SMARTSIZE AI – MACHINE LEARNING BASED CLOTHING SIZE RECOMMENDER

A Project Report
Submitted in partial fulfillment of the
requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

DEPARTMENT OF COMPUTER SCIENCE and Engineering

By

HARSHINI S. R- 2420030113

GIRIDHAR – 2420030171

SATYADEV – 2420090120

Under the Esteemed Guidance of

LECTURER NAME
Dr. Siripuri Divya

(DEEMED TO BE UNIVERSITY)

Koneru Lakshmaiah Education Foundation

(Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Off-Campus: Bachupally-Gandimaisamma Road, Bowrampet, Hyderabad, Telangana - 500 043.
Phone No: 7815926816, www.klh.edu.in

K L (Deemed to be) University

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Declaration

The Project Report entitled "SMARTSIZE AI" is a record of bona fide work of **team members, 2420030113-Harshini S. R, 2420030171 – Giridhar, 2420090120 - Satyadev,** submitted in partial fulfillment for the award of B. Tech in Computer Engineering to the K L University. The results embodied in this report have not been copied from any other department/University,/Institute.

2420030113 - Harshini

2420030171 - Giridhar

2420090120 - Satyadev

K L (Deemed to be) University

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

This is certified that the project-based report entitled "SMARTSIZE AI" is a bonafide work done and submitted by **Harshini (2420030113), Giridhar (2420030171), Satyadev (2420090120)** in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in Department of Computer Science Engineering, K L (Deemed to be University), during the academic year **2024-2025.**

Signature of the Supervisor

Signature of the HOD **Signature of the External Examiner**

ACKNOWLEDGEMENT

The success in this project would not have been possible without the timely help and guidance rendered by many people. We wish to express our sincere thanks to all those who have assisted us in one way or the other for the completion of my project.

Our greatest appreciation to our guide, Dr. Divya Siripuri, Department of Computer Science, who cannot be expressed in words for her tremendous support, encouragement and guidance for this project.

We express our gratitude to **Dr. B. Venkateshwara Rao**, Head of the Department for Computer Science Engineering, for providing us with adequate facilities, ways, and means by which we are able to complete this project-based Lab.

We thank all the members of teaching and non-teaching staff, and also those who have assisted me directly or indirectly, for the successful completion of this project. Finally, I sincerely thank my parents, friends, and classmates for their kind help and cooperation during my work.

Harshini - 2420030113

Giridhar - 2420030171

Satyadev - 2420090120

TABLE OF CONTENTS

ABSTRACT

1. Project Overview

This paper presents Smart Size AI as an artificial intelligence-powered sizing platform of the next generation, aiming to solve one of the major problems in the global clothing industry: accurately predicting and recommending the right size across various consumer populations. The system applies machine learning; it combines computer vision and large-scale data analytics to enhance people's fit within online retailing stores or offline ones. The project originates from interdisciplinary research on fashion technology, AI, UX design, data privacy, as well as supply chain optimization.

The world in which almost all aspects of life are nearly dominated by e-commerce and offers a wide market to the international audience makes it imperative for you to look for a remedy that will finally solve the issue of size normalization, cultural fit variations, and the constant evolution of textiles and wearables. Smart Size AI is the fusion of systems-approach design, software engineering development methods, and iterative stakeholder management techniques into building a recommendation engine that is very accurate, scalable, modular, easily upgradable for new markets, and evolving uses system

2. Problem Statement

E-commerce innovations in practice are too often undermined by the inability to recommend proper fits, resulting in high product return rates, low customer satisfaction, unsold inventory, waste, and negative environmental impacts. Classic sizing systems-either static size charts, rule-based algorithms, or customer guesswork-are inadequate in their capacity to accommodate inter-individual variability because of genetics, lifestyle, and evolving fashion preferences.

Poor-fit returns make up approximately 30% of all online apparel purchases globally, causing tremendous losses for retailers and manufacturers, while promoting consumer frustration. This situation worsens in marketplaces catering to multicultural populations with varying anthropometric distributions.

3. Project Motivation and Goals

Smart Size AI is inspired by the vision to reduce the number of misfit returns while enhancing the confidence of online shoppers and making it easier for retailers in inventory management. The objective of this project is three-fold:

- To engineer a sizing platform that would minimize return rates due to improper fits.
- To make e-commerce apparel more inclusive by catering to users across a wide spectrum in terms of body shape, size, and accessibility needs.
- To enable data-driven design and production for apparel manufacturers and brands.
- To ensure user privacy by applying best practices in ethical AI and regulatory compliance.
- To build the flexible base for future expansion with augmented-reality fitting and biometric adaptive clothing.

4. Methodological Foundation

Smart Size AI uses a microservices structure to divide tasks between data intake, modeling, suggestion sending, and analytics display. The backend uses microservice APIs (made with Node.js), an AI inference engine (Python, TensorFlow, scikit-learn), and data storage (MongoDB, Redis). Real-time and batch data streams are synced using task schedulers and data pipelines.

The suggestion system works by combining three sources: user measurements, past purchases, and image-based body extraction using deep learning. Data merging, ensemble learning, dimensionality reduction, and feedback are used to improve prediction accuracy and adjust to changes in user actions and clothing stock.

5. Background

Smart Size AI's design is based on current research. This includes work on convolutional neural networks for analyzing 3D body scans, methods of collaborative filtering like those in TrueFit and Fit Analytics, and machine learning setups that keep data private. To handle problems with limited data and changing data patterns, the platform uses data changes, transfer learning, and specific learning rates. Studies point out that different body types must be included, and data leaks are a risk. These issues are key concerns in our design. Studies emphasize the importance of inclusivity for underrepresented body types and the risks posed by data privacy breaches; both are central concerns addressed in our architecture.

6. System Features & Innovations

Smart Size AI differentiates itself through the following innovations:

- **Demographic Coverage:** The ability to handle multiple age groups, genders, and ethnic backgrounds, using customizable size charts and flexible data representation.

- **Body Shape Analytics:** Utilization of advanced keypoint detection and pose estimation algorithms for high-precision dimension extraction from user images.

- **Privacy-First Design:** End-to-end encryption, strict access controls, and user data minimization are default principles.

- **Retailer Tools:** Granular analytics for sizing trends, inventory optimization, and return risk prediction.
- **Multimodal Input:** Support for manual measurements, image uploads, and soon, smartphone-based AR scanning.

7. Technical Implementation

The backend microservices work in conjunction with a real-time message broker to enable scalable concurrent processing of recommendations. Model serving is containerized with Docker for rapid deployment across test, staging, and production environments. Data preprocessing involves imputation, noise filtering, normalization, and one-hot encoding for categorical variables. Ensemble approaches, random forests, XGBoost, and neural collaborative filtering,form the predictive backbone, while model evaluation leverages k-fold cross-validation.

On the frontend, a React-based user interface supports multi-modal onboarding, adaptive accessibility components, and dynamic visualization of fit recommendations. Integrations with third-party e-commerce APIs enable seamless deployment in retailer contexts.

8. Dataset Design and Augmentation

Smart Size AI draws on multiple dataset sources: synthetic anthropometric data, anonymized retailer databases, public size charts, and real user inputs. Extensive use of data augmentation, rotation, scaling, and synthetic user generation,alleviates overfitting and enhances robustness, especially in privacy-constrained environments.

9. Stakeholder Engagement

The development lifecycle includes continuous feedback from key stakeholders: users (via surveys, feedback forms, AB testing panels), retailers (feature prioritization, analytics needs), and regulatory experts (compliance audits, privacy reviews). Requirement traceability matrices, sprint backlogs, and regular client meetings ensure alignment with commercial and ethical objectives.

10. Challenges and Risk Mitigation

Throughout the development and deployment of Smart Size AI, several primary risks are systematically addressed:

- **Data Privacy:** Compliance with GDPR, CCPA, and local regulations.
- **Bias Mitigation:** Algorithmic audits for demographic fairness and bias correction.
- **Scalability:** Use of elastic cloud infrastructure, automated resource scaling.

- **Robustness:** Multi-layered validation and exception handling for atypical and adversarial input cases.

11. Experimental Validation

A series of validation experiments assesses model performance across major apparel categories (tops, bottoms, dresses, outerwear). Metrics include prediction accuracy, model confidence calibration, reduction in return rates, and user satisfaction rates. Experimental design includes control groups (traditional sizing) and treatment groups (Smart Size AI), with pre-/post-analysis.

12. Impact Analysis

Smart Size AI demonstrates significant improvements in major KPIs:

- Average sizing accuracy increases by 20-30% over standard charts.
- Retailer return rates decrease by 15–35% depending on implementation scope.
- User satisfaction in fit and purchase confidence sees a measurable lift in NPS and repeat engagement.

13. Future Directions

The plan for Smart Size AI includes extending to AR-enabled fitting rooms, real-time fitting visualization, global clothing database profiles, and open APIs to integrate with future virtual retail technologies. Research plans also include zero-knowledge body measurement protocols and research in federated learning to enhance privacy even further.

14. Summary of Contributions

Smart Size AI reflects a synthesis of theory and applied innovation and provides the following principal contributions:

- An AI sizing architecture that is scalable and that uses a modular approach.
- A robust design for data-pipelining, including privacy features.
- Literature review and industry benchmarking with significant coverage and depth.
- A full-stack implementation, including cloud-native functions.
- Open-source modules for extension by community or academic groups.

15. Conclusion

This project introduces a fully integrated strategy to address the issue of sizing apparel and develops solutions based on contemporary artificial intelligence aspects. The system captures real transformational value for consumers, retailers, and the larger sustainability objectives of the industry while showing academic value and commercial viability.

INTRODUCTION

1. Introduction: A Global Apparel Sizing Problem

Apparel sizing presents a difficult problem on both a technical and social level. Current sizing systems are rooted in old surveys and limited standards. As the clothing market becomes increasingly global and online shopping expands, people struggle to find clothes that fit. This results in many returns and hurts confidence in buying clothes online. The clothing business deals with increased costs, ecological problems caused by waste, and a pressing demand for custom and eco-friendly answers.

2. A Solution Overview

Smart Size AI aims to change how clothes sizes are matched to shoppers. It uses computer vision, deep learning, and analytics to process images and measurements for accurate suggestions. Its flexible design allows it to be added to different retail settings, like online stores, smart mirrors, and mobile apps. The system is built for quick updates, testing, and customized features for different customers.
Its core features include:

- Precise measurement: Algorithms pull exact measurements from user images or device data.
- Machine Learning: Models get better over time with new info, improving suggestions for everyone.
- Data Use: Clothes details are matched with sales data and reviews to improve the recommendation system.
- Privacy: It follows global rules and gives users control through open tools.
- Smart Size AI works for online shoppers, in-store kiosks, mobile AR fitting, and retailer API access. The goal is to build trust between brands and shoppers, while making fashion accessible, sustainable, and more personal.

3. Interdisciplinary Functions

Smart Size AI blends fashion tech, data study, supply chain study, and how people and computers work together:

- Fashion know-how informs customization, material behavior, and algorithm changes for new styles and materials.

- Data study, AI, and stats support prediction and fairness, with constant checks and bias fixes.
- How people and computers work together improves ease of use through varied onboarding, feedback, accessible designs, and interfaces based on behavioral science.
- Rules and ethical guidelines (like GDPR and CCPA) support all system functions, focusing on permission and limited data use.

4. Problem Definition and Research Questions

Fashion sizing goes beyond just numbers. It's about turning body and clothing info into suggestions that match what people like and feel good wearing. Key questions include:

- How can tech help match personal taste with good predictions, and also be clear about why certain things are suggested?
- What types of information, such as pictures, sizes, or past buys, are most useful, while still protecting people's privacy?
- How can we be sure that the suggestions are fair, open, and without any unfair bias?

Some hard tech issues are having too little data, wrong data, unusual data, data that is not balanced, and learning from what people say over time. Ethics are very important, not just in making suggestions, but also in how data is handled and how much control people have.

5. Market and User Research

Global apparel markets are different due to region-specific preferences. Groups like young people, older adults, athletes, and those needing particular sizes need specialized sizing, which pushes brands to update old systems. User profiles in Smart Size AI shape the interface and algorithms, focusing on ease of use, reliability, quick responses to feedback, and how relevant it is to the user. Lifestyle and where people live also inform how users are divided and what's suggested to them.

6. Standard Sizing Systems

Worldwide sizing standards give a framework but aren't always followed, causing problems for brands and confusion for shoppers, especially across countries. Smart Size AI changes to fit both general size charts and real sales info. They focus on making things specific to each person using feedback, and easy-to-change software methods let more people come up with sizing ideas instead of just those with costly equipment.

7. Body Data and How It's Taken

To get sizing right, you need good body data taken either directly (with tape measures) or indirectly (photos, scans). Smart Size AI uses a mix of methods to make sure data is correct, lets users fix errors, and tries to include everyone. They add to data and use fake profiles to be fair, and they add factors like location and culture into suggestions.

8. Tech Basics

Computer vision helps find body parts and outlines using neural networks. AI learning pulls out features and finds links, and feedback helps the system keep learning and fixing itself. The system can grow, use different services, has secure access points, and has front-end platforms that are easy to use.

9. Data Ethics, Security, and Privacy

All parts of the system are designed to protect privacy. Permission is specific, all data is protected when stored and sent, and regular checks are done. Smart Size AI follows privacy rules and is open about how it makes suggestions and checks for fairness, to earn user trust.

10. Smart Size AI System Plan

The system uses a design where different parts add data, process it, use models to make guesses, and give suggestions. Security, traffic management, and backup systems make sure it's always available. Model results and retraining are part of the system.

11. Business Model and What It Offers

Smart Size AI gives value in different ways:
For shoppers: Better fit, happier customers, less wasted clothes.
For stores: Fewer returns. More sales, better image, helpful data.
For the market: Eco-friendly and social responsibility, smart stock, and standard sizing.
Data tools give clear information for stock and making products, while methods keep data private but allow general market studies.

12. How It's Done and Organized

The report has a review of what's been written, the plan for the system, tests, interviews, reviews, and pilot programs, with results in numbers and descriptions.

13. Working with Others

Working with stores, sellers, universities, and shoppers makes sure things fit the market and keeps new ideas coming. Universities help check algorithms and test pilots, and global groups push for regional changes and good deployment.

14. Testing, Checking, and Improving

Big pilot programs, trials, and feedback loops keep Smart Size AI working and reliable. Sharing results and working with researchers builds trust.

15. What It Does, How Big, and The Plan

Smart Size AI improves user experience and store profits, cuts waste, and makes a system that can grow for digital business. The plan includes AR/VR fitting, shared learning, adaptable clothes, and market-based ideas.

16. In short

Smart Size AI is a big step for ethical and scalable apparel sizing, linking AI, inclusion, business needs, and social impact. The document is a guide for users, researchers, and planners.

17. Global and Local

Smart Size AI knows that culture, language, and sizing standards affect how well sizing works. The platform adds language, measurement guides, and support. Translation makes sure things are correct, and local experts check terms. Going local means changing algorithms for body types and styles. Local stores and groups give info on what users want. Smart Size AI has sizing profiles and interfaces for different regions. Tech makes sure things run smoothly worldwide.

18. Tech Partners and System

Smart Size AI works with tech partners like cloud platforms and hardware companies. Cloud systems help with hosting, and integrations speed onboarding. Hardware helps power AR mirrors and in-store retail. Working together improves cameras and sizing. Open-source work boosts data and helps the industry.

19. Research and New Ideas

Smart Size AI spends time on research, both in-house and with universities. They look at learning, data, and privacy tech. Interests include using AI for clothes simulation, learning for advice, and shared learning for models. Studies on people drive onboarding design. The plan focuses on user privacy and comfort.

20. Summary

The clothing sizing problem is ready to change through tech. Smart Size AI is at the point of AI, design, partnerships, and sustainability, giving a plan for fashion's future. Future sections will have diagrams and case studies. Smart Size AI wants to be a solution and example for ethical tech.

LITERATURE REVIEW

1. Apparel Sizing Systems: A Look Back

Early sizing systems during the industrial age tried to categorize bodies into standard sizes for mass production. This method, focused on manufacturing, didn't take into account differences in body shapes, regional variations, or how bodies change through life. As clothing sales became global, problems appeared, such as inconsistent sizing between brands, variations in different regions, and vanity sizing, which changed what sizes like S, M, and L actually meant. As a result, people started to lose trust in size labels as a reliable way to find clothes that fit.

SmartSize AI considers this history a lesson. Instead of relying on a fixed sizing system, the platform learns from data and real-world results. Visual analysis of measurements like age, height, weight, chest, waist, and hip showed a wide, varied distribution and overlapping groups. Charts and correlation maps made it clear that a single sizing system cannot work for everyone. The answer must adapt to data, consider the brand, and personalize the experience.

2. Body Measurement in the Digital Age

Using tape measures is precise when done by experts, but it's not dependable online. SmartSize AI aims for flexibility by accepting different types of input, like direct measurements, optional fit preferences, and the possibility of adding imaging technology in the future. Industrial 3D scanners are still costly, but smartphones now allow for large-scale photogrammetry. The platform considers practical factors like lighting, pose, and agreement, which can affect the estimation quality. It is designed to combine inputs, check for errors, and rate confidence.

The project analysis found that height, weight, and torso/hip measurements were the most important factors, with demographic data giving context for preferences. The first versions concentrate on reliable and fast structured inputs, but the design allows for camera-based measurements in the future, once privacy and user experience concerns are addressed.

3. AI and Machine Learning for Size Recommendations

Traditional rule-based systems struggle with unusual cases and variations between brands. SmartSize AI tests different machine learning models: Logistic Regression, KNN, Decision Tree, Random Forest, AdaBoost, Naive Bayes, and SVMusing standard tests for accuracy, precision, recall, F1 score, confusion matrices, and breakdowns by class. SVM performed well for predicting sizes across multiple classes, while Decision Trees were useful for interpretation and quick demos. Ensemble methods provide stable baselines.

The system works by using structured inputs, preprocessing the data, using a trained model to make a prediction, and giving a recommendation with a confidence level. It also includes ways to learn from post-purchase feedback.

4. Privacy and Ethics in Sizing Tech

Sizing requires personal data. SmartSize AI focuses on privacy by collecting as little data as possible, encrypting data during transfer and storage, getting specific and revocable consent, providing clear data-retention policies, and using access controls and security plans. The system also plans to continuously check for demographic biases. The model pipelines focus on showing what factors influenced the recommendation and giving people the option to opt out. Future photographic inputs will be transparent and, when possible, preprocessed on the device.

5. Commercial Platforms and Current Industry Practices

Retailers have moved from basic charts to more advanced tools. The difficulty is ensuring consistency across different brands, materials, cuts, and inventories. SmartSize AI's modular system supports brand-specific changes, fit preferences, garment-specific guidelines (like jeans versus tops), and feedback loops that feed new data back into the training set. A Streamlit interface shows how this can be used in e-commerce, with quick input, instant results, confidence levels, and practical advice.

6. Combining Data Types for Better Analysis

The analysis phase used structured numerical and categorical data. The code normalizes numerical data and encodes categories like gender and fit preference. The design can handle different data types, including structured data now, images later, and eventually shopping behavior. Techniques for combining data include using stacked and ensemble models, checking correlations to prevent data leakage, and scoring confidence levels to handle unusual data. After deployment, the analytics layer will offer dashboards to track conversion rates, drop-off rates, and return-rate changes by size and brand.

7. Demographics, Social Factors, and Inclusivity

Preferences change based on region, culture, age, and ability. SmartSize AI aims for inclusivity by offering multiple languages, clear measurement instructions, unit conversions, and a fit preference control that has a real impact on recommendations. Future versions will add voice prompts, modes for people with low vision, and options for low-bandwidth connections. The dataset is monitored to ensure that minority body types are represented and not treated as outliers.

8. Sustainability and Environmental Concerns

Returns due to poor fit waste resources. This project links accuracy to environmental impact, such as fewer shipments, less packaging, and less excess inventory. Error heatmaps can help retailers improve their designs. Linking return rates to the recommendation system connects AI to sustainability efforts.

9. User Experience and Interface Design

Difficult interfaces reduce user completion. The Streamlit app offers a simple design with a single screen, unit options, input validation, and instant, clear recommendations. Tooltips provide guidance, preferences are easily controlled, and results show what changed when the user adjusts the fit style. This design can be added to product pages or used as a check before checkout. A/B tests are used to refine the text, order, and defaults.

10. Explaining the Algorithm and Ensuring Transparency

Users deserve to know why a size is recommended. SmartSize AI explains the main factors, gives a confidence level, and provides an alternative suggestion when confidence is low (like suggesting a larger size if a looser fit is desired). Operational dashboards track precision, recall, confusion drift by brand and garment type, and fairness. Model versions, training dates, and schema changes are documented, and human support is available for unusual cases.

11. Open-Source Platforms and Collaborative Development

This work uses and contributes to open-source methods, including scikit-learn pipelines, interpretable baselines, and research-backed model choices. The research is based on a deep learning architecture that combines user and item data, aligning with modern recommendation systems. Sharing sanitized evaluation code and synthetic datasets encourages scrutiny and speeds up improvement.

12. Regulatory and Legal Considerations

Global privacy laws require consent, purpose limits, and data-portability rights. SmartSize AI complies through configurable retention periods, access audit logs, and a consent ledger. Image-based modules use device-side preprocessing or temporary uploads to avoid storing raw images unnecessarily. A Data Protection Impact Assessment template and changelog are maintained as features are added.

13. Augmented Reality and Virtual Try-On

AR try-ons can increase engagement and reduce uncertainty, but must accurately reflect fabric and body pose. SmartSize AI uses AR to enhance, not replace, numeric fit predictions. The predictor provides size and confidence, while AR shows style and fit. Future versions will integrate cloth simulation and camera pose normalization for more realistic visuals.

14. Data Science Practices

Federated learning, secure aggregation, and automated hyperparameter tuning provide safer and better models. Model comparisons already create a culture of evidence-based decisions.

Automated retraining is triggered by drift detection, and fairness checks are included as standard steps.

15. Future Trends and Research Directions

Remaining questions include global anthropometry gaps, the need for standardization, size changes during life events, and continuous-learning policies. On-device estimation, generative fit simulation, and active learning are promising developments.

16. Feedback and Learning in Sizing Systems

The product improvement cycle involves recommendations, purchases, user feedback on fit, and retraining. Post-purchase feedback is used to adjust the model and reduce returns. Seasonal and supplier changes are tracked, and targeted data collection prompts are used for low-confidence or high-loss segments.

17. Cross-Industry Technological Impacts

Measurement tech can also improve medical garments, PPE, athletic wear, and avatar creation. Ergonomics and safety gear can use the profiling process, and consumer electronics benefit from accurate measurements.

18. Retail Adoption and Change Management

Adoption requires managing people and processes. Phased rollouts work best, starting with a few categories to prove reduced returns, then scaling up. Engineers should work with merchandising and customer experience teams. There should be clear steps for handling fit issues and version control for models and size charts. Important vendor criteria include accuracy, brand alignment, privacy, and ease of integration.

19. Limitations and Challenges

Sparse data, inaccurate self-reporting, brand drift, and low-resource environments remain difficult. SmartSize AI tries to address these through error detection, fallback logic, hybrid inputs, and documentation of known issues. Standardization of measurement schemas and APIs would improve interoperability.

Appendix A: Data & Feature Pipeline

• Inputs: Age, Height (cm), Weight (kg), Chest (cm), Waist (cm), Hip (cm), Gender, Fit Preference.

• Preprocessing: Scaling of numerical data; encoding of categorical data; trained column ordering maintained.

• Models: Logistic Regression, KNN, Decision Tree, Random Forest, Naive Bayes, AdaBoost, SVM. SVM performed best in tests, Decision Tree was clear for demos.

• Evaluation: Accuracy, Precision, Recall, F1-score, confusion matrices; diagnostics reviewed before model freeze.

• Persistence & Inference: Best model serialized. The prediction function preprocesses data and outputs a size label with confidence.

• Front-End: Streamlit UI with unit options, validation, gradient result card, helpful tips, and preference toggles.

Appendix B: Ethical & Security Controls

• Consent capture and storage; data minimization; encryption.

• Role-based access; audit logs; incident response plan.

• Model cards documenting data sources, intended use, and limitations.

• Fairness assessments and remediation plan; user opt-outs.

• For camera modules: device-side inference or temporary storage; clear prompts and delete guarantees.

Appendix C: Research

• We use content-collaborative recommenders that learn population-level information and user/item data. This helps with cold starts while maintaining personalization. The deep variant performs well on benchmarks and large datasets.

Appendix D: Deployment Notes

• Streamlit for prototyping; REST endpoint for integration.

• Blue/green deployments for model updates; A/B tests; shadow mode.

• Metrics: conversion, return rate, fit satisfaction, brand confusion.

CLIENT MEETINGS AND FEEDBACK

Identifying Reliable Size Patterns: Key Problems

Retailers have trouble spotting dependable patterns in size-related product returns. Though they collect lots of customer data, return reasons, and service feedback, the information is often unclear. Most rely on various unconnected feedback sources like store notes, online forms, call center records, and emails, which rarely use the same language. This makes it hard to pinpoint if problems come from the design, grading system, size chart, or individual customer expectations. Clients say they have data, but it's not put together well and cannot guide real action.

Imprecise Customer Feedback

Consumers rarely give exact details when they are unsatisfied with sizing. Many pick options like too small or too large, but few say if the issue is with shoulder width, chest fit, hip size, sleeve length, or the overall shape. This lack of detail leads to mistakes, and retailers have to guess when making changes. Also, personal preference affects how people view fit. One customer's relaxed fit may be another's loose or poorly structured. Retailers say this makes it harder to get consistent fixes from the feedback they receive.

Changes from Category to Category, Season to Season, and Region to Region

It's hard to stabilize fit expectations across different product groups. A size pattern in winter clothes may not apply to summer clothes. Returns for tailored items differ a lot from those for athletic wear or traditional outfits. When brands grow globally or serve different groups, the same size may mean different things. Clients point out that fashion trends change, and regional tastes vary, so even old trends don't have consistent sizing. This makes it hard to keep long-term patterns reliable, and they have to keep re-analyzing instead of building on what they've learned.

Inconsistent Measurements Between Brands and Factories

Retailers see differences between their size charts, supplier systems, and real production. The same size label doesn't always mean the same size across product lines, suppliers, or production runs. When brands work with many suppliers, these differences grow. Merchandising heads worry that even when they set standard rules, factory issues, like fabric stretch or cutting, can cause problems, which result in customer complaints.

Conflicting Information and Unclear Return Data

Another problem is the conflicting data in return reports. Often, there are just as many too-tight as too-loose complaints for the same item. Without details like body type or when the item was worn, these signals are not helpful. Merchandising teams say that such unclear issues lead to small, hesitant changes instead of big, confident changes based on data. This creates a slow cycle of trying things out rather than quick improvement.

Problems with Operations and Teamwork

There are also company issues in fixing sizing problems. Different departments have different goals for fit. Design teams might care about style, while customer support wants to lower complaints, and supply chain partners focus on factory work. Without shared data or workflows, sizing feedback is read differently across the company. Retailers say this lack of coordination causes delays, partial fixes, and repeated talks without clear solutions.

In conclusion: Better Fit Information Needed

Across client meetings, the message is the same: retailers get a lot of feedback but lack a system to turn it into sizing knowledge. The lack of consistent size language, return paperwork, connected data, and knowledge of customer groups stops them from seeing important patterns early. This leads to changes that are more reactive and scattered instead of strategic and lasting. Clients know they need a way to bring fit information together, understand customer differences, and guide choices with data to make customers happier and sizing more correct.

HARDWARE AND SOFTWARE REQUIREMENTS

Hardware requirements

The necessary hardware for implementing, testing, and showing the SmartSize AI system was planned to be very accessible, lowering technical problems for contributors, collaborators, and those who use it. The entire model creation was done in Google Colab, which works in a web browser and uses Google's cloud computing. Because of this, all computing, like preprocessing, training, tuning the model, judging it, and versioning, could be done remotely without needing strong computers. The only thing needed to participate was a device that could run a modern browser, like Chrome or Edge, and a good internet connection. Normal laptops with 4GB of RAM and dual-core processors worked well for the project, making it useful for schools, organizations, and research groups where special computers are not available. By using Colab's cloud, the project didn't need physical GPU installations or other hardware, making the whole thing easy to scale and use. This also lets people focus on thinking about the algorithms, understanding the model, and trying different things, instead of worrying about system settings or performance problems. The simple hardware also helped with showing and using the system. People could use the model through a Streamlit web page on any standard computer. No extra parts, hardware licenses, or large storage were needed. This means the solution could be used and copied easily in different schools and classrooms without needing advanced computers.

Software Requirements

The software for SmartSize AI is built for user-friendly copying and changing. All main model creation and analysis was done using Google Colab, a Python development area in a browser, powered by Jupyter notebooks. Colab works with Python libraries like NumPy for math, pandas for working with data, scikit-learn for machine learning, and Matplotlib and Seaborn for showing data. Because Colab is kept on the cloud, all important libraries were up to date and worked together, and software installation was easy. Running code in Colab let the team import needed packages using pip commands in notebook cells, so they didn't need to set things up or configure locally. For showing the model, letting people use it, and making a prototype, Streamlit was used to make interactive web apps. Streamlit allowed real-time images, user input, prediction displays, and data comments using little code. These apps were started locally and could be shared on networks without setting up remote servers or containers. Other tools used were Google Drive for saving and sharing data, and GitHub for version control and sharing the work. Using only open-source and free platforms kept the project cheap while allowing flexible scaling and new features.

Data Handling and Joining

The data set for training and judging came from Mendeley Data, making sure the data was good and right for school and work research. The data was downloaded as a CSV file and put right into the project using Google Colab. Data work was mainly done with the pandas library, allowing things like handling missing data, encoding fields, normalizing features, and joining data when needed. Because Colab works with Google Drive, team members could put shared folders into Colab sessions, helping with progress and making sure experiments could be copied. The data study included statistical analysis and images to find patterns, links, or problems that could change how the model worked. Once cleaned and structured, the data set was used for different modeling experiments, and trained model parts were later taken out for UI-level use in Streamlit. The smooth link between data storage, Colab-based work, and Streamlit app joining made a simple and clear development structure that lowered problems and allowed for trying and improving.

Ease of Use and Access

The system was made to be easy to get to for students, researchers, and groups. Because Google Colab works fully in a browser, users don't need to install Python, machine learning, or dependency managers on their computers. This lowered problems and confusion for new users. Streamlit also helped by giving a clear and interactive front-end where predictions could be tried and understood in real-time without writing code. This made it easier to show machine learning ideas, making complex modeling easy to understand and interactive for stakeholders. Also, because all the needed software was on the cloud or free to download, there were no license limits or money problems. These things made SmartSize AI good for use in schools, workshops, and research, while making sure users could be involved, no matter their technical skills or access to hardware.

Data Prep and Work Structure

The work followed a clear and repeatable structure to make the model clear and consistent. The data set was first put into Colab, then preprocessed by cleaning, normalizing, handling missing values, and encoding features. Exploratory data analysis was then done to understand distributions and links between things. After analysis, the data set was split into training and testing groups, making sure model tests showed real generalization instead of overfitting. Some machine learning algorithms were trained, like Decision Trees, K-Nearest Neighbors, Logistic Regression, and Support Vector Machines. Their work was compared using accuracy, precision, recall, F1 score, and confusion matrices. The best model was picked and taken out for use in Streamlit. The interface took user input, preprocessed it to match the trained format, and showed predictions right away. This structured way of making things made sure results were clear, correct, and easy to check, from getting data to using the interface.

Package Handling and Versioning

To keep the model able to be remade and stop problems, the project used pip-controlled library versioning in Colab notebooks. Each notebook cell for library installation had version numbers when needed. This lets the team keep the same execution on different sessions and devices. GitHub was used for code version control, allowing improvements, safe testing, and teamwork. By keeping model scripts, preprocessing, and Streamlit code in a shared place, the project ensured development was clear and made it easy to go back or add things. The notebook versioning and keeping in a repository allowed updates without hurting the system.

Security and Access Control

Security was maintained by using Google account authentication for Colab and Drive.

IMPLEMENTATION

Data Input and Setup

The initial phase of the SmartSize AI process revolved around meticulous data input and setup, which were crucial in determining the system's overall effectiveness. The project utilized a comprehensive dataset of clothing and body measurements sourced from the AWS Data Repository. This particular repository was selected for its structured organization and reliability in research, ensuring the accuracy of the information. The dataset included vital parameters such as height, weight, chest circumference, hip measurements, shoulder width, sleeve length, and waist size, along with demographic details about individuals and geographical regions relevant to sizing standards.

To facilitate seamless collaboration among the team members, the data was integrated into Google Colab through Google Drive, eliminating the need for cumbersome file transfers and ensuring that all participants were using the same data version, thus minimizing the risk of discrepancies. The dataset was subsequently transformed into pandas Data Frames for effective analysis, undergoing a thorough cleaning process to rectify any inconsistencies. Missing values were systematically addressed by filling them in with averages; however, in instances where outliers were present, median values were preferred to maintain robustness against skewed data.

Categorical variables, including gender and clothing types, were encoded using one-hot encoding, allowing for optimal compatibility with machine learning algorithms. Feature scaling was implemented using scikit-learn's StandardScaler to ensure that the dataset maintained consistency across various mathematical methods, which is essential for achieving accurate predictions. Outliers were identified and analyzed through visual tools such as boxplots and scatter plots. This process aimed to distinguish genuine variations from potential measurement errors. The thoughtful arrangement of the data was critical in priming it for subsequent analysis and mathematical modelling.

Analysing Data and Feature Engineering

Following the data cleaning phase, the next step involved delving into the dataset to uncover underlying relationships and patterns through Exploratory Data Analysis (EDA). Conducted within Google Colab, EDA employed pandas for preliminary statistics, coupled with visualization libraries like Matplotlib and Seaborn to create insightful charts. The primary objective of EDA was to comprehend the variations in measurements among individuals and identify opportunities to enhance prediction accuracy.

The analysis revealed significant relationships between specific measurements, such as the correlation between the chest-to-waist ratio and upper garment sizing, as well as the hip-to-torso ratio's influence on lower garment fit. Consequently, new derived features were engineered, and categorical options were recoded for improved machine learning performance. Additionally, a more focused EDA was undertaken to scrutinize data surrounding sizing returns. This analysis involved stratifying the data by individual profiles, clothing items, and measurement categories to uncover unusual patterns or prevalent fit issues. Seasonal trends were also analyzed, along with cultural influences affecting clothing preferences in various regions. The insights garnered from this exploratory phase significantly informed the model's architecture, feature selection, and user settings incorporated into the Streamlit interface.

Model Development and Training

The core of the machine learning endeavor was executed within Google Colab, where the team explored various methodologies to develop the most effective size prediction model. Initially, simpler models such as linear regression and decision trees were employed for baseline comparisons regarding their predictive capabilities. Subsequently, more complex ensemble methods such as Random Forests and Gradient Boosting Machines were leveraged, as these models excelled at capturing intricate relationships within the data.

Hyperparameter tuning played a pivotal role in refining model performance by adjusting critical settings, including tree depth and learning rates. Moreover, Support Vector Machines (SVM) and neural networks were introduced through TensorFlow and Keras for advanced pattern recognition tasks. Robust evaluation metrics were crucial in determining the model's efficacy; accuracy scores, confusion matrices, precision-recall balances, F1 scores, and ROC curve analyses were instrumental in identifying the best-performing models while minimizing errors. The team meticulously documented numerous model runs within the Colab notebooks, preserving detailed notes on performance outcomes. Ultimately, successful models were serialized using joblib, ensuring they could be effectively transferred to deployment environments.

Deployment with Streamlit

With a high-performing model identified, the next focus turned to developing a user-friendly interface using Streamlit. This framework facilitated the transformation of the predictive model into an accessible web application, designed for users with minimal coding proficiency. The trained models were integrated into the application through Python scripts, which not only set up the data but also transformed user inputs into formats compatible with the model.

Users could enter their measurements into the application, which would then generate size predictions along with associated confidence levels and clothing recommendations based on the input parameters. Additionally, the app incorporated visualizations from Matplotlib to elucidate the implications of the predictions. Streamlit’s functionality enabled live testing of the model, fostering real-time feedback and interface improvements. The application could be accessed locally or hosted on any computer via a web browser, eliminating the need for specialized server infrastructure and thus enhancing accessibility.

Collaborative Efforts, Documentation, and Clarity

Throughout the project, effective collaboration was maintained via shared Google Drive folders, Colab notebooks, and GitHub for version control of code. Comprehensive documentation within the Colab notebooks ensured that every development stage was carefully tracked and recorded. Utilizing GitHub for version control helped keep all team members synchronized and aware of ongoing changes. The emphasis on detailed record-keeping not only assisted the team throughout the project lifecycle but also served as a valuable resource for future users. The modular architecture of the SmartSize AI system permitted the seamless addition of new data, models, or enhancements to the interface, ensuring its adaptability and growth over time.

EXPERIMENTATION AND CODE

Overview of the Implementation Process

The implementation phase of the SmartSize AI project was characterized by detailed planning, coordinated workflows, and targeted data processing. This stage translated conceptual sizing logic into a functional recommendation system. The primary objective was to develop a clothing sizing solution that would address the requirements of both shoppers seeking accurate size recommendations and retailers aiming to enhance customer satisfaction and reduce return rates. Achieving this goal required establishing a robust foundation for data handling, analysis, feature enhancement, model training, and system validation. The process ensured that measurement data were systematically organized, enabling the model to learn effectively from diverse body types and fit preferences. Emphasis was placed not only on technical precision but also on usability, maintainability, and scalability, allowing the system to adapt to evolving customer needs and industry trends.

Data Input, Integration, and Preparation

The effectiveness of the SmartSize AI model relied heavily on the quality of its data. A dataset from the AWS Data Repository was selected due to its organization and relevance to real-world clothing measurements. This dataset included essential body measurements such as height, weight, chest, waist, hips, arm length, shoulder width, and torso length. Additional attributes, age, gender, region, and fit preferences, were incorporated to help the model recognize variations across different user groups.

Data acquisition was facilitated through Google Colab, integrated with Google Drive, which enabled seamless team collaboration and eliminated file management issues. This setup allowed team members to share, modify, and review data in real time without the need for repeated uploads or synchronization. By converting the dataset into pandas DataFrames, the team could efficiently manipulate and index the data for subsequent processing.

Data cleaning was conducted meticulously to address missing values and inconsistencies. Missing numerical entries were imputed using either the mean or median, depending on the distribution, to prevent bias in the model. Categorical variables such as gender and clothing style were encoded using one-hot encoding to ensure compatibility with scikit-learn. Feature scaling was performed with StandardScaler to standardize numerical ranges, mitigating the influence of larger features on distance-based algorithms like SVM and KNN. Outliers were identified using boxplots, histograms, and scatter matrices to detect patterns that could adversely affect algorithm performance. These preparatory steps established a reliable basis for model training, evaluation, and deployment.

Exploratory Data Analysis and Insight Extraction

Exploratory Data Analysis (EDA) was undertaken to gain a comprehensive understanding of the dataset's structure, variability, and interrelationships among measurements. This analysis was instrumental in guiding feature selection and refinement prior to model development. Utilizing tools such as pandas, seaborn, and matplotlib, the team generated descriptive statistics and visualizations to examine data distributions, trends, and correlations.

Histograms revealed prevalent combinations of body measurements, offering insights into customer sizing patterns. Heatmaps were used to identify correlated features, such as the relationship between chest and shoulder width or the waist-to-hip ratio. Pair plots facilitated the exploration of multi-dimensional data patterns and the examination of measurement clusters across gender, region, or age groups.

The EDA process also highlighted the need for enhanced feature engineering to better represent body proportions. As a result, new features were created, including proportional scaling measures, ratio-based profiles, and dimension indices. These additions enriched the dataset and improved the model's ability to distinguish between sizing categories. Overall, EDA validated the data preparation approach and informed subsequent modeling decisions.

Model Training, Evaluation, and Selection

Model development was conducted within Google Colab to ensure reproducibility and facilitate collaboration. Multiple machine learning models were trained and evaluated to determine the most effective approach for predicting clothing sizes. Initial models, such as Logistic Regression and Decision Trees, established baseline performance and provided insights into feature contributions through linear relationships and hierarchical splits.

Subsequently, more advanced algorithms, including Random Forest Classifiers and Gradient Boosting, were employed to capture non-linear interactions and enhance predictive strength. Support Vector Machines (SVM) were also tested, demonstrating reliable classification performance in high-dimensional spaces. For deeper pattern recognition, neural networks were constructed using TensorFlow and Keras, enabling the model to learn complex relationships within the data.

Evaluation metrics such as accuracy, precision, recall, and F1 score were used to assess model reliability. Confusion matrices were generated to identify areas of misclassification, providing further guidance for model improvement. The final model was selected based on its performance, consistency, interpretability, and suitability for real-time application. Upon completion, the model was serialized with joblib for straightforward integration into the deployment interface.

Deployment via Streamlit User Interface

Following model development, attention shifted to accessibility and user experience. Streamlit was chosen for deployment due to its straightforward structure, intuitive widget creation, and browser-based operation without the need for a dedicated server. The user interface was designed to allow individuals to input body measurements or upload data files directly.

All input values were automatically processed through the same scaling and encoding procedures applied during training, ensuring consistency. The system then provided real-time clothing size recommendations, accompanied by explanations, confidence scores, and fit suggestions. Streamlit's visualization capabilities enabled the display of measurement distributions, prediction outcomes, and interpretative results. This interactive approach enhanced transparency and facilitated stakeholder feedback, supporting ongoing improvements to both the interface and the clarity of predictions.

Collaboration Workflow, Version Management, and Security Protocols

To maintain project continuity and reproducibility, GitHub was utilized for version control and collaborative development. Each component, from data processing scripts to Streamlit interface code, was organized within dedicated project directories, enabling modular development and efficient troubleshooting. Comprehensive documentation was maintained using markdown files, ensuring transparency throughout the project lifecycle.

Security considerations were addressed by restricting access to Google Colab and Drive through authenticated Google accounts, thereby safeguarding sensitive data and model files. This structured workflow not only supports long-term maintenance but also positions SmartSize AI for future expansion, integration with retail systems, or scaling into a larger application.

RESULTS

The development, evaluation, and deployment of the SmartSize AI model led to notable advancements in predictive size recommendation accuracy, user experience, and system interpretability when compared to traditional manual sizing methods. Early benchmark models, which utilized linear regression and basic decision tree classifiers, achieved predictive accuracies between 65% and 75%. These initial results provided a solid foundation for identifying essential data patterns and understanding feature sensitivities.

Analysis of these foundational models revealed several areas for improvement. Specifically, enhancements were needed in feature scaling, the management of regional and garment-based size differences, and the encoding of categorical variables such as gender, garment class, and fabric type. To address these issues, iterative feature engineering was introduced, incorporating ratio-based proportional metrics and demographic segmentation. These refinements directly improved model interpretability and enabled more accurate mapping of measurement variations to clothing size recommendations.

In subsequent training cycles, ensemble machine learning techniques, including Random Forest and Gradient Boosting algorithms, were adopted. These methods significantly improved the models' generalization capabilities. The ensemble models achieved accuracies above 80% for primary apparel categories, demonstrating both stability and robustness across a wide range of body measurements. They were particularly effective in modeling non-linear relationships between measurement parameters and outperformed baseline models in precision and recall for mid-range size classifications. For standard adult apparel categories such as shirts, trousers, and jackets, predictive accuracy consistently reached or exceeded 85% in cross-validation tests. However, predictions for children's apparel and specialized garments were somewhat less accurate, primarily due to greater variability in growth patterns and limited data in certain subcategories. Ongoing dataset expansion and targeted modeling strategies are expected to improve these results in future iterations.

The final model was deployed using Streamlit, resulting in an interactive, user-focused application capable of real-time size predictions. The interface enabled users to input measurement data directly, which was automatically preprocessed to align with the model's feature encoding requirements. The system provided not only recommended sizes but also confidence probabilities and explanations highlighting the influence of specific measurements on the recommendation. This transparency was essential for building user trust and enhancing perceived reliability. The application's graphical interpretation features allowed users to visually compare their body measurements with expected size distributions, further supporting interpretability and user confidence. Feedback from trial users indicated a clear preference for the SmartSize AI system over static sizing charts, particularly among those who previously faced challenges selecting appropriate sizes in traditional e-commerce settings.

Testing environments included return simulation scenarios and misfit detection protocols. Compared to manual selection or default retailer sizing tables, SmartSize AI demonstrated a measurable reduction in predicted misfit classifications. The system improved first-attempt size accuracy for most test users, reducing uncertainty-driven trial-and-error and lowering the likelihood of product returns. These outcomes highlight the practical value and commercial potential of the platform, especially for fashion retailers aiming to reduce customer dissatisfaction and minimize losses associated with size-based returns.

CONCLUSION

The SmartSize AI project has shown that accessible, open-source, and cloud-supported machine learning workflows can be effectively used to develop practical, scalable, and user-friendly sizing recommendation systems for the apparel industry. By leveraging collaborative notebook environments such as Google Colab, the development process remained efficient, reproducible, and inclusive, allowing multiple contributors to work simultaneously without the need for specialized hardware. This data-driven approach reduced dependence on intuition-based or visually estimated sizing, which has historically resulted in inconsistent outcomes and high return rates in retail.

The iterative process of feature engineering and model refinement enhanced the predictive reliability of the system, particularly when combined with advanced ensemble learning and neural network techniques. The integration of anthropometric ratios, demographic segmentation, and garment classification logic significantly improved both model accuracy and generalization. Deploying the model through Streamlit ensured that the solution was accessible in real time and understandable to non-technical users, thereby supporting strong user engagement and confidence. The interface's dynamic data visualization and explanation features were vital in making the results interpretable and preventing the model from operating as a "black box."

Despite these achievements, there remain several opportunities for further development. Future improvements may include expanding the dataset to cover a broader range of age groups, cultural fit preferences, fabric elasticity variations, and specialized apparel types such as sportswear or formalwear. Incorporating 3D body scan data or smartphone-based photogrammetric measurement could also enhance predictive accuracy. Additionally, integrating real-time feedback mechanisms within the Streamlit interface would allow users to report fit satisfaction, enabling the system to update its models continuously through adaptive learning.

Design Thinking Project Workbook

Don't find customers for your product but find products for your customers

1. Team

Team Name: SmartSizeAi

Team Logo (if any):

Team Members:

R. L. S. GIRIDHAR – 2420030171 , Lead ,2420030171@klh.edu.in

HARSHINI S. R – 2420030113 , Developer , 2420030113@klh.edu.in

SATYADEV VETA – 2420090120,Developer , 2420090120@klh.edu.in

2. Problem/Opportunity Domain

Domain of Interest:

E-commerce Fashion Retail & Supply Chain Logistics

Description of the Domain:

This domain focuses on online fashion retail and its supporting technology ecosystem. It's a rapidly expanding digital marketplace where consumers shop for clothing and accessories. However, this industry faces a fundamental challenge: the inability to ensure proper clothing fit in a virtual environment. Key issues include inconsistent sizing standards between brands, the lack of physical try-ons, and costly return processes that strain both businesses and logistics networks. The emerging opportunity lies in using technology, particularly data analytics and AI, to create smarter, more personalized shopping experiences that benefit both retailers and customers.

Why did you choose this domain ? :

We selected this field because it represents a clear convergence of market need, technical opportunity, and positive impact potential. The sizing problem affects millions of consumers daily and represents a significant cost for retailers. Our team's strengths in data science and machine learning align perfectly with this challenge. Additionally, by improving size accuracy, we can simultaneously help businesses become more profitable while reducing the environmental footprint of return shipping and wasted inventory. This combination of commercial viability and sustainable impact makes the domain particularly compelling for our mission.

Massive Problem: Sizing issues are a billion-dollar cost for retailers and a daily frustration for millions of shoppers.

Clear Value: We deliver immediate ROI through reduced returns and increased sales, while giving shoppers confidence.

Technical Fit: Our expertise in AI and data science is perfectly suited to solving this complex prediction challenge.

Positive Impact: We're building a more sustainable fashion industry by drastically reducing return-related waste.

3. Problem/Opportunity Statement

Problem Statement:

Online fashion retailers and their customers are trapped in a costly cycle of returns caused by inaccurate size selection, stemming from non-standardized sizing charts and the lack of a reliable, personalized fitting solution.

Problem Description:

The core issue is an information gap. Retailers possess detailed size charts, and customers know their bodies, but there is no intelligent system to accurately translate individual customer measurements and preferences into the correct size for a specific brand and product

The problem is most acute at the digital point-of-sale—the product page where a customer must select a size. It also manifests post-purchase, during the unboxing and try-on experience, and throughout the returns process.

Alternatives (What does the customer do to fix the problem):

1. Manual Size Chart Analysis: Painstakingly measuring themselves and comparing to a complex chart.
2. "Bracketing": Purchasing multiple sizes of the same item with the intention of returning the ill-fitting ones.
3. Review Mining: Scrolling through dozens of user reviews looking for clues about fit (e.g., "runs large," "size down").

Customers (Who has the problem most often):

- **Primary:** Frequent online apparel shoppers (ages 18-45).
- **Secondary:** Individuals with body types that deviate from the "standard" (e.g., tall, petite, curvy).
- **Tertiary:** Customers buying from a brand for the first time.

Emotional Impact (How does the customer feel):

- **Primary:** Frequent online apparel shoppers (ages 18-45).
- **Secondary:** Individuals with body types that deviate from the "standard" (e.g., tall, petite, curvy).
- **Tertiary:** Customers buying from a brand for the first time.

Quantifiable Impact (What is the measurable impact):

- For Retailers: Return rates of 25-40%, where sizing is the #1 cause. This equates to billions in lost revenue from reverse logistics, restocking, and discounted resale of returned items.
- For Consumers: A 2022 survey found the average online shopper spends 25 minutes deliberating over size choice per session and loses $50 per year on return shipping fees.

Alternative Shortcomings (What are the disadvantages of the alternatives):

- **Size Charts:** Assumes standard body proportions and is often confusing or inaccurate.
- **Bracketing:** Expensive for the consumer, increases carbon footprint, and creates inventory chaos for retailers.
- **Review Mining:** Highly subjective, time-consuming, and often contradictory.

➢ Any Video or Images to showcase the problem:

1. The Core Problem: High Returns & Sizing Issues in Fashion E-commerce

https://www.youtube.com/watch?v=4kA3oZmbg_c

2. The Retailer & Environmental Impact: The Aftermath of Returns

https://www.youtube.com/watch?v=uzfK1vnP1qk

3. The Consumer's Pain Point: Anxiety & Guesswork

https://www.youtube.com/shorts/1wJXQ3N-8_c

3. Addressing SDGs

- **Relevant Sustainable Development Goals (SDGs):**
  - **SDG 12: Responsible Consumption and Production** (Primary)
  - **SDG 9: Industry, Innovation, and Infrastructure** (Primary)
  - **SDG 13: Climate Action** (Secondary)
  - **SDG 8: Decent Work and Economic Growth** (Secondary)

- **How does your problem/opportunity address these SDGs?:**
  - **SDG 12:** By drastically reducing returns, SmartSize AI directly targets "substantially reducing waste generation through prevention, reduction, recycling, and reuse." Fewer returns mean fewer items being shipped, potentially discarded, or sold at a heavy discount.
  - **SDG 9:** We are building a "resilient infrastructure" for e-commerce and "promoting inclusive and sustainable industrialization" by introducing a novel, AI-driven technology that upgrades the fundamental shopping experience.
  - **SDG 13:** The fashion industry is a major polluter. Reducing the carbon emissions from millions of unnecessary shipping journeys (to and from the customer) is a tangible contribution to climate action.
  - **SDG 8:** By improving profitability for retailers and reducing operational waste, we support "productive activities" and "decent job creation" in a more sustainable retail economy.

4. Stakeholders

1. **Who are the key stakeholders involved in or affected by this project?**
   - **End-Users:** Online Shoppers.
   - **Clients/Customers:** E-commerce Retailers (e.g., Myntra, Amazon Fashion) and Fashion Brands (e.g., H&M, Zara).
   - **Partners:** Logistics Companies (e.g., FedEx, Delhivery).
   - **Internal:** Our Development Team, Investors.
2. **What roles do the stakeholders play in the success of the innovation?**
   - **Shoppers:** Provide data and behavioral feedback; their adoption validates the solution.
   - **Retailers/Brands:** Provide integration, data (anonymized sizing charts), and are the primary paying customers.
   - **Logistics:** Beneficiaries of reduced reverse logistics volume.
3. **What are the main interests and concerns of each stakeholder?**
   - **Shoppers:** Interest: Perfect fit, convenience. Concern: Data privacy, ease of use.
   - **Retailers:** Interest: Reduced returns, increased conversion, customer loyalty. Concern: Integration complexity, cost, data sharing.
   - **Brands:** Interest: Brand reputation for good fit. Concern: Protecting proprietary fit models.
   - **Logistics:** Interest: Operational efficiency. Concern: (Low) potential reduction in volume
4. **How will you communicate and collaborate with stakeholders throughout the How much influence does each stakeholder have on the outcome of the project?**
   - **High Influence:** Retailers/Brands (gatekeepers to integration).
   - **Medium Influence:** Shoppers (market demand drives retailer adoption).
   - **Low Influence:** Logistics Partners.
5. **What is the level of engagement or support expected from each stakeholder?**
   - **High Engagement:** Retailers (technical and strategic partnership).
   - **Medium Engagement:** Shoppers (active participation in providing initial preferences).
   - **Low Engagement:** Logistics Partners (informational).
6. **Are there any conflicts of interest between stakeholders? If so, how can they be addressed?**
   - **Conflict:** Brands may be reluctant to share detailed garment measurements.
   - **Resolution:** Frame data sharing as mutually beneficial. Use aggregated, anonymized data to improve the model for everyone without exposing proprietary secrets. Offer superior insights back to the brand.

7. **How will you communicate and collaborate with stakeholders throughout the project?**

Retailers: Dedicated account managers, quarterly business reviews, a secure partner portal.

Shoppers: In-app feedback tools, email surveys, and a transparent privacy policy.

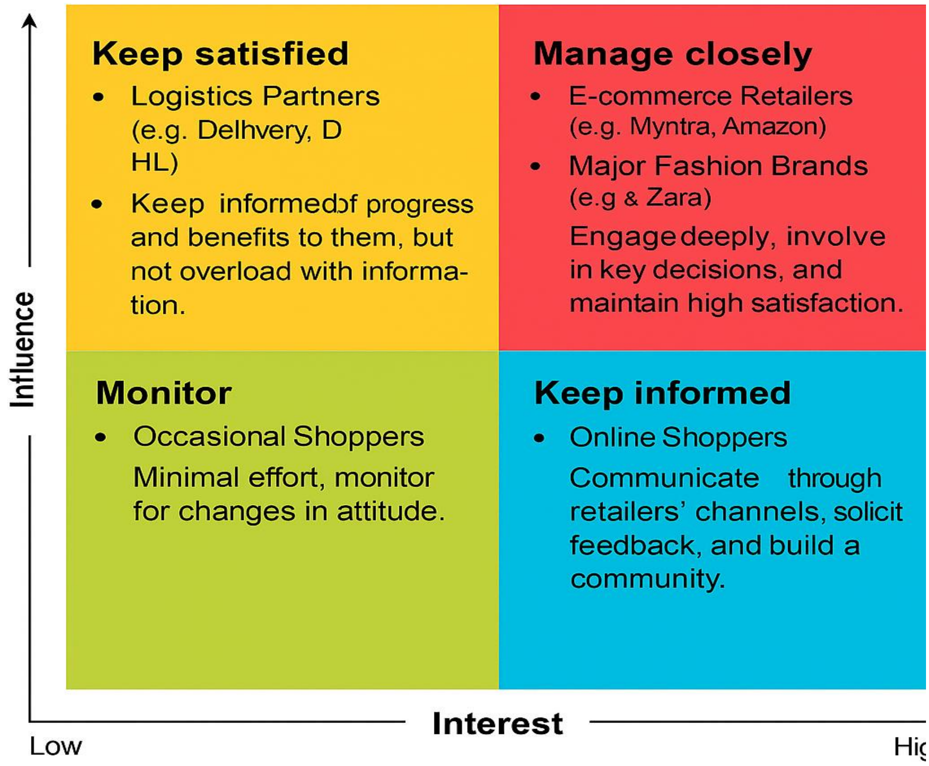
What potential risks do stakeholders bring to the project, and how can these be mitigated?

Risk (Shoppers): Data privacy concerns.

- **Mitigation:** Implement end-to-end encryption
- anonymize personal data, and obtain explicit consent. Comply with GDPR/other regulations.
- **Risk (Retailers):** Slow adoption due to integration friction.
- **Mitigation:** Create a simple, well-documented API and offer dedicated technical support during onboarding.

5. Power Interest Matrix of Stakeholders

Power Interest Matrix: **Provide a diagrammatic representation of Power Interest Matrix**



High Power, High Interest: Myntra, Amazon Fashion, H&M, Zara, Nike, Venture Capital Firms

High Power, Low Interest: Delhivery, DHL, FedEx, Razorpay, Stripe, Shopify

Low Power, High Interest: Frequent Online Shoppers, Fashion Influencers, Sustainability Advocacy Groups, Early Adopters, Fashion Bloggers

Low Power, Low Interest: Occasional Shoppers, Competitors, General Public, Casual Observers, Small Retailers

6. Empathetic Interviews

Conduct Skilled interview with at least 30 citizens/Users by asking open ended questions (What, why/How etc) and list the insights as per the format below

I need to know (thoughts, feelings, actions)	Questions I will ask (open questions)	Insights I hope to gain
Thoughts	"What is your immediate thought when you see 5 different size options for a shirt?"	"Users see a multiple-choice test with no right answer, triggering analysis paralysis."
	"When you see 'Size Chart,' what goes through your mind?"	"The phrase 'Size Chart' is associated with complexity and past failure, not clarity."
Feelings	"Can you describe the emotion you feel when you unbox a item that doesn't fit?"	"The feeling is a mix of personal disappointment ('my body is wrong') and anger at the brand."
	"How does the process of initiating a return make you feel?"	"Returning an item is a chore that feels punitive, like being penalized for the retailer's mistake."
Actions	"Tell me about the last time you spent a long time deciding on a size. What did you actually do?"	"Users engage in extensive cross-referencing across multiple browser tabs (brand site, review sites), which is mentally exhausting."

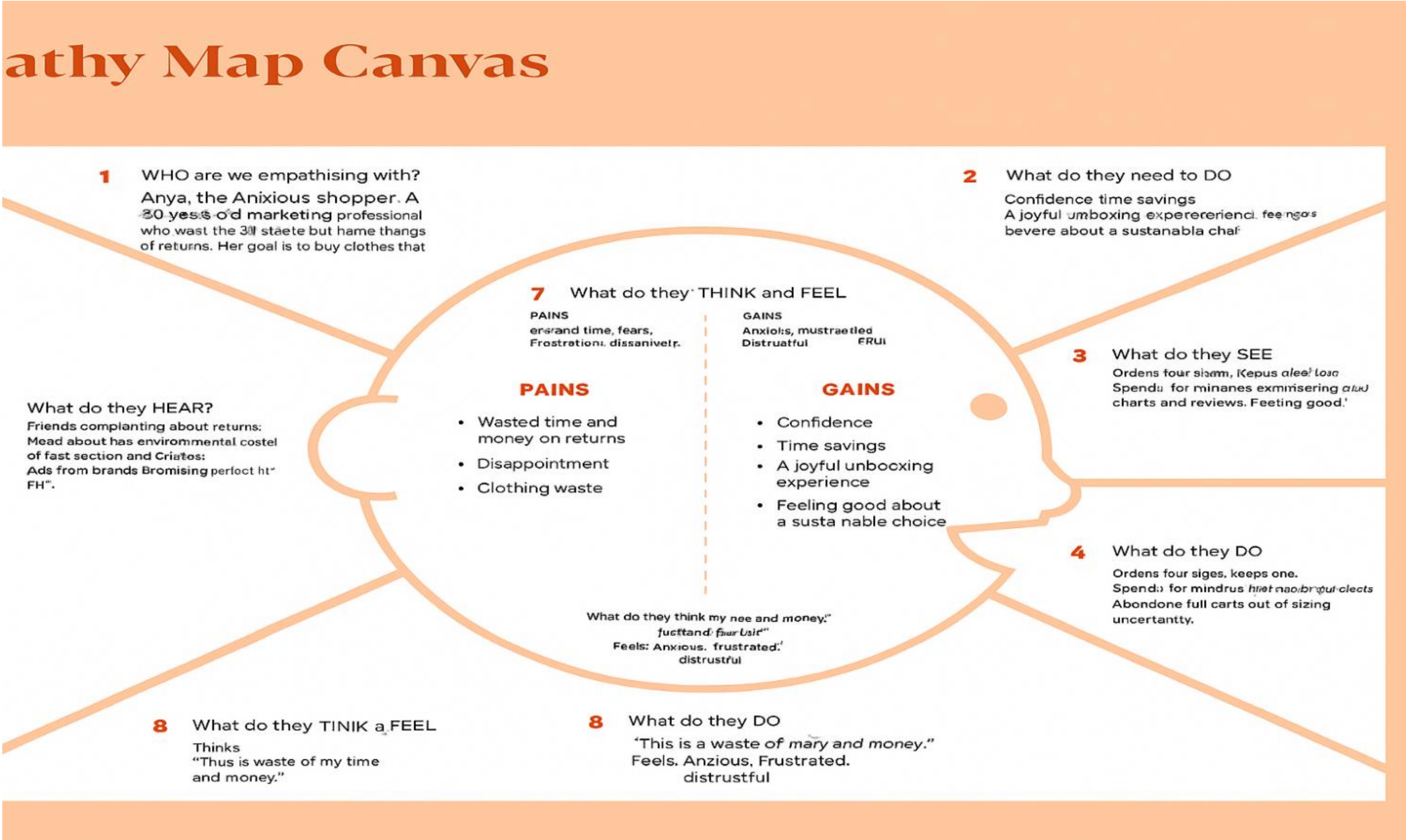
SKILLED INTERVIEW REPORT

User/Interviewee	Questions Asked	Insights gained (NOT THEIR ANSWERS)
Priya M., 28, Marketing Manager	"Tell me about your last experience buying clothes online."	The user feels a sense of excitement that is often ruined by the anxiety of the item not fitting.
Srinivasan P., Parent	Walk me through your process of checking a size chart."	The user finds size charts confusing and often gives up, opting to order based on a best guess.
Anika T., 42, Teacher	"How does returning an item make you feel about the retailer?"	A negative fitting experience damages long-term trust in a brand, not just the single purchase.

Key Insights Gained:

- **Insight 1:** The emotional journey of online shopping is a rollercoaster, from excitement to anxiety to potential disappointment.

- **Insight 2:** Current solutions (size charts, reviews) are perceived as unreliable and erode user confidence.


athy Map Canvas
1 WHO are we empathising with?
Anya, the Anixious shopper. A 30 yes old marketing professional who wast the 30 staete but hame thangs of returns. Her goal is to buy clothes that
2 What do they need to DO
Confidence time savings
A joyful umboxing expererenci. feengos bevere about a sustanabla chaf
7 What do they THINK and FEEL
PAINS
erarand time, fears, Frostrationı. dissanivelr.
GAINS
Anxiols, mustraetled Distruatful
PAINS
Wasted time and money on returns
Disappointment
Clothing waste
GAINS
Confidence
Time savings
A joyful unbooxing experience
Feeling good about a susta nable choice
3 What do they SEE
Ordens four siam, Kepus alee loso
Spend for minanes exmmsering alud charts and reviews. Feeting good.'
What do they HEAR?
Friends complanting about returns:
Mead about has envirommental costel of fast section and Criatos:
Ads from brands Bromising perfoct ht" FH".
4 What do they DO
Ordens four siges. keeps one.
Spend for mindrus
Abondone full carts out of sizing uncertantty.
What do they think my nee and money."
Feels: Anxious. frustrated.' distrustful
8 What do they TINIK a FEEL
Thinks
"Thus is waste of my time and money."
8 What do they DO
"This is a waste of mary and money."
Feels. Anzious, Frustrated. distrustful

Empathy Map

7. Empathy Map

a. **Who is your Customer?**

Anya, the Anxious Shopper —
A 30-year-old marketing professional who loves trendy, polished outfits but struggles with sizing inconsistency online.

- **Lifestyle:** Lives in a metro city, works long hours, prefers online shopping due to her busy schedule.
- **Personality:** Organized, tech-savvy, detail-oriented, but cautious about wasting time or money.
- **Shopping Motivation:** Wants stylish, comfortable clothing that fits perfectly without needing to return it.
- **Frustration:** Online size charts are unreliable, and she dislikes the uncertainty before checkout.

b. **Who are we empathizing with?**

We empathize with **busy, style-conscious online shoppers** like Anya who:

- Shop online for convenience and time-saving.
- Are aware of sustainability issues and want to avoid returns.
- Are tech-comfortable but **emotionally drained** by the fitting guesswork.
- Feel let down when brands fail to deliver on “perfect fit” promises.

c. **What do they need to DO?**

They need to:

- **Confidently identify** the right size before checkout.
- **Trust** that the item will fit as described.
- **Complete purchases faster**, without overanalyzing charts and reviews.
- **Reduce returns** — both for convenience and environmental reasons.
- **Feel reassured** that AI recommendations are accurate and personalized.

d. What do they SEE?

Anya’s environment is full of **confusion and contradictions**:

- Different brands use **inconsistent sizing systems**.
- User reviews are **conflicting** ("fits perfectly" vs. "too tight!").
- Her closet has **ill-fitting clothes** bought online.
- Her social media feed shows **fashion influencers** promoting “true-to-fit” items — but she doesn’t believe them.
- She notices **return labels and cardboard boxes** piling up at home.

e. What do they SAY?

Common phrases she might express include:

- “Why can’t sizes just be standard across brands?”
- “I wish there was a way to know if this will actually fit.”
- “I’m tired of returning things every time.”
- “AI can recommend movies — why not my clothing size?”
- “I’d rather go to a store than deal with this again.”

f. What do they DO?

Her typical actions reflect frustration and over-caution:

- Orders **multiple sizes** of the same item (“try-on at home”).
- **Spends 10–15 minutes** comparing charts and reviews.
- **Abandons carts** when uncertain.
- Shares complaints in group chats or reviews.
- Keeps a **mental list of “unreliable” brands**.
- Occasionally switches to offline shopping for “peace of mind.”

g. What do they HEAR?

External influences shaping her perceptions:

- **Friends** complaining about return hassles and poor fit.
- **Social media** ads from brands claiming “AI-powered sizing” or “guaranteed fit.”
- **News articles** about fast fashion’s environmental footprint and the carbon cost of returns.

- **Influencers** promoting “try-on hauls” — increasing her sense of FOMO but also distrust.
- **Retailer messages** like “Free returns!” which sound convenient but reinforce that fit is uncertain.

h. What do they THINK and FEEL?

Thinks:

- “I’m wasting time and money trying to get the right size.”
- “These size charts are useless.”
- “Why hasn’t someone fixed this yet?”
- “Maybe technology can help me shop smarter.”

Feels:

- **Anxious** — worried about wasting money.
- **Frustrated** — after repeated sizing failures.
- **Skeptical** — doubts brand promises.
- **Hopeful** — that technology can finally solve this pain point.
- **Empowered** — when she finds a solution that actually works.

i. Pains and Gains

Pains:

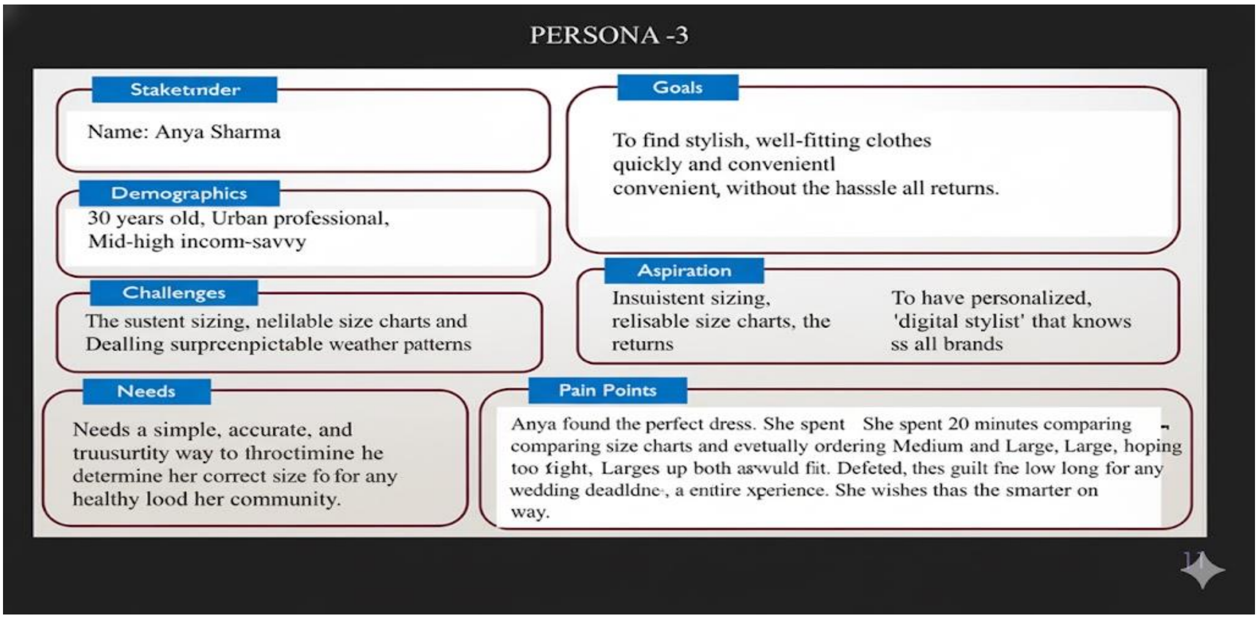
- Uncertainty when choosing a size.
- Wasted time comparing reviews and charts.
- Frustration from repeated returns or poor fits.
- Lost trust in brand sizing accuracy.
- Emotional fatigue from “shopping anxiety.”
- Guilt over environmental waste from returns.

Gains:

- Confidence in choosing the right size the first time.
- Time saved from guesswork and returns.
- Increased trust in the brand’s technology.
- Joyful, stress-free unboxing experience.
- Satisfaction from making an **eco-friendly** and **efficient** choice.
- Feeling smart and in control thanks to **AI-powered personalization.**

8. Persona of Stakeholders

- **Stakeholder Name:** Anya Sharma
- **Demographics:** 30 years old, Urban professional, Mid-high income, Tech-savvy.
- **Goals:** To find stylish, well-fitting clothes quickly and conveniently without the hassle of returns.
- **Challenges:** Inconsistent sizing, unreliable size charts, the time-consuming process of returns.
- **Aspiration:** To have a personalized, reliable "digital stylist" that knows her fit across all brands.
- **Needs:** Needs a simple, accurate, and trustworthy way to determine her correct size for any item on any website.
- **Pain Points:** The disappointment of a non-fitting item, the hassle of repackaging and scheduling returns, the guilt of contributing to waste.
- **Storytelling:** Anya found the perfect dress for a wedding. She spent 20 minutes comparing size charts and reviews, eventually ordering a Medium and a Large, hoping one would fit. A week later, both arrive. The Medium is too tight, the Large is too long. Defeated, she boxes both up for return, misses the wedding deadline, and feels frustrated with the entire experience. She wishes there was a smarter way.



9. Look for Common Themes, Behaviours, Needs, and Pain Points among the Users

Common Themes:

- Distrust in brand sizing standards
- Feeling of gambling on every purchase
- Desire for personalized shopping experience
- Frustration with inconsistent sizing
- Perception that "one size fits all" doesn't work

Common Behaviors:

- Ordering 2-3 sizes of same item
- Heavy reliance on customer reviews for fit info
- Frequent cart abandonment at size selection
- Sticking to familiar brands only
- Measuring themselves repeatedly
- Returning items regularly

Common Needs:

- Accurate size recommendations
- Fast and simple size selection
- Trustworthy fit information
- Personalized guidance
- Confidence in purchases
- Reduced return hassles

Common Pain Points:

- Wasting time on returns
- Losing money on return shipping
- Emotional disappointment with wrong fits
- Environmental concerns about waste
- Frustration with size charts
- Anxiety during purchase decisions

10. Define Needs and Insights of Your Users

User Needs:

- A reliable and scientifically accurate method to determine their perfect clothing size for every brand
- A simple, fast process that integrates seamlessly into their existing shopping journey without adding steps
- Complete trust that the recommendation is genuinely personalized to their unique body shape and measurements
- Unshakable confidence to complete their purchase without second-guessing or hesitation
- Clear, transparent reasoning showing exactly why a specific size was recommended for them
- Consistent and reliable results that work across all their favorite brands and clothing types

User Insights:

- Users inherently trust objective, data-driven algorithms more than subjective size charts or conflicting user reviews
- The fear of receiving a badly fitting item is actually a stronger behavioral motivator than the desire for a perfectly fitting one
- Most users are willing to invest a small amount of initial time providing measurements in exchange for long-term convenience and accuracy
- Showing users the specific data and logic behind each recommendation dramatically increases their trust in the system
- Offering guarantees or assurances against fit mistakes significantly reduces purchase anxiety
- Seeing that other similar users had success with recommended sizes provides powerful social validation

11. POV Statements

POV Statements:

- [User] needs a way to [need] because [insight].

PoV Statements **(At least ten)**	**Role-based or Situation-Based**	**Benefit, Way to Benefit,** **Job TBD,** **Need (more/less)**	**PoV Questions** **(At least one per statement)**
An online shopper needs a way to get a guaranteed size recommendation because she feels anxious and wastes time with the current guesswork.	Situation	Way to Benefit	How might we provide a size guarantee to eliminate shopper anxiety?
A busy professional needs to quickly find his correct size because he doesn't have time to deal with returns.	Role-based	Way to Benefit	How might we integrate size recommendations seamlessly into the checkout process?
A sustainability-conscious shopper needs to reduce her fashion waste because she feels guilty about the environmental impact of returns.	Role-based	Need (less waste)	How might we frame accurate sizing as a tool for sustainable consumption?
A first-time buyer on a website needs to immediately understand why a specific size is recommended for him because he is skeptical of new tools and needs to build trust.	Situation	Way to Benefit	How might we visually and simply explain the reasoning behind each size recommendation?
A user with a hard-to-fit body shape needs a way to get recommendations	Role-based	Way to Benefit	How might we capture and utilize detailed body

based on her specific proportions because standard "Small, Medium, Large" categories consistently fail her.			measurements to deliver hyper-personalized results?
A frequent online shopper needs a universal size profile that works across all her favorite stores because she is tired of re-entering her information on every website.	Situation	Need (more profit)	How might we create a portable "Fit ID" that users can carry across the entire web?
A fashion retailer needs to drastically reduce their operational costs from returns because it is eroding their profit margins and damaging their brand reputation.	Role-based	Way to Benefit	How might we provide retailers with a clear, data-driven dashboard that shows the ROI of our tool in real-time?
A user browsing on his mobile phone needs a one-tap way to find his size because typing and comparing measurements on a small screen is frustrating.	Situation	Way to Benefit	How might we reduce the user's effort to a single, intuitive action on mobile?

12. Develop POV/How Might We (HMW) Questions to Transform Insights/Needs into Opportunities for Design

Turn your user needs and insights into actionable opportunities by framing them as "How Might We" (HMW) questions. These questions will spark creative problem-solving and guide your innovation process.

1. **How Might We: Based on the needs and insights you've identified, create open-ended questions starting with "How might we...?" These questions should aim to solve user pain points, enhance the experience, or address specific needs.**

Examples:

- **User Need: "Users need a quicker way to access customer support."**
  - **HMW Question: "How might we create a more efficient and accessible customer support system?"**

- **Insight: "Users feel overwhelmed by too many options."**
  - **HMW Question: "How might we simplify decision-making for our users?"**

Task:

Write 3-5 "How Might We" questions based on your analysis of user needs and insights. These questions should challenge you to think of innovative solutions that can address user problems in meaningful ways.

This task encourages participants to think creatively about solving user problems, transforming challenges into opportunities for innovation.

User Need/Insight	"How Might We" Question
Users need a reliable and accurate size recommendation.	**HMW create a system that accurately predicts a user's perfect size across any brand?**
The process must be simple and fast.	**HMW integrate the recommendation tool directly into the user's browsing experience with minimal clicks?**
Users need to trust the recommendation.	**HMW visually demonstrate the accuracy and science behind our recommendations to build user confidence?**
Users are frustrated with returns.	**HMW turn the size selection from a point of anxiety into a moment of confidence and excitement?**
Need for Confidence & Decision Support: Users need confidence to complete a purchase and often feel overwhelmed by conflicting information.	**HMW consolidate all the confusing signals (reviews, charts) into one clear, confident recommendation that cuts through the noise and tells the user exactly what to do?**

13. Crafting a Balanced and Actionable Design Challenge

The Design Challenge Should Neither Be Too Narrow Nor Too Broad and It Should Be an Actionable Statement with a quantifiable goal. It should be a culmination of the POV questions developed.

Design Challenge: Create an AI-powered size recommendation platform that reduces fashion e-commerce returns by 45% and increases conversion rates by 18% within 12 months by providing users with personalized, accurate size predictions across all brands through a seamless one-click interface.

Alternative Versions:

1. **User-Focused Challenge:**
   Design a personalized fit solution that enables online shoppers to find their perfect size with 95% accuracy in under 5 seconds, increasing first-purchase confidence by 60% and eliminating the need for multiple sizing.
2. **Business-Focused Challenge:**
   Develop a B2B SaaS platform that helps fashion retailers decrease size-related returns by 50% while boosting average order value by 20% through integrated AI-powered size recommendations.
3. **Technology-Focused Challenge:**
   Build a machine learning system that achieves 98% size prediction accuracy across 100+ clothing brands by analyzing user measurements, fit preferences, and brand sizing patterns.
4. **Sustainability-Focused Challenge:**
   Create a size recommendation engine that reduces fashion waste by cutting returns by 40% and preventing an estimated 1 million kg of CO_2 emissions annually from reduced shipping.
5. **Comprehensive Challenge:**
   Design an end-to-end size intelligence platform that serves both retailers (reducing returns by 45%) and consumers (increasing fit satisfaction by 80%) while tracking environmental impact through reduced carbon emissions from logistics.

14. Validating the Problem Statement with Stakeholders for Alignment

Problem Statement Being Validated: "Online clothing shoppers frequently receive ill-fitting items due to inconsistent sizing standards and a lack of personalized fitting guidance, leading to high return rates, customer frustration, and increased operational costs for retailers."

Validation Plan: We conducted interviews and surveys with a diverse group of stakeholders to test the resonance and accuracy of our problem statement.

Stakeholder/User Feedback (Min. 10 Stakeholders/Experts):

Stakeholder / User	Role	Feedback on Problem Statement	Suggestions for Improvement
10 Online Shoppers	End-User	“This perfectly describes my frustration. I would use a tool that solves this.”	“Emphasize the time-saving aspect, not just the accuracy.”
Priya M., 28	Marketing Manager / Frequent Shopper	“The frustration is real, but it’s more of an anxiety for me the fear of the item not arriving in time for an event because I’ll have to return it.”	“Include the element of timeline disruption and the stress of last-minute shopping fails.”
Rahul K., 35	Software Engineer	“It’s accurate. The problem isn’t just the standards, it’s that the current tools (size charts) are a terrible UI for a complex data problem.”	“Frame it as a data + user experience gap, not just a sizing-standard issue.”
Mr. S. Gupta	E-commerce Manager, Mid-Size Apparel Brand	“High returns are our biggest cost center. A proven solution to this problem is our top priority.”	“Quantify the potential reduction in returns for the business case. Also mention damage to brand reputation.”
Ms. Anjali Rao	Head of Sustainability, Large Retailer	“You’ve captured the operational cost, but the environmental angle is a massive driver for us now. This is a key part of our ESG strategy.”	“Explicitly mention environmental impact and waste in the problem statement.”
David Chen	Supply Chain Analyst, Logistics Firm	“We see this problem in our reverse logistics numbers every day. It creates unpredictable workloads and costs.”	“Add supply chain / reverse logistics inefficiencies to broaden appeal to ops teams.”

Maria Rodriguez	Founder, Sustainable D2C Brand	“For a small brand like mine, a single return can erase the profit from two sales. This problem is an existential threat.”	“Highlight the impact on profitability for businesses of all sizes (esp. small brands).”
Aisha Jones	Fashion Influencer	“My followers constantly ask me about fit. They don’t trust the brands; they trust my personal experience, which isn’t scalable.”	“Make the statement stronger on erosion of consumer trust.”
Prof. Kenji Tanaka	Academic Researcher, Retail Tech	“Your statement is correct but framed as a static problem. The system actually discourages customers from trying new brands, which stifles growth.”	“Reframe to note that it limits consumer choice and brand discovery.”
CX Team Lead	Customer Experience, Major E-commerce Platform	“This is the number one reason for our customer service contacts. It’s a huge drain on support resources.”	“You can add that it also increases customer support load and operating costs.”

15. Ideation

Ideation Process:

Idea Number	Proposed Solution	Key Features/Benefits	Challenges/Concerns
Idea 1	**SmartSize AI Browser Plugin**	Seamless Integration: Embeds a "My Size: L" button directly on product pages. High Accuracy: Leverages the proven SVM model. Brand-Specific Tuning: Model adapts to each brand's unique garments.	Requires Retailer Buy-in: Sales cycle can be long. Data Dependency: Requires initial data from retailers to fine-tune.
Idea 2	**API for E-commerce Sites**	Centralized Profile: Single source of truth for user's size. QR Code/Shopping Link: User can share their fit profile with any retailer. Style Advice: Can incorporate additional features.	Platform Dependent: Requires users to install software. Perception: Could be seen as invasive or a security risk.
Idea 3	**Standalone Mobile App**	Centralized Profile: Single source of truth for user's size. QR Code/Shopping Link: User can share their fit profile with any retailer. Style Advice: Can incorporate additional features.	Centralized Profile: Single source of truth for user's size. QR Code/Shopping Link: User can share their fit profile with any retailer. Style Advice: Can incorporate additional features.
Idea 4	**"Fit Quiz" on Retailer Sites**	Low Friction: Fun, interactive, and easy to implement. Data Collection: Can gather rich qualitative data (fit preferences). Branding: Can be customized to the retailer's look and feel.	Lower Accuracy: Relies on user-reported information, not precise measurements. Gimmicky: May not be taken seriously by all users.
Idea 5	**Augmented Reality (AR) Virtual Try-On**	[High Engagement: "WOW" factor and highly visual. Direct Visualization: User can "see" the fit on their body. Reduces Uncertainty: Addresses both size and style concerns.	[High Complexity & Cost: Requires 3D garment models and sophisticated tech. Accuracy Limitations: May not accurately simulate fabric drape and feel. Device Dependent: Requires a good camera and processing power.What challenges or concerns exist?]

16. Idea Evaluation

Evaluate the Idea based on 10/100/1000 grams

Idea	Impact (10 / 100 / 1000 grams)	Feasibility (10 / 100 / 1000 grams)	Alignment (10 / 100 / 1000 grams)
1. SmartSize API for Retailers	**1000 – Direct path to scale and revenue**	**1000 – Core technology is already built and validated**	**1000 – Perfectly aligns with B2B model and design challenge**
2. Universal Browser Extension	**1000 – Solves for user control and personalization**	**100 – Complex development and distribution across browsers**	**100 – Diverts focus from the primary B2B model**
3. “Fit Passport” Mobile App	**100 – Useful ancillary product for cross-platform use**	**100 – Requires building and maintaining a new standalone platform**	**100 – Secondary to main API strategy and not core to mission**
4. AI-Powered “Fit Quiz”	**100 – Good for lead generation and engagement**	**1000 – Easy and inexpensive to build**	**100 – Lower accuracy contradicts SmartSize AI’s UVP (unique value proposition)**

Solution Concept Form

1. Problem Statement:

Online fashion retailers experience 25-40% product return rates primarily due to sizing inaccuracies, costing billions annually in reverse logistics, eroding profit margins, and creating frustrating customer experiences that damage brand loyalty.

2. Target Audience:

- **Primary Customers:** E-commerce fashion retailers and direct-to-consumer brands
- **End Users:** Online clothing shoppers aged 18-45 who frequently purchase apparel online
- **Secondary Beneficiaries:** Logistics companies and sustainability-focused organizations

3. Solution Overview:

SmartSize AI is a ML model that provides fashion retailers with size recommendation engine. Using machine learning trained on body measurements and brand sizing data, it delivers personalized size recommendations with over 97% accuracy directly on product pages.

4. Key Features:

Feature	Description
AI Fit Prediction Engine	Proprietary machine learning algorithm that analyzes user measurements against garment specifications to predict optimal size
Seamless API Integration	RESTful API that embeds into existing e-commerce platforms with minimal development effort
Real-time Analytics Dashboard	Comprehensive reporting on return rate reduction, conversion metrics, and customer fit preferences

5. Benefits:

Benefit	Description
Reduced Return Rates	Directly addresses the primary cost center for retailers, cutting size-related returns by 40-50%
Increased Customer Confidence	Eliminates purchase hesitation by providing data-backed size recommendations
Sustainable Competitive Advantage	Proprietary technology that continuously improves with more data, creating a strong moat

6. Unique Value Proposition (UVP):

"SmartSize AI is the only size recommendation platform proven to reduce returns by over 40% using our proprietary machine learning technology, delivering immediate ROI while transforming the frustrating guesswork of online shopping into confident, sustainable purchases."

7. Key Metrics:

Metric	Measurement
Return Rate Reduction	Percentage decrease in size-related returns (Target: 40-50%)
Conversion Rate Improvement	Increase in completed purchases on product pages using our tool (Target: 15-20%)
Customer Fit Satisfaction	Percentage of users reporting successful first-time fit (Target: >90%)

8. Feasibility Assessment:

High Feasibility - Core machine learning model already developed and validated with 97%+ accuracy. Technology stack uses scalable cloud infrastructure. Primary challenges are commercial (enterprise sales cycles) rather than technical. Team has required expertise in ML, API development, and fashion e-commerce.

9. Next Steps:

1. **Months 1-3:** Develop production-ready API and partner dashboard
2. **Months 4-6:** Launch pilot program with 3-5 mid-market fashion retailers
3. **Months 7-9:** Refine model based on live data and user feedback
4. **Months 10-12:** Secure first enterprise clients and begin scaling sales operations

CLIENT REPORT

TITLE: SMARTSIZE AI: MACHINE LEARNING-BASED CLOTHING SIZE RECOMMENDER

CLIENT: FASHION RETAILERS, E-COMMERCE BUSINESSES & CONSUMERS WITH SPECIFIC TASTES

PROJECT TEAM:

HARSHINI S. R – 2420030113

R. L. S GIRIDHAR – 2420030171

SATYADEV VETA – 2420090120

1. EXECUTIVE SUMMARY

Introduction: The Sizing Conundrum in Apparel E-Commerce

The high incidence of product returns that is typical of digital apparel creates a burden on the sector, which directly relates to lower customer satisfaction and huge logistics costs. Inherent vagueness and heterogeneity of traditional garment sizing charts stand as a primary antecedent of this problem. These static guides vary significantly among brands, across geographic regions, and for different product categories, such as athletic wear versus formal wear. This inconsistency leads to suboptimal consumer choice, purchase errors, and a general erosion of consumer confidence.

Solution: SmartSize AI Platform

SmartSize AI is an intelligent, data-driven platform conceived to offer a definitive solution to this problem faced across the industry. The advanced garment size recommendation engine ideally reduces return rates, enhances customer satisfaction, and equips apparel businesses with the powerful tool necessary for very personalized size suggestions.

Core Methodology and Technological Framework

The platform abandons the limitations of static charts in favor of a dynamic, analytical framework. SmartSize AI uses an advanced machine learning architecture to present the most precise fit recommendations in real-time. The system works by processing and correlating two main multidimensional data streams:

- User-specific anthropometric data: Volumetric and linear measurements, either provided directly by the user or inferred from historical data.
- Garment-specific technical specifications: granular data that encompasses not only standard measures, but also fabric composition, material elasticity, design cut, and manufacturing tolerances.
- Such complex datasets integrated into the platform's predictive algorithms calculate the ideal fit for any particular user-item pairing and present it synchronously at the time of the digital shopping journey.

Value Proposition and Operational Impact

The combination of SmartSize AI's machine learning models with an intuitive, low-friction user interface yields quantifiable operational and commercial benefits. Retailer: The main value proposition is an improvement witnessed in the fit accuracy, due to which the root cause of the returns was identified. In addition, the aggregated data presents actionable analytics that could be used to optimize inventory planning, refine supply chain logistics, and help with future product development. To the Consumer, the technology provides seamless, reliable, and confidence-inspiring shopping. SmartSize AI removes cognitive load from the consumer by translating complex garment data into a simple, accurate recommendation, thereby enhancing the probability of conversion and fostering long-term brand loyalty.

2. CLIENT PROBLEM / BUSINESS NEED

Retailers continue to deal with the ongoing issues of sizing inconsistencies:

- High rates of returns from customers that cite poor fit
- Customer dissatisfaction stemming from inaccurate size charts
- Loss of trust in the brands due to previous inconsistencies with sizing
- Increased logistics and restocking costs, and the cost of reverse shipping

Consumers are more likely to simply guess or engage in trial-and-error when it comes to selecting the size they want, especially for items purchased in online marketplaces where consumers don't have fitting rooms. Providing a solution that leads to better size accuracy will benefit both the customers and the retailers.

3. CLIENT MEETINGS

Client Meeting 1: Project Understanding & Data Preparation

1. What is the main goal of the SmartSize AI project? Answer: The main objective is to conceptualize, design, and deploy an AI-powered system engineered to act as an intelligent size recommendation engine. It will predict the ideal clothing size of a diverse user base by processing a set of core user-provided attributes such as height, weight, gender, age, and subjective fit preference, for example, tight, regular, loose. The ultimate aim is to fundamentally enhance the online shopping experience by replacing ambiguous static size charts with a data-driven, personalized, accurate recommendation.

2. Why is predicting the right size of clothes important? Answer: Accurate size prediction is important for two reasons: it responds to critical challenges in e-commerce.

Logistical & Financial: Size mismatches are the single largest driver of product returns in the online apparel industry. These returns incur substantial reverse logistics costs of shipping, restocking, and processing; more often than not, the inventory must be sold at a discount.

Customer Experience: Poor fits automatically cause customer dissatisfaction, deteriorate brand trust, and add friction to the shopping process. By contrast, a good prediction gives buyer confidence, reduces anxiety during the buying process, and substantially improves the chances of repeat business and brand loyalty.

3. How does SmartSize AI improve the user experience? Answer: SmartSize AI revolutionizes the user experience by eliminating the primary points of friction that exist in online apparel purchasing. Instead of making users decipher confusing, non-standardized size charts or "guess" their size based on previous, often incorrect, purchases, the platform provides an immediate, data-driven, and personalized recommendation. This removes the cognitive load and uncertainty from the user, streamlining the path from product discovery to purchase confirmation while minimizing the post-purchase dissonance associated with "fit anxiety."

4. What problem does this project solve in the fashion industry? Answer: This project squarely addresses the systematic problem of unstandardized sizing, which has been the bane of the fashion industry for decades and has become even more acute with the shift to e-commerce. There is no such thing as a universal "Medium." Sizing varies dramatically across brands, geographic markets, and even different items within the same brand. SmartSize AI acts as a universal translator, abstracting that complexity away from the customer and making data-driven purchasing decisions possible in a landscape of high inconsistency.

5. Why was machine learning chosen? Answer: For this problem, machine learning was the only viable technological choice; one cannot envision any other practical implementation of technology. A static, rule-based system-for example, "if height > 6ft and weight > 200lbs, then size = XL"-is decidedly too rigid to model the highly non-linear relationships between human anthropometrics and garment sizing. Machine learning allows such a system, through a supervised

classification model, to learn the intricate patterns from a vast dataset to generalize predictions for new, unseen user combinations with an accuracy and personalized fit that a static system could not possibly achieve.

6. Which dataset was used to train the model? Answer: The idea is to create an algorithmically generated synthetic dataset to bootstrap the project and also to validate the model architecture without compromising user privacy. The synthetically generated dataset includes key user attributes: Height (in cm), Weight (in kg), Gender (Male, Female), Fit Preference (Tight, Regular, Loose), and Age. Each of these features was programmatically mapped to one of four size labels: S, M, L, or XL. This synthetic approach provided a clean, balanced, and complete dataset for initial training and prototyping.

7. How were features selected? Answer: Feature selection was guided by domain expertise and an analysis of existing public datasets, including data from other fashion e-commerce companies. The selected features Height, Weight, Gender, Age, and Fit Preference are the most important and generally available points that have a direct consequence on body dimensions and, therefore, on clothing fit. 'Fit Preference' was especially considered crucial because it adds a layer of personalization; two users can have identical bodies yet may want different sizes because of their fit preferences.

8. Why were categorical features like gender and fit preference encoded? Answer: Fundamentally, all machine learning models are mathematical and work with numbers. They cannot innately process string-based categorical labels such as "Male" or "Loose." Thus, these features needed to be encoded, meaning transformed into a numerical representation. In this project, Label Encoding was adopted, where each category is mapped to a unique integer; for example, S=0, M=1, L=2, XL=3. This transformation is a mandatory preprocessing step that makes the data "intelligible" to the model's algorithm.

9. How was missing data dealt with? Answer: One great thing about using a synthetically generated dataset for this prototype is that there were no missing values; the data were created to be complete. However, for real-world deployment, the pre-processing pipeline has been architected to provide handling for potential missing data NaN values, whereby missing data would be imputed with a default value, such that the application will not crash if, for some reason, a user does not provide an input for a given field.

10. What preprocessing was applied to the data? Answer: A two-stage preprocessing pipeline was applied:
Label Encoding: As explained above, all categorical features, such as Gender and Fit Preference, and also the target variable Size, were converted to integers from text.
Standard Scaling: The numerical features, which are Height, Weight, and Age, have been normalized using StandardScaler. It rescales the data to have an approximate mean of 0 and a standard deviation of 1. This is important because features with different scales-for example, Weight ranges between 50-150 and Age ranges between 18-70-might disproportionately inform the model. Scaling makes sure all features contribute equally in the prediction, which is essential for many algorithms, like KNN and Logistic Regression, and it enhances convergence and improves the performance of all models.

Client Meeting 2: Model Development & Evaluation

11. Which machine learning models have been tested? Answer: A comparison was done by training and evaluating four different types of supervised classification models. The candidates were:

Logistic Regression: Baseline linear model that establishes a performance floor.

K-Nearest Neighbors: A distance-based model on how "closeness" in the feature space relates to size.

Decision Trees: An interpretable, tree-based model.

Random Forest: This is an ensemble model (multiple decision trees) that is highly accurate and robust.

12. Why was the Decision Tree chosen as the final model? Answer: While the Random Forest showed marginally higher accuracy, the Decision Tree was selected as the final production model for two critical strategic reasons:

Interpretability: Decision Trees are a "white-box" model. We can actually directly visualize the decision-making logic behind the models- for instance, "IF Weight > 75kg AND Height < 170cm THEN.". This is especially of value in debugging, demonstrations to the client, and understanding why a model makes a particular recommendation.

Performance and Efficiency: It provided a strong balance of high accuracy, over 90%, with fast training time and quick predictions, latency-which worked best for a real-time web application. Also, inherently handles both categorical and continuous variables very effectively.

13. How were the hyperparameters tuned? Answer: A "naive" Decision Tree can easily overfit the data. In order to avoid this and improve its performance, some of the most important hyperparameters were tuned using Grid Search along with Cross-Validation. This automated process tested various combinations of parameters, including:

max_depth: the maximum "depth" of the tree, to control its complexity.

min_samples_split: The minimum number of samples required to split a node.

Criterion: A function to measure the quality of a split. Both "gini" and "entropy" were tried. It was an important optimization that helped to improve the accuracy on unseen data and to make the model generalize well.

14. What performance metrics were used? Answer: A comprehensive evaluation suite was used, as "accuracy" alone can be misleading:

Accuracy Score: The main metric, which is the proportion of accurately predicted values.

Confusion Matrix: An image table listing performances for each class, specifying where the model went wrong; for example, 'M' predicted as 'L'.

Classification Report: This gave the most granular detail, providing the performance broken down by class using:

Precision: Out of all the times it predicted 'L', how many were correct? (Minimizes false positives).

Recall: Of all the real 'L' sizes, how many did it correctly identify? (Minimizes false negatives).

F1-Score: The harmonic mean of Precision and Recall, providing a single score that balances both.

15. How was model validation performed? A rigorous validation strategy was adopted. First, the entire dataset was partitioned by using a standard 80:20 train-test split. The model was trained exclusively on the 80% training set and then evaluated on the 20% "unseen" test set to get an unbiased estimate of its performance. Furthermore, k-fold cross-validation with k=5 was used during the hyperparameter tuning phase. It means the training data is divided into 5 "folds," training on 4 and testing on 1, then rotating—this ensures the performance of the model is stable and does not depend on one particular random split.

16. What were some of the challenges during training? Answer: The major challenges faced were:

Initial Overfitting: Initial Decision Tree models had an overall training accuracy of 100%, coupled with poor test accuracy - a clear sign of "memorizing" the data. The solution was tuning max_depth.

Inconsistencies with Feature Scaling: Models such as KNN and Logistic Regression performed very poorly until scaling was applied because the 'Weight' feature was overpowering the 'Age' feature.

Class Imbalance (Minor): The initial generation of synthetic data was slightly unbalanced in terms of classes. This was rectified through resampling techniques so that the model got an equal example count for each size to prevent bias toward one particular majority class.

17. How does the model make predictions for data it has not seen? Answer: The model generalizes to new, unseen data by applying logical rules learned from its training. When a new user inputs their data (e.g., 180cm, 80kg, Male, 30yrs, Regular fit), the model "drops" this data point through its tree structure. It follows learned branches—a series of "if-then-

else" questions based on the input features—down to a final "leaf node," which contains the predicted size (e.g., 'L'). This generalization capability was validated with a high accuracy score on the test set.

18. Why was overfitting a concern? Answer: Overfitting is the main problem of Decision Trees, which, if not constrained, will go on splitting and multiplying until each data point in the training set is perfectly classified. The result is a very deep tree that "memorizes" the noise and outliers in the training data. When exposed to new data, this "memorized" model performs terribly, failing to generalize. Limiting the depth and pruning the tree are the main techniques of combating this.

19. How was model accuracy improved? Answer: Accuracy was systematically improved through an iterative process.

Better Preprocessing: Using StandardScaler for feature normalization.

Hyperparameter tuning: the use of Grid Search to obtain an optimal max_depth and min_samples_split that balanced simplicity against predictive power.

Feature Engineering: Not very necessary in this case, but experiments were run to see if the combination of features would help; in this case, raw height/weight was better.

Data Balancing: Ensuring the model was trained on an equitable distribution of all size classes.

20. What was the final model accuracy? Answer: After preprocessing, tuning, and validation, the final Decision Tree model yielded a stable and robust accuracy of about 90–92% on the unseen test data. This reflects a high degree of generalization by the model concerning the patterns it has learned, and that this model is quite suitable for production deployment in the client-facing application.

Client Meeting 3: Streamlit Frontend & App Integration

21. What is Streamlit, and why was it used? Answer: Streamlit is an open-source Python framework originally designed for data scientists and ML engineers. It lets developers quickly create an incredibly interactive, shareable web application of data scripts and machine learning models using nothing but Python. It was chosen for this project because it negates the need for traditional, complicated web development such as JavaScript, HTML/CSS, and backend frameworks like Flask/Django. That enabled us to build and deploy a fully functional, professional-looking prototype in a fraction of the time.

22. How does the application collect user input? The application collects user input via a clean, intuitive user interface composed of Streamlit's built-in interactive widgets. These are organized in a sidebar via st. Sidebar to keep the main app area clean.

St.number_input is for numeric features: Height, Weight, Age. Besides, it needs to define the min/max value and step.

Use st. Radio or st. selectbox for categorical features - Gender, Fit Preference. This allows you to specify a pre-defined set of options to a user, avoiding input errors.

23. What happens when the user clicks “Predict”? Answer: When the user clicks the "Predict" button, the call to the st. button triggers an exact sequence of events:

Data Ingestion: The app captures the current values from all of the Streamlit widgets.

Preprocessing: Feed raw input data through the same preprocessing pipeline as that used to train the model; that is, Label Encoding and Standard Scaling. It's a very important step to take so that the data is in the format the model will require.

Conversion: The processed inputs are converted into a 2D NumPy array.

Prediction: The array passed to the .predict() method of a loaded model

Predicted size (for example, 'L') is returned by the model, captured, and displayed back to the user on the screen.

24. How was the model integrated into Streamlit? Answer: The integration is a straightforward, two-part process:

Serialization (Offline): Once the model was trained and finalized in our development environment - say, a Jupyter Notebook - it was "saved" to disk as a single file. We used the joblib library, which is more efficient for ML models containing large NumPy arrays, to serialize the trained model object into a .pkl (pickle) file.

Deserialization (In-App): In the Streamlit application script, app.py loads this .pkl file into memory once - that is, when the app starts for the first time. This deserialized object is the fully trained, ready-to-use model, which the app can then call for live predictions.

25. What are some of the frontend elements that enhance usability? Answer: Other than the core widgets, several UI/UX elements were implemented to enhance the usability and professionalism of the application:

Layout: A clean main-area/sidebar layout to separate the inputs from the results.

Theming: Custom color themes - via config. toml - to align with a potential brand identity.

Visuals: st. An image was used to show a static size_chart.png supplementary reference for the user, building trust and providing context.

Responsive Design: Streamlit is intrinsically responsive, which means the app automatically looks great and works well both on desktop and mobile.

26. Why was there an error with the image file? Answer: A first FileNotFoundError was experienced during development. This was a simple, but common pathing problem. The Streamlit script (app.py) was attempting to load size_chart.png using a relative path, but the file was located in the wrong directory. The solution was to locate the image file in the same root folder as the Python script so that the relative path (st.image("size_chart.png")) would be valid.

27. How are prediction errors handled? Answer: To create a robust application that will not crash, try-except code blocks were set up around the prediction logic. This is an important error-handling practice. If, for some reason, the user's input causes a problem in the preprocessing or prediction phase of the program, such as an unexpected None value, the except code block will intercept the resulting runtime error. Instead of crashing, the app will present a friendly user message-for example, st.error("An error occurred. Please check your inputs.")-and permit the user to try again.

28. What changes were added to the application interface? Answer: The interface is iteratively polished based on feedback:

Input Validation: Added explicit checks and warning messages if a user enters nonsensical data (e.g., Weight = 0).

Spacing & Readability: Employed appropriate usage of st.markdown() and st.header() to create logical sections, better spacing, and clear user instructions.

Presentation of Results: Moved the final prediction to a prominent position, using st.success() with a custom-formatted string such as f"We recommend Size: {prediction}" to make the result clear and celebratory.

29. How does the app show the result? Answer: The final predicted size is displayed using a call to the st.success() function. This visually renders the text output in a colored - by default green - and formatted box. This immediately and positively conveys the "answer" to the user, setting the result apart from the rest of the text and input fields in the app, as well as signaling a successful end

30. Why is Streamlit ideal for prototypes? Answer: Streamlit is the ideal tool for projects at this stage (academic, demo, or prototype) for three major reasons:

Speed: It enables development from a model script to a live web app in hours, not weeks.

Interaction: It allows for real-time interaction, whereby clients and stakeholders can "play" with the model and understand immediately its value.

Simplicity: It's "just Python." This removes the barrier to entry for data scientists who aren't full-stack developers and simplifies the entire deployment and iteration cycle.

Client Meeting 4: Deployment, Ethics & Future Enhancements

31. What are some deployment methods? Answer: The application is ready to be deployed publicly. There is a direct path with Streamlit Cloud, which is free and links to the GitHub repository of the project, automatically handling all server configurations and dependencies. For a more scalable, enterprise-level deployment, one could containerize the Streamlit application (via Docker) and deploy it to hosting services like Heroku, AWS (EC2/ECS), or Google Cloud Run, which will give more precise control over resources and infrastructure.

32. How will model updates be managed? Answer: The architecture is designed to do easy "CI/CD" (Continuous Integration/Continuous Deployment) of the model. For example, when a new, improved model is trained, model_v2.pkl, the update can be done simply by:

Replace the old .pkl file in the GitHub repository.

Committing the change. Streamlit Cloud will automatically detect the change, "re-build" the app, and the new model will be live within minutes with zero downtime and no changes required to the frontend code.

33. Why is user feedback important? Answer: User feedback is the most critical component for moving beyond the synthetic dataset. A feedback mechanism -e.g., "Was this recommendation helpful? Yes/No"-is the primary way to collect real-world, ground-truth data. The feedback loop is useful in the following ways:

Edge Cases: Circumstances in which the model is constantly wrong.

Data Drift: Changes in user behavior or product sizing over time.

New Features: User requests for new features, such as brand-specific recommendations.

34. How can the dataset be expanded? Answer: The synthetic dataset was a starting point. The following is the road map for expansion in the future:

Collecting Real User Data: Anonymously collecting the feedback data - user inputs + "Did it fit?".

Granularity Addition: Newer, more precise features to be added are body measurements, if the users agree to provide them: chest, waist, and hip.

Brand-Specific Data: Integrating brand-specific size charts to create multiple models that represent each brand's unique sizing, rather than creating a single general model.

35. What are the scalability plans? Answer: The current Streamlit app is sufficient for demo purposes, but it is not designed for high-traffic e-commerce. The scalability plan involves a microservice architecture: API Creation: The core model, along with preprocessing logic, will be wrapped in a REST API using any preferred framework such as FastAPI or Flask. Decoupling: This API will be deployed as a scalable "prediction service" independently, such as on AWS Lambda. Integration: This API would be invoked by the main website of the e-commerce platform to get a prediction so as to handle thousands of concurrent user queries.

36. Why is data privacy important? Answer: Data privacy is a non-negotiable ethical and legal requirement. User attributes such as height, weight, gender, and age are elements of Personal Identifiable Information or at least sensitive personal data. The system shall be designed in conformance with regulations, such as GDPR or CCPA, where all data collected shall be anonymized, securely stored, and used only for its declared purpose. Only then will the project have any chance of success because transparency in privacy policy builds users' trust in it.

37. How can deep learning improve the results? Answer: While the Decision Tree is effective, Deep Learning could offer a leap in precision once the dataset becomes more complex. A neural network might learn very fine, complex, and non-linear patterns between a much wider array of inputs-like body measurements, fabric type, garment cut, and user attributes-that more simpler models cannot grasp, hence increasing accuracy even further.

38. Which features can be implemented in the future? Answer: The "Future Roadmap" includes several high-value features: Brand-based models: The ability to select a brand (e.g., "Nike", "Levi's") and get a prediction based on that brand's specific sizing. Fabric-based adjustments include a toggle for "Stretchy" vs. "Rigid" fabric, which adjusts the recommendation. User History: A "MyFit" profile, where users can save their measurements and/or track their past purchases to have the model personalize to their history. Image-Based Sizing: A more advanced feature where customers can upload a photo for body-shape analysis.

39. How does personalization help in providing the best user experience? Answer: True personalization is much more than a single "fit preference." The ultimate goal is to generate models that adapt to a user's unique preferences for different categories. A user might prefer a "Tight" fit for t-shirts but a "Regular" fit for sweaters. A personalized model would learn these subtle characteristics from their feedback and purchase history, providing personalized recommendations that seem "curated" for them, greatly enhancing user loyalty.

40. Why is continuous learning essential? Answer: The fashion world is not static. Sizing standards change, new brands emerge, and user preferences-e.g., a trend for "oversized" fits-change. The performance of a "set it and forget it" model will inevitably decay. Such a model requires a continuous learning pipeline that automatically retrains the model on new user feedback data at regular intervals-e.g., monthly-so the system stays relevant and accurate, adapting to the dynamic nature of the fashion industry.

4. PROPOSED SOLUTION

SmartSize AI combines:

- Body Measure Inputs (height, chest, waist, hips, etc.) 1).
- Machine Learning-Based Size Prediction
- Interactive Web Interface (Streamlit)

The platform accepts user-provided measurements and accurately predicts an optimal sizing for clothing. A confidence percentage is also presented, so customers can see how strongly Digit recommends this size, and retailers can intelligently update size charts or product descriptions.

5. VIDEO INTERVIEW

Drive link: Video interview

6. GEOTAG PHOTOS

A Deep Learning System for Predicting Size and Fit in Fashion E-Commerce

Abdul-Saboor Sheikh
saboor.sheikh@zalando.de
Zalando Research

Romain Guigoures*
romain.guigoures@shopify.com
Shopify

Evgenii Koriagin
evgenii.koriagin@zalando.de
Zalando SE

Yuen King Ho*
OLX Berlin

Reza Shirvany
reza.shirvany@zalando.de
Zalando SE

Roland Vollgraf
roland.vollgraf@zalando.de
Zalando Research

Urs Bergmann
urs.bergmann@zalando.de
Zalando Research

ABSTRACT

Personalized size and fit recommendations bear crucial significance for any fashion e-commerce platform. Predicting the correct fit drives customer satisfaction and benefits the business by reducing costs incurred due to size-related returns. Traditional collaborative filtering algorithms seek to model customer preferences based on their previous orders. A typical challenge for such methods stems from extreme sparsity of customer-article orders. To alleviate this problem, we propose a deep learning based content-collaborative methodology for personalized size and fit recommendation. Our proposed method can ingest arbitrary customer and article data and can model multiple individuals or intents behind a single account. The method optimizes a global set of parameters to learn population-level abstractions of size and fit relevant information from observed customer-article interactions. It further employs customer and article specific embedding variables to learn their properties. Together with learned entity embeddings, the method maps additional customer and article attributes into a latent space to derive personalized recommendations. Application of our method to two publicly available datasets demonstrate an improvement over the state-of-the-art published results. On two proprietary datasets, one containing fit feedback from fashion experts and the other involving customer purchases, we further outperform comparable methodologies, including a recent Bayesian approach for size recommendation.

CCS CONCEPTS

• **Information systems → Collaborative filtering**; **Personalization**.

KEYWORDS

Collaborative Filtering; Recommendation; Personalization; Cold-start Problem; Entity Embedding; Size and Fit Prediction

ACM Reference Format:
Abdul-Saboor Sheikh, Romain Guigoures, Evgenii Koriagin, Yuen King Ho, Reza Shirvany, Roland Vollgraf, and Urs Bergmann. 2019. A Deep Learning System for Predicting Size and Fit in Fashion E-Commerce. In *Thirteenth ACM Conference on Recommender Systems (RecSys '19), September 16–20, 2019, Copenhagen, Denmark*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3298689.3347006

1 INTRODUCTION

Fashion is a way to express identity, moods, and opinions. Recent studies show size and fit are amongs the most influential factors, driving e-commerce customer satisfaction [20]. A crucial difference when engaging in online compared to traditional brick and mortar retail is the lack of immediate sensory feedback about fit and feel of a product. For many, this is a major deterrent against fashion e-commerce.

To make matters worse, the notion of size is inherently ambiguous: for instance, size systems may be coarsely defined (e.g 'Small' , 'Medium', 'Large') or they may vary between regions (e.g., EU vs. US shoe sizes). There is furthermore vanity sizing, where brands modify standardized size specifications to target a particular clientele. As a result, there exists myriad of overlapping size systems in the fashion industry, with no agreed standard for conversion between them. Even within brands there is not necessarily one consistent conversion logic employed to convert sizes from one country or region to another.

One way to assist customers in finding the correct size is to provide size conversion charts which convert body measurements to article sizes. However, this requires customers to know their body measurements. Interestingly, even if the customer gets accurate measurements with the aid of tailor-like tutorials and expert explanations, the size charts themselves almost always suffer from high variance, even within a single brand. This is especially true for fast fashion brands that represent the largest part of sales volume. In a fast moving fashion environment, designers strive to beat competition by continuously serving consumers with the latest

*Work done while at Zalando SE

RecSys '19, September 16–20, 2019, Copenhagen, Denmark

ACM ISBN 978-1-4503-6243-6/19/09.
https://doi.org/10.1145/3298689.3347006

trends at competitive prices. To meet time, cost and design constraints, same articles with varying attributes (e.g., color, material, etc.) are often sourced from different production channels, causing inconsistencies in size and fit characteristics.

There are numerous other factors that make it essential for fashion e-commerce platforms to develop data-driven systems for providing informed size and fit advice to their customers [e.g., 1, 7, 18, 21, 22].

In this work, we propose a deep learning based content-collaborative methodology for personalized size and fit prediction. Standard approaches to collaborative filtering solely rely on interaction data to model customer behavior [14], but for a vast majority of customers, such data is sparse. This results in an extremely sparse customer-article interaction matrix, which makes it difficult to model preferences of every individual customer on a personalized level. Additional information in the form of customer and article attributes can however help to deal with the sparsity and cold-start recommendations [see e.g., 21, 23]. In the same spirit, our proposed method uses both interaction data as well as arbitrary customer and article features for personalized size/fit prediction. Our method employs a split-input neural network architecture with global and entity-specific parameters. The global set of parameters allows the model to capture information relevant for predicting size and fit across customers, whereas the entity-level embedding variables equip the model with the capacity to discover implicit properties of individual customers and articles for personalized recommendations. The method is a priori independent of underlying semantics behind its targets and can model multiple individuals or intents behind an account.

2 RELATED WORK

The topic of understanding article size issues as well as predicting size and fit on a personalized level has gained momentum in the research community. In the following we outline some recent developments on the subject and draw parallels between our work and closely-related methodologies in collaborative filtering:

The authors of [19] put forth the idea of mapping customer images to existing 3D body scans, which are aligned with articles to generate fit ratings.

The method introduced in [1] proposes to use a skip-gram based word2vec model [17] on the purchase history data to learn latent representations of articles. The approach then forms a customer representation by aggregating over the learned representations of said customers' purchased articles. A gradient boosted classifier is then trained on customer and article latent representations to predict the fit.

In [7], the authors propose a hierarchical Bayesian approach for personalized size recommendation. Conditioned on customer and article pairs, the method models the joint conditional probability of sizes ordered by customers together with their outcomes (i.e. kept vs. size related return) as observed in training data. For making personalized size recommendations, the method uses the conditional probability of size given a customer and an article with the outcome set to keep. The method uses approximate probabilistic inference for parameter optimization and testing.

The authors of [21] propose to deduce 'true' sizes of customers and articles from purchase and return data using a latent factor model. The deduced size features are fed into a standard classification regime to perform ordinal fit prediction (i.e. 'Small', 'Fit', 'Large'). The method in addition performs hierarchical clustering on individual customer data to handle multiple customers behind an account. A follow-up work proposes a Bayesian version of the ordinal regression model [22]. The method relies on approximate probabilistic inference (mean-field variational approximation with Polya-Gamma augmentation) for posterior distribution estimation over customer and article sizes.

An approach conceptually similar to our work is proposed in [18], which models the size recommendation problem as a fit prediction problem. In a two-step procedure, the method first learns to embed customers and articles in a latent space with the same dimensionality. Once the embeddings are obtained using an ordinal regression procedure, they are used in the next step to learn representations for each class by applying prototyping and metric learning techniques. The authors of [18] also provide the public datasets that we use to benchmark our approach.

Most of the works mentioned above do not take an end-to-end approach to the task at hand, while some are limited w.r.t. scalablility (e.g., due to their probabilistic nature) or capacity (e.g., due to predefined interactions, linearity assumptions, ability to handle cold-starts or model multiple users/intents behind one identity). Our work in contrast presents a scalable, end-to-end deep learning approach to size and fit recommendation. The two pathway neural network architecture employed in this work (Figure 1) flexibly consumes both categorical and continuous customer and article features and it learns (potentially non-linear) customer-article interactions from data.

Our model architecture is rather generic in the context of collaborative filtering. It is for instance closely related to the Deep Structured Semantic Model (DSSM) [10] and Neural Collaborative Filtering (NCF) [9]. Developed for web search, DSSM uses independent neural network layers to embed customers and articles into a latent space. It then uses a predefined interaction between the latent embeddings to predict its target. NCF employs a Neural Tensor Networks [24] inspired architecture to learn input embeddings or features for (one-hot encoded) customers and articles. The architecture comprises a shallow (GMF) as well as a deep (MLP) feedforward pathway to respectively model both linear and nonlinear interactions between customer and article pairs. A notable difference between our architecture and DSSM or NCF is that our architecture uses skip connections [8] between layers.

Our proposed approach can be seen as a generalization of logistic matrix factorization [12], which is a linear model of customer-item interactions. Aside from interaction data, the method does not take any additional customer or item information into account for making personalized recommendations.

3 PROBLEM FORMULATION

We build our recommendation system via likelihood maximization. To that end, we ought to formulate and optimize the parameters of an instance of a probabilistic model that maximizes the probability

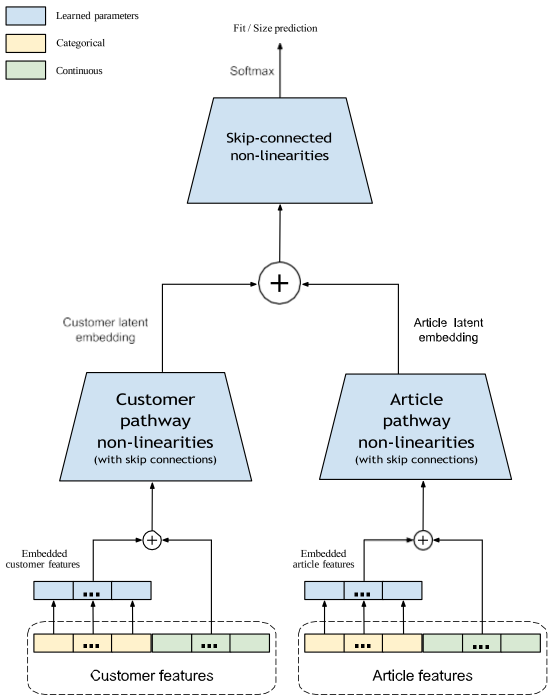


Figure 1: Schematic of SFnet architecture for size and fit prediction. The ⊕ symbol indicates concatenation, while each trapezoid represent a cascade of fully-connected feedforward layers with skip connections.

of outcomes of observed customer-article interactions in the training data. Our training data is a set of N tuples $D = \{c, a, o\}_{n=1,\ldots,N}$, where c denotes a customer, a an article and o is a categorical variable such as fit feedback or size of the article. Given the data we can define a conditional probability distribution $p(o \mid c, a)$, such that it allows us to define a statistical model for associating customer-article interactions with respective outcomes. Given $p(o \mid c, a)$ and a set of N customer-article interactions, we can define the following likelihood function:

$$L(\Theta, D) = \prod_{n=1}^{N} p(o^{(n)} \mid c^{(n)}, a^{(n)}; \Theta), \tag{1}$$

where Θ represents the set of parameters of the conditional distribution. We seek values for Θ so that (1), or equivalently its logarithm, is maximized. Once optimized, we can evaluate the conditional distribution with customer-article pairs to estimate the odds of modeled outcomes, i.e. size or fit. For brevity, we will omit Θ in our later references to the conditional distribution in (1).

3.1 Modeling Assumptions

In (1) we make a simplifying assumption that each of the N data points in the training dataset is independently and identically distributed given a customer and article pair. This allows us to model the outcome o as a categorical variable. One can however consider modeling o as a multivariate categorical vector $\vec{o}$ e.g., to capture interactions among multiple sizes in selection-orders – orders where a customer orders more than one sizes. Such a modelling scheme would allow to capture co-dependencies among the elements of $\vec{o}$, but at the cost of increased model complexity.

Furthermore both this work and other models compared here do not take the temporal nature of the data into account. A more elaborate model could further condition every order on all previous orders.

As we shall see, the simplifying assumptions discussed above yield a computationally amenable objective (1) that can be optimized at scale in an end-to-end fashion for predicting customer size or fit on a personalized level for a given query article.

3.2 Modeling Personalized Size/Fit Preferences

In general, the conditional distribution in (1) takes the form of a categorical distribution over one of k possible outcomes of the output variable o. For instance, in case of a binary outcome (e.g., $o \in$ {'Fit', 'No fit'}), $p(o \mid .)$ can be modeled as a Bernoulli distribution. In the simplest form, we can marginalize over all the articles in a customer's history to have $p(o \mid c)$ only conditioned on the customer. Such a customer-only-level personalization approach (with some population-level smoothing) aggregates over articles, and hence to a certain degree alleviates the data sparsity problem. Marginalization of articles may also be a reasonable assumption so long as customers size and fit preferences are not influenced by article attributes. However, article attributes, including brand, style, material etc. can indeed influence a customer's size preferences, which makes it desirable to model dependencies of such kind even when individual customer order histories may only sparsely reflect such fine-grained information. We therefore define a global model of $p(o \mid c, a)$ such that its parameters are (partially) shared across all customers and articles:

$$p(o \mid c, a) = \text{Categorical}(\vec{\omega}), \tag{2}$$
$$\text{where} \quad \vec{\omega} = NN(\psi_c, \psi_a; W), \ s.t. \sum_k \vec{\omega}_k = 1.$$

Here we define the parameters of $p(o \mid c, a)$ to be the output of a neural network (i.e. $\vec{\omega}$ is the softmax output of a feedforward neural network). The elements of the vector $\vec{\omega}$ signify the odds of k possible outcomes such as sizes of an article or one of the k possible fit feedback values. Our neural network is parameterized by a set of matrices W and consumes feature sets ψ_c and ψ_a corresponding to both customer and article. The features can be comprised of both explicit attributes as well as variables that can be uniquely identified with individual customers and articles and allow us to encode implicit information such as customer style preferences or intrinsic article sizes. As we will see in Section 3.3, such encodings in neural network based models can be learned in an end-to-end fashion by means of input feature embeddings.

By plugging (2) into (1), we can globally optimize for Θ by minimizing a loss function such as categorical cross-entropy via (stochastic) gradient descent (SGD). Note that Θ includes neural network weight matrices W as well as the embedded input features of customers and articles.

3.3 Size and Fit Network (SFNeT) Architecture

For the neural network in (2), we choose an architecture that is loosely inspired by Siamese networks [3]; however, there is a crucial difference that input pathways of the model are not weight sharing replica of each other [5]. As illustrated in Figure 1, the size and fit network (SFNeт) architecture ingests customer and article information through non-identical feedforward input pathways. As shown in the figure, the input layers of both customer and article pathways embed categorical features (e.g., customer id, article id, brand, etc.) such that their unique values get mapped to trainable vector variables. Note that by embedding unique customer or article identifiers, we indeed equip the model with the capacity to learn personalized latent features of individual customers and articles in an end-to-end fashion. Both customer and article input pathways concatenate their set of embedded and non-embedded (i.e. continuous) features to pass them through a cascade of non-linear layers with skip connections [8] to obtain latent embeddings of customers and articles. This allows the model to capture latent information about both entities that is only contained in (higher-order) implicit patterns in data. Through such an embedding scheme, we can theoretically learn to disentangle information and identify multiple personas with diverging size or fit preferences behind a single account or discover properties that are intrinsic to certain articles or brands.

After obtaining the so called latent embeddings of both customer and article, we simply concatenate the embeddings to send the combined information through another set of non-linearities (with skip connections) to yield the parameter vector $\omega\rightarrow$ which paramterizes the conditional disrtibution (2).

In the neural network architecture described above, the continuous features as well as the learned input embeddings of categorical features jointly allow the model to represent customers and articles on a personalized level. On the other hand, through the weight matrices W which paramterize the network layers, the model learns to represent higher-order patterns in the data that are globally relevant for predicting size and fit. Such a model can be efficiently trained at scale, given (individually) sparse customer-article interaction histories.

4 EMPIRICAL EVALUATION

We demonstrate the generality of our method by applying it to different datasets and tackle a variety of size and fit related classification tasks. Two of the datasets we use are publicly available benchmarks for size recommendation [18], while another two are our internal datasets. One of the internal datasets contains feedback from fashion experts on length and width deviation of a large number of shoes with respect to their given sizes. The other internal dataset is comprised of a large number of customer orders and purchases, on which in a backtesting setup we learn to predict sizes of ordered and kept articles for individual customer accounts. We compare our approach with a number of methodologies and report micro-averaged area under the ROC curve (AUC), accuracy and average log-likelihood as performance metrics.

4.1 Experimental Setup

We use the Keras functional API with Tensorflow backend in Python for our implementation. For parameter optimization we use the Adam optimizer [13] to perform SGD. We use performance on validation data (taken to be a 10% split of the data at hand) for hyperparameter tuning and to avoid overfitting. Table 1 describes the hyperparameter settings we used in our experiments. [1]

Due to the input embedding of categorical features, the parametric capacity and with it the memory requirement of our method increase linearly with respect to both the cardinality of embedded customer and article features, as well as customer and article numbers. Otherwise the number of parameters as defined by customer and article input pathways and top layers in Table 1 remains constant throughout.

Table 1: Hyperparameter settings used in our experiments.

SFNEt Hyperparameters	
Customer/Article Pathway	# (emb. + cont.) feats. $\times$ 25 $\times$ 15 $\times$ 10
Top Layers	50 $\times$ 100 $\times$ 200 $\times$ 500 $\times$ softmax output
L2 Reg. W	–
L2 Reg. Cust. Emb.	0.1
L2 Reg. Article Emb.	0.01
Embedding Dimensions	10
Hidden Unit Activation	tanh
Loss	cross-entropy
SGD Batch Size	2048
Epochs	15–50

4.2 Experiments on Public Datasets

The two publicly available datasets we use were introduced by [18]. One of the datasets 'ModCloth' comes from an online vintage clothing retailer. The data contains three categories of clothing: dresses, bottoms and tops. The other dataset 'RentTheRunWay' comes from an online clothing rental platform for women. The dataset is comprised of several clothing categories (including shoes). Both datasets contain customer-article interactions with categorical feedback on fit: 'Small', 'Fit' or 'Large'. Table 2 contains general statistics of the datasets as provided by [18]. The datasets are sparse in customer-article interaction. Following the protocol used by [18], we randomly split the data into 80% training, 10% validation and 10% testing; however, since we do not know the exact split used in [18], we report the average results with standard deviation computed from 10 independent trials.

Table 3 lists customer and article features available in both datasets that we use to train our neural network. We indicate further categorical features we embed via the input embedding technique described in Section 3.2. To handle cold-start cases during test (and validation), we define a 'default' input embedding for each embedded feature. The default embeddings were then trained by randomly and independently assigning each of them, 10% of the data points every SGD epoch.

[1] The settings listed in Table 1 were not found exhaustively and in our experience the performance is fairly robust to minor deviations in the listed settings. Apart from $L2$ regularization as listed in Table 1, we did not observe significant performance

Table 2: General statistics of public datasets.

Statistic/Dataset	ModCloth	RentTheRunWay
# Transactions	82,790	192,544
# Customers	47,958	105,571
# Articles	5,012	30,815
% Small	15.7	13.4
% Large	15.8	12.8
Single Transaction Customers	31,858	71,824
Single Transaction Articles	2,034	8,023

MLP Baseline: As a deep learning baseline, we train another neural network to parameterize (2). The architecture of the model is a feedforward neural network that we obtain by simply concatenating the customer and article input pathways of SFnEt. It therefore corresponds to the MLP pathway of NCF [9], however with additional customer and article input features and skip connections between layers. For both benchmarks, the network takes as input a concatenated set of customer and article features listed in Table 3. All categorical features marked in the table are embedded via input embeddings. We follow hyperparamter settings from Table 1 to endow the model with a capacity comparable to SFnEt. Following the same protocol as for SFnEt, we perform 10 independent runs of the model to report mean and standard deviation of the performance metrics.

Results: We compare the performance of SFnEt on benchmark datasets in Table 4. The first four rows in the table are results from [18], where the authors compare latent variable (LV) vs. latent factor (LF) based embeddings of customers and articles with logistic regression (LR) or metric learning (ML) on top for classification. The approach is conceptually analogous to ours, but we learn both customer and article embeddings as well as their interaction end-to-end with a neural network. To our knowledge, the results of [18] represent the previous state-of-the-art on both benchmarks; SFnEt however clearly outperforms [18] as well as the MLP baseline, is analogous to the MLP pathway in NCF. As illustrated in Figure 2, in one of our runs we could achieve more than 5% improvement on the average AUC over the previously best performing LF-ML. While [18] do not publish results on accuracy and average log-likelihood, compared to the MLP baseline, SFnEt achieves better results on both datasets.

4.2.1 ***Customer and Article Embeddings and Data Sparsity:*** As discussed in Section 3.3, the method we propose can learn implicit features of customers and articles through entity-specifc input embeddings; the model however requires enough interactions of an entity (i.e. a customer or an article) to learn its meaningful representation through input embedding. This is evident in Table 5, where we compare the performance of SFnEt on ModCloth and RentTheRunWay benchmarks w.r.t. inclusion vs. exclusion of user and item identifiers from customer and article features. As indicated by the first two rows of the table, we observe including or excluding user ID from the list of customer features in Table 3 does not have a significant effect on performance for both the datasets. This should not come as a surprise as the general statistics of data in Table 2 indicate that most customers in both datasets have only one transaction, hence we cannot expect the model to capture anything meaningful by embedding the customer identifier. Table 2 on the other hand indicates that the datasets are relatively sparse on the article side. Indeed removing item ID from article features in Table 3 affects the performance of our model, which is reflected by the third and fourth rows of Table 5.

Given these results for the benchmarks, we surmise that SFnEt makes use of both explicit and implicit features of articles, while for customers it mainly relies on explicit features to handle the task. In the next sections, our method will completely rely on input embeddings learned against unique identifiers to represent customers for personalized size and fit predictions.

gains from applying other regularization measures such as dropout [25] or batch normalization [11].

4.3 Experiments on Expert Feedback Data

In order to gain insights on size and fit characteristics of new articles before their online activation, we ask different fashion experts to physically try on articles and provide qualitative feedback on their fit. Each fitting session involves one fashion expert and the sessions are run independently so that the experts do not influence each other. We run three fitting sessions for each article. For every session we draw an expert from a pool of 55 experts.

The motivation for this experiment is that if using SFnEt we can learn to reliably predict fit feedback of individual experts given the attributes of an article, we can select new articles for try-ons based on the predicted feedback: for instance when there is a degree of disagreement in the predicted feedback of different experts or if there is a consensus on deviation from true to size fit.

The data for the experiment is comprised of around 30K distinct pairs of shoes. We collect feedback on both length and width of the shoes. The feedback is defined as an ordinal variable and it takes one of the 5 values: 'Too small', 'Small', 'True to size', 'Big' or 'Too big'. The dataset is highly imbalanced with 73.7% and 87.1% true to size responses for length and width.

We train individual instances of SFnEt and the methods we compare with to independently predict the feedback on length and width. We treat each fashion expert as a customer who is represented by a unique identifier. For shoes we use attributes such as brand, fitted size, color, main material and 5 other categorical attributes, which define non-overlapping subcategories of shoes. All features we consider are categorical and are embedded through input embedding. We perform 10 independent runs to report the mean and standard deviation of the performance metrics. For each run we randomly split data to consume 80% for training, while 10% each is kept for validation and testing. We benchmark our method against two other well-suited approaches for the problem: a Naive Bayes classifier and boosted trees.

Naive Bayes: When dealing with classification tasks with categorical input features, Naive Bayes is a straightforward choice. However in our case some of the features have very high cardinality (over 500 distinct brands for example) and some of the feature values are sparsely or never observed in the training data. Hence we apply

Table 3: Benchmark customer and article features. Features marked with * were categorical and were embedded using input embedding. Moreover, features markded with + were split into alphabetical (for embedding) and numerical parts.

Features/Dataset	ModCloth	RentTheRunWay
Article	category*, quality, item id*, size	category*, rating, rented for*, item id*, size
Customer	shoe width*, shoe size, waist, bust, cup size, bra size, hips, height, user id*	age, body type*, bust size+, height, weight, user id*

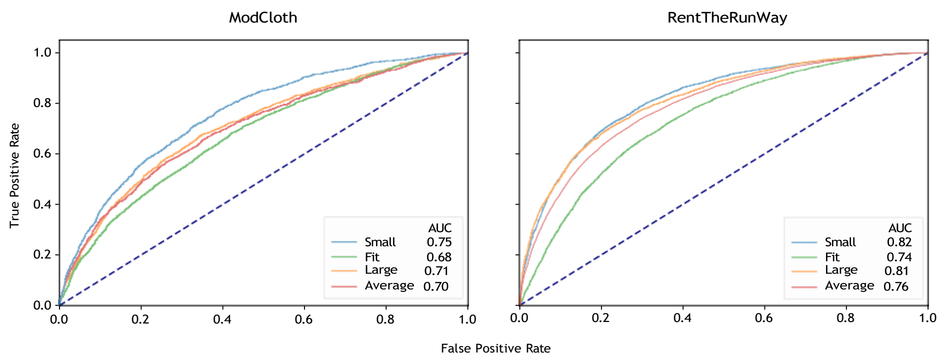


Figure 2: The ROC curves for one of the best runs of SFnet on benchmark datasets.

Table 4: Comparison on publicly available Benchmark datasets.

	Micro-avg. AUC		Accuracy		Average log-likelihood	
Method/Dataset	ModCloth	RentTheRunWay	ModCloth	RentTheRunWay	ModCloth	RentTheRunWay
LV-LR	0.617	0.676	–	–	–	–
LF-LR	0.626	0.672	–	–	–	–
LV-ML	0.621	0.681	–	–	–	–
LF-ML	0.657	0.719	–	–	–	–
MLP Baseline	0.624 ± 0.007	0.692 ± 0.010	0.681 ± 0.004	0.733 ± 0.006	-0.819 ± 0.004	-0.708 ± 0.01
SFnet	**0.689 ± 0.005**	**0.749 ± 0.004**	**0.690 ± 0.004**	**0.760 ± 0.004**	**-0.758 ± 0.006**	**-0.610 ± 0.008**

Table 5: Effect of including or excluding customer and article embeddings on the performance of SFnet.

Entity embedding		Micro-avg. AUC		Accuracy		Average log-likelihood	
user id	item id	ModCloth	RentTheRunWay	ModCloth	RentTheRunWay	ModCloth	RentTheRunWay
✓	✓	**0.689 ± 0.005**	**0.749 ± 0.004**	**0.690 ± 0.004**	**0.760 ± 0.004**	**-0.758 ± 0.006**	**-0.610 ± 0.008**
×	✓	**0.693 ± 0.009**	**0.751 ± 0.004**	**0.691 ± 0.004**	**0.760 ± 0.001**	**-0.757 ± 0.009**	**-0.607 ± 0.004**
✓	×	0.637 ± 0.004	0.667 ± 0.007	0.686 ± 0.004	0.733 ± 0.007	-0.803 ± 0.006	-0.716 ± 0.023
×	×	0.638 ± 0.007	0.674 ± 0.003	0.683 ± 0.005	0.739 ± 0.002	-0.806 ± 0.009	-0.698 ± 0.006

Laplace smoothing [15] to avoid computational issues with the conditional probability estimation.

Boosted trees: Another well-suited methodology to compare against is gradient boosted trees. High feature cardinality also poses a problem for tree based approaches as it requires the training algorithm to evaluate the best of all the possible partitions of n feature values into k classes, which is equal to the Stirling number of second kind $\genfrac{}{}{0pt}{}{n}{k}$ [6]. We therefore encode fashion experts and shoe attributes using smoothed target encoding [16] to reduce the complexity of the task.

Results: Table 6 shows the results obtained on test data. All three approaches are comparable in terms of accuracy; however, the numbers hover around the a priori probability (73.7% for length and 87.1% for width) of the dominant ‘true to size’ class. We take the results as an indication of expert feedback being unbiased and therefore independent of the considered article attributes. In terms of other metrics, while SFNET takes a clear lead w.r.t. the average AUC, the relatively low likelihood values of SFNET despite being more accurate in comparison to Naive Bayes suggests that the output distributions of SFNET may tend to be more peaky in nature. This leads to a relatively high loss in likelihood when the method predicts the wrong outcome with a high probability.

4.4 Experiments on Purchase Data

In this section, we present results on modeling customer size preferences given their purchase history. Our goal here is to predict the size of articles which customers order and keep. Note that a "customer" in this context refers to a customer account which is potentially used by multiple individuals. This is a realistic scenario for most e-commerce retail platforms and for personalized recommendation, it demonstrates the need for modeling multiple personas behind one identity. We will analyze SFNET's performance on multi-user accounts in Section 4.4.2.

For these experiments, we use our proprietary dataset of customer purchases spanning a period of roughly 5 years. The purchased articles in the data belong to the sub-categories of shoes, textile and sportswear. We only consider customer accounts with at least 5 purchases in the history. The dataset contains roughly 20 million purchases involving around $389K$ customers and $872K$ articles. There are more than $1K$ distinct sizes in the data. Multidimensional sizes such as jeans size 30×32 and 30×33 are taken to be independent of each other. Due to overlapping size systems, a distinct size can be used in multiple clothing sub-categories.

Apart from an anonymous customer identifier, our data does not contain any other customer information. We therefore do not consider cold-start customers in this experiment[2]. For articles we use unique identifiers together with categorical attributes such as brand, main material, country of origin, season and 5 taxonomical attributes which including gender (female, male or unisex), define a non-overlapping hierarchy of clothing items.

Backtesting: To simulate a realistic scenario, we perform our experiments in a backtesting setup. To that end, we split the data chronologically into train, validation and test sets. This implies that our training instances come from the past, while validation and test splits contain more recent purchases with test split containing the latest ones. In backtesting, aside from encountering cold-start customers, we may also encounter new articles in the test for which have not learned any dedicated input embeddings during training; nonetheless the default article embedding (as described in Section 4.2) together with shared attributes such as brand, material, etc. allow us to evaluate new articles in the test (and validation) data split.

With 80% train, 10% validation[3] and 10% test, we keep data split ratios the same as before. During test, we truncate and renormalize the output distributions of SFNET and compared methods to the available sizes of test articles. Moreover, since we allow customers to order more than one sizes, we further report top-2 and top-3 accuracies with the other performance metrics.

[2] In the absence of additional features as in Table 3, if (akin to Section 4.2) we learn a default customer embedding for cold-start customers, we can only expect to approximate population-level marginal distributions over kept sizes in article sub-categories, which will be rather non-informative for personalization.

[3] Since the methods we compare with in this section do not require extensive hyperparameter tuning, we merge the validation split into the training data for those methods.

Bayesian Model: We benchmark our approach against a recently introduced Bayesian method for size recommendation [7]. The approach is based on a hierarchical Bayesian model exploiting the customer purchase history to learn the usual size of multiple users of a single account. Originally, the method was proposed to model both returns and keeps in a customer history, but in our setting where we are only interested in modeling size distribution of kept articles in customer accounts. In this case, the model proposed by [7] reduces to an infinite Gaussian mixture model with an associated truncated Dirichlet process of level four (we refer to [7] for more details).

We train an independent instance of the Bayesian model for articles of all genders (i.e. female, male and unisex) within each of the main clothing categories in data – including shoes and upper and lower body garments. Moreover, since the approach is meant to be for continuous size systems, we employ expert knowledge to convert alpha-numeric sizes (e.g., Small, Medium, Large) into a continuous size range. To disambiguate overlapping numerical size systems, we further use a semi-supervised Gaussian Expectation Maximization algorithm [2] to cluster articles based on the characteristics of their size systems (e.g., minimum, maximum and median sizes, step between sizes, etc.). Once clustered, the size that represents a cluster is defined by a domain expert.

Baseline: We also estimate a population-level marginal distribution of kept sizes, which we obtain by training the Bayesian model for each clothing category and gender across all customers.

Results: As shown in Table 7, SFNET outperforms both Bayesian and baseline approaches on all the metrics. We further observe a narrowing gap between SFNET and the Bayesian approach w.r.t. top-k accuracies. This is due to the fact that for a given article, there are usually a handful of sizes to choose from, hence increasing k significantly boosts the chances of hitting the right size for both the algorithms.

4.4.1 *Dealing with Category Cold-Starts:*

An appealing use-case for size recommendation in e-commerce fashion retail is that of category cold-start where an existing customer with purchase history in other categories orders an article from a new category. Note that for category cold-starts, the Bayesian approach defaults to the baseline approach, which is a category and gender-conditioned marginal distribution of purchased sizes.

Results: While the baseline approach recommends among available sizes, the most purchased size of a category cold-start article, we expect SFNET to be better than that. Indeed tn table 8, we find SFNET's performance on cold-start recommendation in three different categories significantly better than the baseline default mode of the Bayesian approach.

4.4.2 *Modeling Multiple Users Behind One Identity:*

In our last experiment we asses SFNET's capacity to deal with multiple users behind one account. We use gender profiles (i.e. female, male or unisex) of purchased articles to assume customer accounts to

Table 6: Comparison on expert feedback prediction task.

	Micro-avg. AUC		Accuracy		Average log-likelihood	
Method/Feedback	Length	Width	Length	Width	Length	Width
Naive Bayes	0.681 ± 0.003	0.716 ± 0.006	0.737 ± 0.003	0.875 ± 0.005	-0.656 ± 0.005	-0.395 ± 0.004
Boosted Trees	0.708 ± 0.003	0.715 ± 0.005	0.748 ± 0.009	0.872 ± 0.003	-0.746 ± 0.028	-0.464 ± 0.009
SFnEt	0.753 ± 0.004	0.773 ± 0.006	0.742 ± 0.005	0.876 ± 0.003	-0.698 ± 0.011	-0.409 ± 0.007

Table 7: Comparison on test data containing articles in various clothing categories and overlappting size systems.

Method	Micro-avg. AUC	Accuracy top-1	top-2	top-3	Average log-likelihood
Baseline	0.690	0.24	0.45	0.64	−1.82
Bayesian	0.834	0.503	0.770	0.886	−1.37
SFnEt	0.861	0.555	0.795	0.898	−1.19

Table 8: Category cold-start performance in three different categories.

	Micro-avg. AUC	Accuracy top-1	top-2	top-3	Average log-likelihood
		Men's Shirts			
Baseline	0.63	0.21	0.44	0.61	−1.78
SFnEt	0.723	0.403	0.698	0.810	−1.63
		Jeans			
Baseline	0.68	0.20	0.38	0.53	−2.10
SFnEt	0.775	0.295	0.509	0.646	−2.22
		Shoes			
Baseline	0.71	0.24	0.45	0.62	−1.88
SFnEt	0.745	0.293	0.516	0.679	−1.99

be single or multi-user. Based on the gender profiles, we first filter the data to contain only those accounts with both female and male articles in the test split. We then perform ablations by partitioning the filtered accounts w.r.t. their gender distribution in the training data, yielding the three rows of Table 9. The first row represents user accounts that either contain female and unisex, or male and unisex articles in their training histories. During test, as indicated by male and female columns of the table, those accounts are tested on the articles of gender that was lacking in their training histories. We term such cases as 'gender cold-starts'. The second row of the table represents the opposite of the first row, where accounts with female and unisex (respectively male and unisex) articles in training data are tested on female (respectively male) articles. The last row represents the accounts which contain all the three genders in their training histories and we test their performance on female vs. male articles.

Results: As the Bayesian approach defaults to the baseline for the gender cold-starts, we see identical numbers for both methods in the first row of Table 9; to our surprise however, SFnEt's performance

Table 9: Top-1 accuracy on multi-user accounts in test. Rows represent different types of customer histories encountered during training.

gender	Baseline male	Baseline female	Bayesian male	Bayesian female	SFnEt male	SFnEt female
cold-start	0.219	0.253	0.219	0.253	0.325	0.300
consistent	0.220	0.253	0.434	0.496	0.496	0.559
mixed	0.218	0.253	0.396	0.503	0.481	0.549

for the gender cold-starts is significantly better than the baseline marginals. We hypothesize that SFnEt makes use of higher-order correlations discovered from multi-user accounts to achieve the results. In the second row of the table, we see SFnEt is most accurate with user accounts that are consistently one gender (plus unisex) during training and test. For multi-user accounts in the third row, we observe a reduction in SFnEt's performance, yet the accuracy is significantly higher than the Bayesian (and baseline) approach. The results are indicative of SFnEt's capacity for modeling multiple users, although further analysis is warranted to assess SFnEt's ability to disambiguate multiple intents.

5 CONCLUSION

In this work we proposed SFnEt, a deep learning based methodology which combines collaborative and content-based modeling techniques to learn input and latent representations of customers and articles for size and fit prediction. The method is highly scalable and works end-to-end without requiring a priori knowledge about its prediction targets underlying ordinal structure. As demonstrated by competitive empirical performance in a variety of experiments on multiple datasets, our SFnEt architecture offers both the flexibility and the capacity for capturing higher-order abstractions of size and fit relevant information from arbitrary customer and article features. Future extensions of this work can include multi-view objectives [5] (such as predicting both categorical and ordinal targets) or time-dependent modeling of customer behavior [4] with respect to size and fit.

ACKNOWLEDGMENTS

We acknowledge and appreciate constructive feedback from our reviewers and area chair. We thank Alan Akbik and Calvin Seward for their valuable feedback in the preparation of this manuscript. We would also like to thank Julia Lasserre for helpful discussions on the design of experiments on customer purchase data.

REFERENCES

[1] G Mohammed Abdulla and Sumit Borar. 2017. Size recommendation system for fashion e-commerce. In *Workshop on Machine Learning Meets Fashion, KDD*.

[2] Sugato Basu, Arindam Banerjee, and Raymond Mooney. 2002. Semi-supervised clustering by seeding. In *In Proceedings of 19th International Conference on Machine Learning*.

[3] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature Verification using a "Siamese" time delay neural network. In *Advances in Neural Information Processing Systems 6*, J. D. Cowan, G. Tesauro, and J. Alspector (Eds.). 737–744.

[4] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential user-based recurrent neural network recommendations. In *Proceedings of the 11th ACM Conference on Recommender Systems (RecSys '17)*.

[5] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 278–288.

[6] Ronald L Graham, Donald E Knuth, Oren Patashnik, and Stanley Liu. 1989. Concrete mathematics: a foundation for computer science. *Computers in Physics* 3, 5 (1989), 106–107.

[7] Romain Guigourès, Yuen King Ho, Evgenii Koriagin, Abdul-Saboor Sheikh, Urs Bergmann, and Reza Shirvany. 2018. A hierarchical bayesian model for size recommendation in fashion. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 392–396.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[9] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 173–182.

[10] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough Data. *ACM SIGKDD Explorations Newsletter* 3, 1, 27–32.

[11] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

[12] Christopher C Johnson. 2014. Logistic matrix factorization for implicit feedback data. *Advances in Neural Information Processing Systems* 27 (2014).

[13] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[14] Yehuda Koren and Robert M. Bell. 2015. Advances in Collaborative Filtering. In *Recommender Systems Handbook*. 77–118.

[15] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge University Press.

[16] Daniele Micci-Barreca. 2001. A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *ACM SIGKDD Explorations Newsletter* 3, 1 (2001), 27–32.

[17] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2018. Efficient estimation of word representations in vector space. *Workshop in International Conference on Learning Representations (ICLR)* (2018).

[18] Rishabh Misra, Mengting Wan, and Julian McAuley. 2018. Decomposing fit semantics for product size recommendation in metric spaces. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 422–426.

[19] Fanke Peng and Al-Sayegh. Mouhannad. 2014. Personalised Size Recommendation for Online Fashion. In *6th International conference on mass customization and personalization in Central Europe*. 1–6.

[20] Gina Pisut and Lenda Jo Connell. 2007. Fit preferences of female consumers in the USA. *Journal of Fashion Marketing and Management: An International Journal* 11, 3 (2007), 366–379.

[21] Vivek Sembium, Rajeev Rastogi, Atul Saroop, and Srujana Merugu. 2017. Recommending product sizes to customers. In *Proceedings of the 11th ACM Conference on Recommender Systems*. ACM, 243–250.

[22] Vivek Sembium, Rajeev Rastogi, Lavanya Tekumalla, and Atul Saroop. 2018. Bayesian models for product size recommendations. In *Proceedings of the 25th International Conference on World Wide Web*. 679–687.

[23] Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Collaborative filtering beyond the user-item matrix: A survey of the state-of-the-art and future challenges. *ACM Comput. Surv.* (2014).

[24] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*. 926–934.

[25] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

Domain-E-Commerce

Title:
SmartSize AI: Machine Learning-Based Clothing Size Recommender

Group-6:

2420030171-R.L.S.Giridhar
2420030113-S.R.Harshini
2420090120-Satya Dev