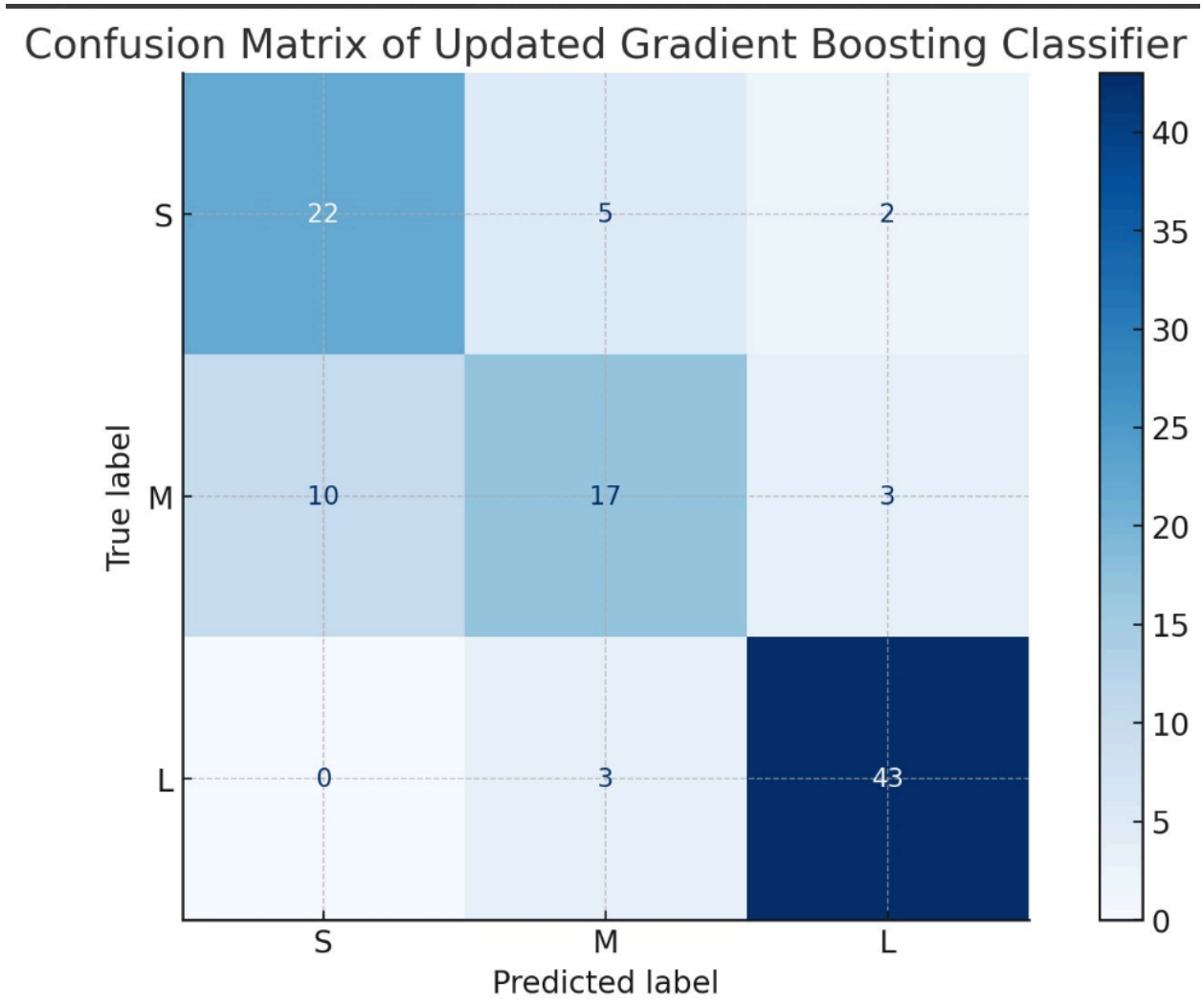
Social event recommendations: Suggesting items that fit both the user's body type and social occasions.
Real-time updates: Continuous improvements as new purchase data becomes available.

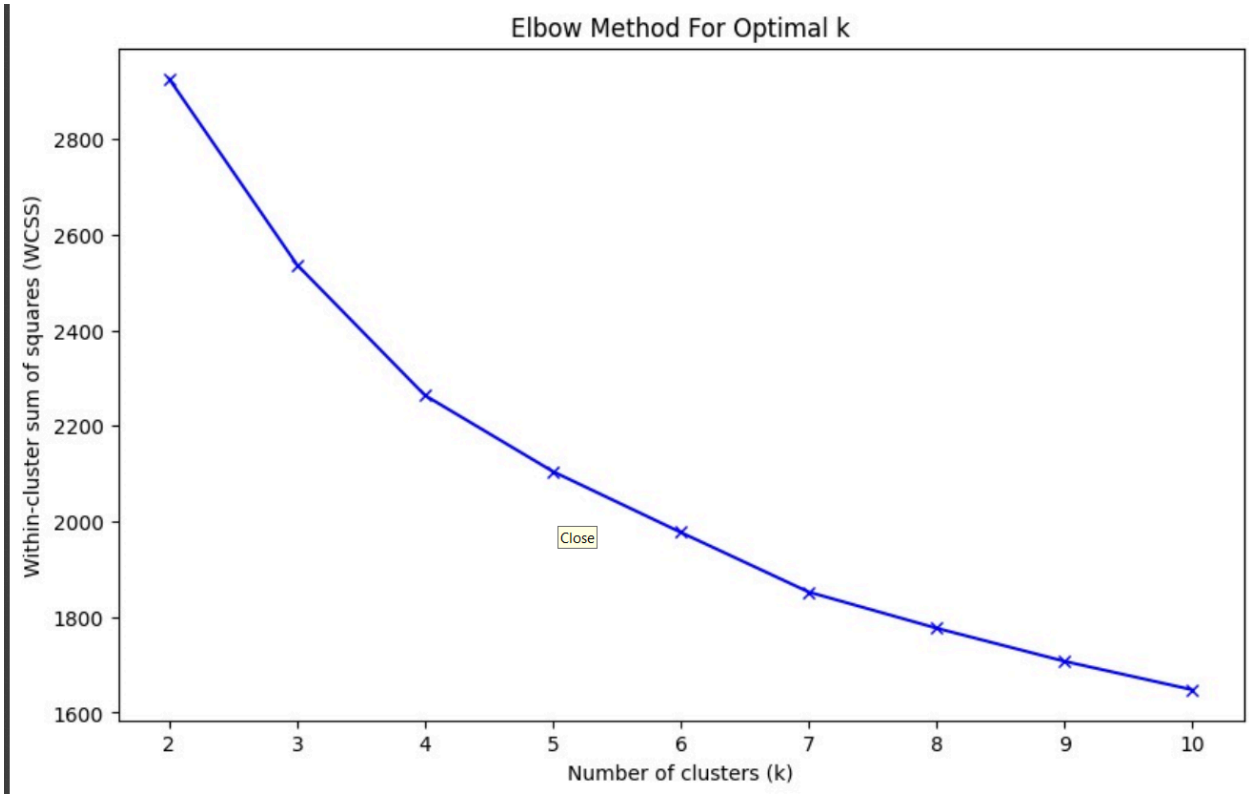

HISTORICAL USER FEEDBACK
user id,
return status,
feedback size
HISTORICAL USER FEEDBACK DATASET
same user id in both data set used to map and accordingly based on feedback update cluster points
DATASETS
FEMALE
MALE
WEIGHT
CLUSTERING
We will assign the weight to clusters based on the previous data and progressively improve the clustering accuracy
Training Data
Highly Weighted Error
Weak Learner Prediction
More Accurate Prediction
Gradient Boosting Classifier
We will apply separate clustering to Male and Female dataset based on similar characteristics of following features:
bust/chest, body shape index, height weight, hip, waist, cup size .
We will use the final updated clusters based on Historical User Feedback to create a classification model to give our final dynamic AI generated size chart which keeps into consideration the user choices.
Made with Whimsical


Initial Spectral Clustering Results After PCA
Principal Component 2
Principal Component 1
4
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

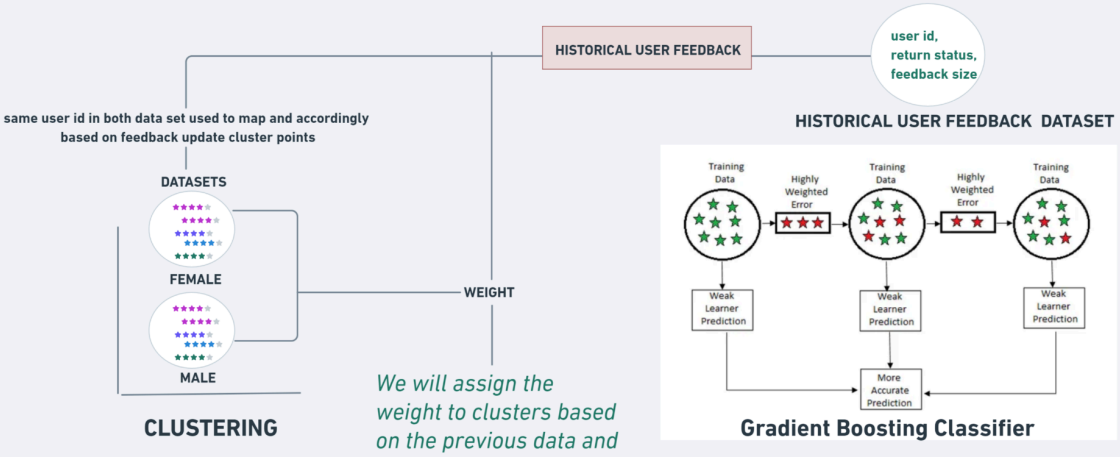

Elbow Method For Optimal eps
Distance
Data Points
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0
100
200
300
400
500


Silhouette Plot for Systematically Fine-Tuned Agglomerative Clustering
Cluster
Silhouette Coefficient
1
0
500
400
300
200
100
0
−0.1
0.0
0.1
0.2
0.3


Confusion Matrix of Updated Gradient Boosting Classifier
True label
Predicted label
S
M
L
22
5
2
10
17
3
0
3
43
40
35
30
25
20
15
10
5
0


Elbow Method For Optimal k
Within-cluster sum of squares (WCSS)
2800
2600
2400
2200
2000
1800
1600
Close
2
3
4
5
6
7
8
9
10
Number of clusters (k)

```python
from sklearn.cluster import AgglomerativeClustering
from sklearn.metrics import silhouette_score, davies_bouldin_score, calinski_harabasz_score
from sklearn.model_selection import ParameterGrid
import matplotlib.pyplot as plt

# Expanded Parameter Grid Search with all linkage methods and a wider range of clusters
param_grid = {
    'n_clusters': range(2, 7),  # Exploring clusters from 2 to 6
    'linkage': ['ward', 'complete', 'average']  # All available linkage methods except 'single'
}

best_silhouette = -1
best_params = {}
best_model = None
best_db_score = float('inf')
best_ch_score = 0

for params in ParameterGrid(param_grid):
    model = AgglomerativeClustering(n_clusters=params['n_clusters'], linkage=params['linkage'])
    labels = model.fit_predict(women_df_pca)

    silhouette_avg = silhouette_score(women_df_pca, labels)
    db_score = davies_bouldin_score(women_df_pca, labels)
    ch_score = calinski_harabasz_score(women_df_pca, labels)

    # Combining metrics to find the best clustering
    if silhouette_avg > best_silhouette and db_score < best_db_score:
        best_silhouette = silhouette_avg
        best_db_score = db_score
        best_ch_score = ch_score
        best_params = params
        best_model = model

print(f"Best Parameters after systematic fine-tuning: {best_params}")
print(f"Best Silhouette Score: {best_silhouette}")
print(f"Best Davies-Bouldin Score: {best_db_score}")
print(f"Best Calinski-Harabasz Score: {best_ch_score}")

# Visualize the final clustering
plt.figure(figsize=(10, 6))
plt.scatter(women_df_pca[:, 0], women_df_pca[:, 1], c=best_model.labels_, cmap='viridis')
plt.title('Systematically Fine-Tuned Agglomerative Clustering Results After PCA')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.show()

# Generate and visualize the updated Silhouette Plot
plt.figure(figsize=(10, 6))
silhouette_vals = silhouette_samples(women_df_pca, best_model.labels_)
y_ticks = []
y_lower, y_upper = 0, 0
```



CLUSTERING

We will apply separate clustering to Male and Female dataset based on similar characteristics of following features:

bust/chest, body shape index, height weight, hip, waist, cup size .

We will assign the weight to clusters based on the previous data and progressively improve the clustering accuracy

We will use the final updated clusters based on Historical User Feedback to create a classification model to give our final dynamic AI generated size chart which keeps into consideration the user choices.

RECOMMENDATIONS

- **AI-Driven Size Recommendations:** Use AI to suggest accurate sizes based on user data, integrated into product pages.
- **Smart Filters:** Offer size recommendations through filters that analyze user measurements for each apparel category.
- **AR Virtual Try-On:** Implement AR try-on to visualize fit using AI-generated size data, minimizing returns.

FUTURE WORKS

- **User Feedback Loop:** Implement a feedback mechanism where users can provide input on the fit of their purchases, allowing the AI system to continuously learn and improve its recommendations.
- **Using Gemini API:** Optimizing the User Feedback Loop so that we can decode every aspect of the user feedback using the Gemini API as it is an Large Language Model (LLM) Machine helping us to achieve the same.

TABLE OF CONTENTS

clustering

	Silhouette Score: 0.16	Davies-Bouldin Score: 1.7	Calinski-Harabasz Score: Not available
Kmean-			
DBSCAN	0.179	4.88	Na
Agglomerative	0.20	0.62	2.4
Gausian mixture model	0.17	1.62	124.7
Specteral	0.17	1.60	: 123.1

- Clustering: Users' body measurements are clustered into distinct size categories (e.g., S, M, L) and mapped to corresponding size labels.
- Refinement: Purchase history, return, and exchange data are used to dynamically adjust and refine clusters, improving size accuracy over time.
- Iterative Updates: The system continuously updates size charts based on new data and feedback, reducing the likelihood of returns.
- Classification Model: A model is trained on the refined clusters to accurately predict clothing sizes for future users, minimizing size-related returns.
- Outcome: The approach enhances customer satisfaction and business growth by ensuring accurate size recommendations and reducing return rates.

Gradient Boosting Classifier

ESTIMATOR: 100

LEARNING RATE: 0.03

MAX DEPTH: 3

ACCURACY: 78%

MIN SMAPLE SPLIT: 2

- Comprehensive Clustering Methodology
- Algorithms Used: K-Means, DBSCAN, Agglomerative, Spectral Clustering, GMM
- Optimization of Performance Metrics: Silhouette Score, Davies-Bouldin Score, Calinski-Harabasz Scor
- Advanced Techniques:
  - PCA for Dimensionality Reduction
  - Grid Search & Bayesian Optimization for Hyperparameter Tuning
- Key Tools: Python's scikit-learn & Skopt