From Pixels to Purchases

A Deep Learning Approach to Heterogenous Consumer Aesthetics in Retail Fashion

Pranjal Rawat

May 17, 2024

Georgetown University

Table of Contents

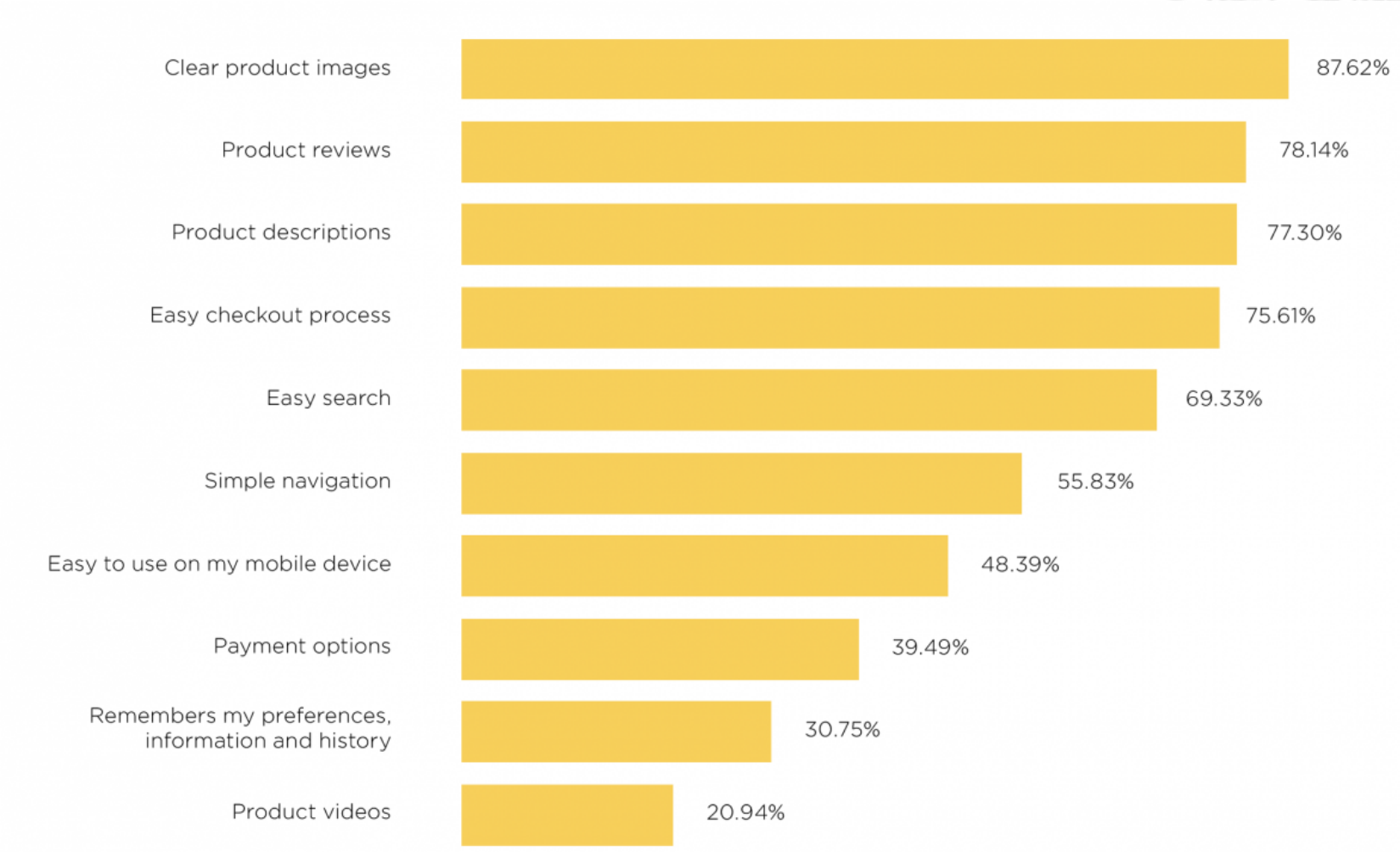
Introduction

Research Questions

- In some markets, **product aesthetics drives purchases**.
  - Fashion and Apparel, Designer Bags, Interior Design, Home Decor, Watches, Jewellery, Art, Luxury Cars, Shoes, Dating and Marraige.
- **Research Questions**:
  - How do we extract product aesthetics from an image?
  - How can we capture a diversity of aesthetic taste?
  - Will this change estimates of price sensitivity?
  - Can AI design new aesthestics?

Motivation

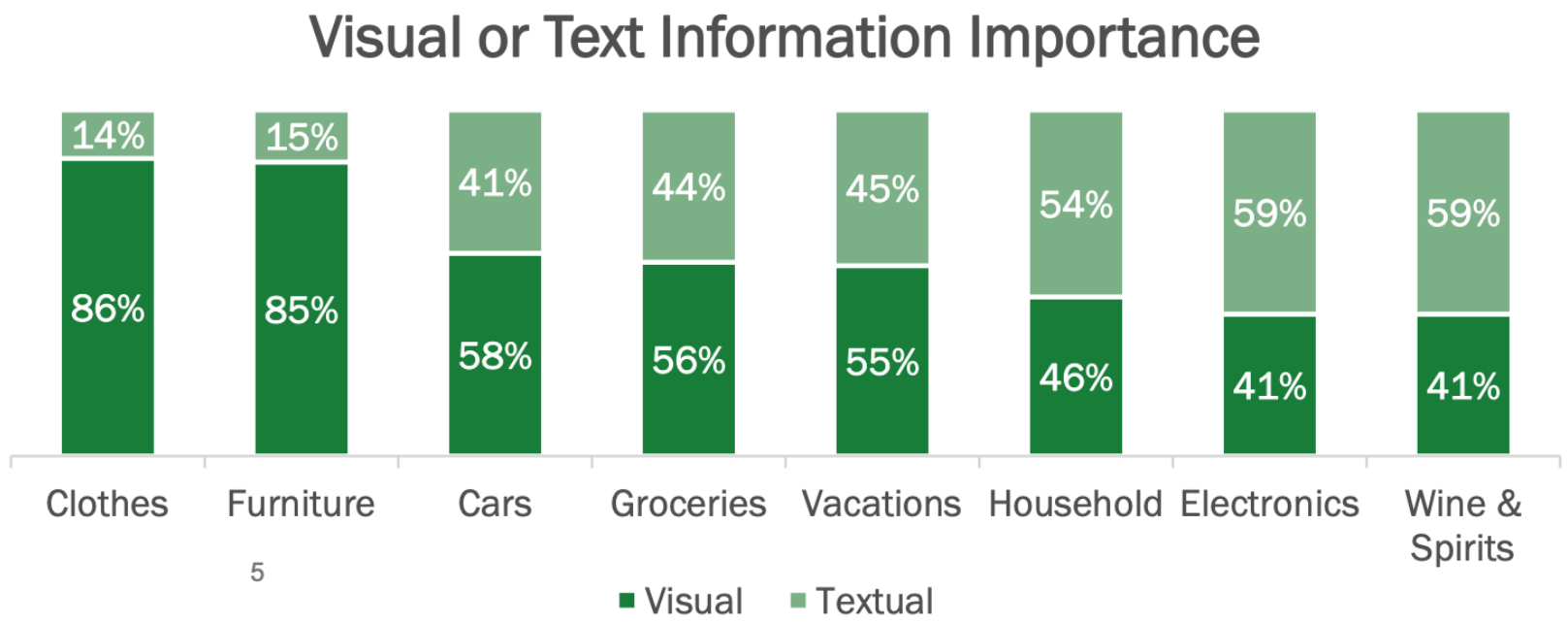
Survey[1]: What makes for a great shopping experience?



[1]Namogoo 2018 Survey: A national sample of 1,372 U.S. online shoppers.

Motivation

2018 Survey[2]: What is important when searching for a product online?



[2]Intent Lab, Northwestern U: A national sample of 1000 U.S. residents.

Motivation

Why should we care about images?

- Consumers care a lot about them.
  - 88% say high-quality product imagery is important (Nfinite 2022).
- Help us understand how consumers respond to prices:
  - By controlling for confounders.
  - By building instruments.
- **They are also interesting in their own right:**
  - For understanding the aesthetic taste of a demographic.
  - For developing and testing new designs.

Literature

Only recently[3], some papers use images to estimate **price sensitivity**:

- Quah & Williams (2021): Logit-demand for shoes using images and debiased machine learning.
- Giovanni et al., (2021): Similarity between Amazon product webpages used to model error covariances in a nested logit.
- Han et al., (2021): Demand for fonts using dense embeddings from image autoencoders.

Other papers also use high dimensional information:

- Magnolfi el al., (2022): Demand estimation for cereal with tSNE embeddings of product characteristics from consumer surveys.
- Zhang (2024): Differentiates consumer price sensitivities by browsing activity in a market for smartphones.

[3]Image processing only took off in 2014, high dim econometrics after 2016.

Contribution

Two main limitations:

- No individual consumer heterogeneity.
- Focus has been mainly on price sensitivity.

The **contribution** of this project:

- **Modelling the variation in consumer aesthetic taste.**
- Estimating individual level price sensitivities.
- Showing how deep learning can be useful to economic modelling.

Methodology

The methodology for this project is derived from:

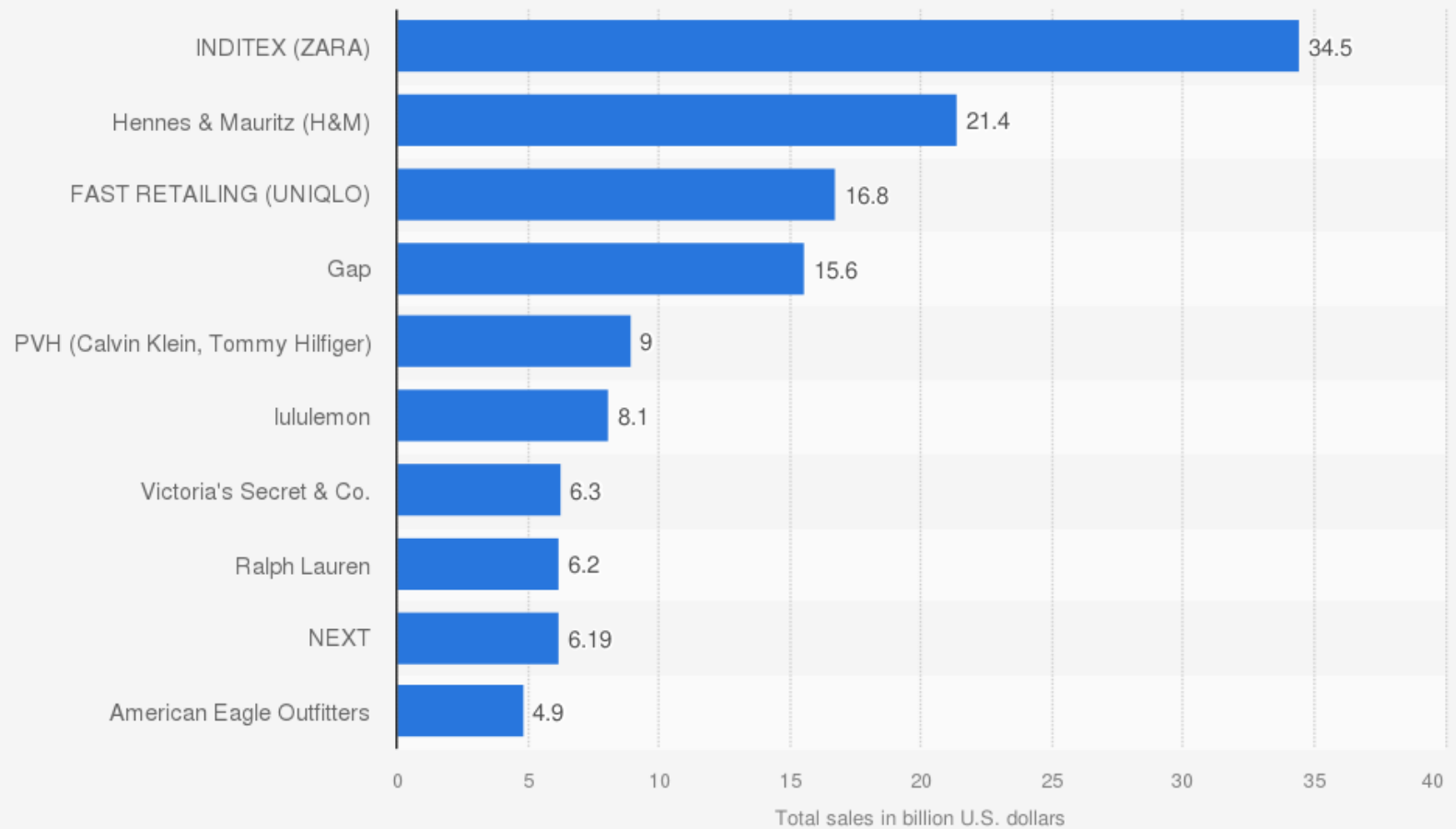
- **Farrell et al. (2018, 2021): Neural networks can be used to model parameters and test hypotheses.**
- Chernozhukov et al. (2018, 2022): Sample-splitting can eliminate bias due to the use of regularized machine learning methods.
- Berry and Haile (2021), Ken Train (2009), Russell (2015): Surveys on demand estimation, discrete choice, and brand choice.
- Ludwig & Mullainathan (2023): Hypothesis generation through machine learning methods.

Data

Data Source

- Hennes & Mauritz (H&M) is a Swedish **fast fashion** brand:
  - 75 geographical markets
  - 4800 stores
  - 107,000 employees
  - 24.8 Billion USD annual revenue
- Motto: "to make fashion accessible and enjoyable for all"
  - Runway to store in 2 weeks.
- **Trendy and affordable clothes of lower quality.**

Industry Snapshot



Data Summary

- Sample of **two years of shopping transactions** in an undisclosed European market (possibly the Netherlands):
  - 1,371,980 Customers
    - Median age 32, 66% Active
  - 108,775,015 Granular Products[4].
    - Ladieswear (25%), Divided (14%), Menswear (12%)...
    - Trousers (11%), Dress (10%), Sweater (9%)...
    - 45875 products, **43404 text descriptions**
  - 31,788,324 Transactions
    - 70% offline, 30% online
- **Women's Dresses**: Popular category that has a large number of products with a wide range of distinct SKUs (2300).

[4] Stock Keeping Units (SKUs)

Dresses: Best Sellers

ID: 745475001
Share: 0.38%

Short dress in woven fabric with a small stand-up collar and V-neck opening at the top. Dropped shoulders, short sleeves and a rounded hem. Slightly longer at the back.

ID: 817353008
Share: 0.37%

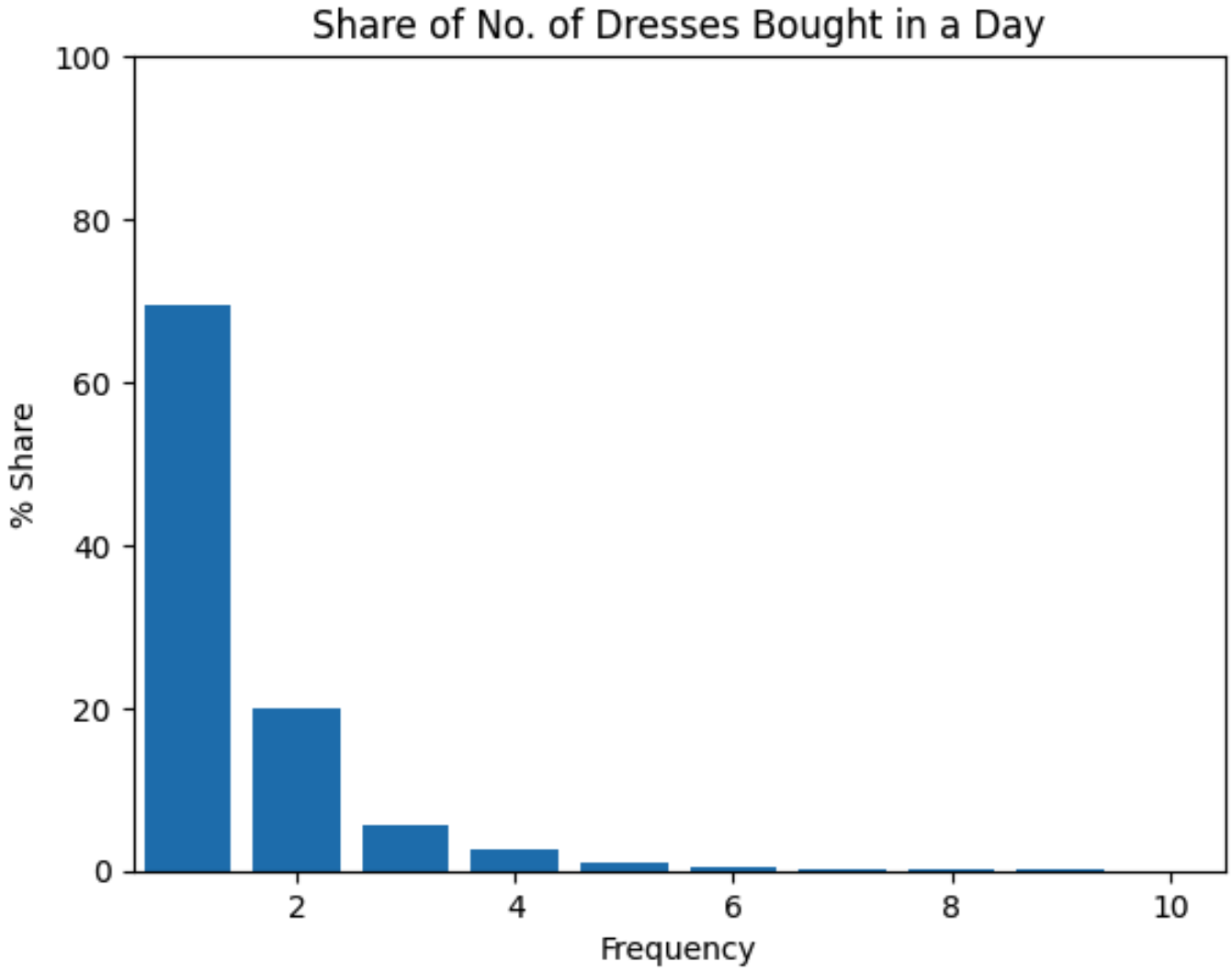
Short dress in woven fabric with a V-neck, buttons down the front, short puff sleeves with smocked trims and a seam at the waist. Unlined.

ID: 745475002
Share: 0.35%

Short dress in woven fabric with a small stand-up collar and V-neck opening at the top. Dropped shoulders, short sleeves and a rounded hem. Slightly longer at the back.

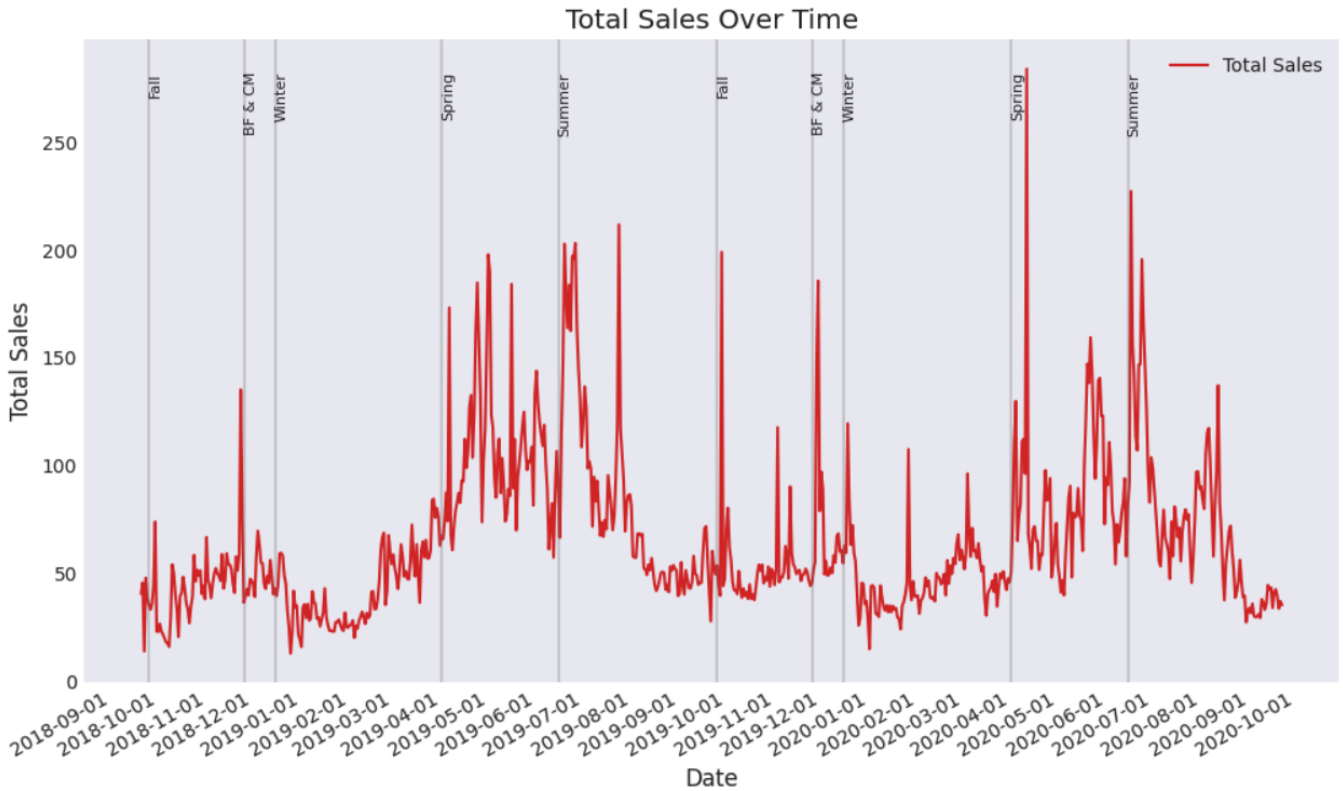
Dresses: Purchases Per Day

Majority of shoppers buy only one dress in a session.



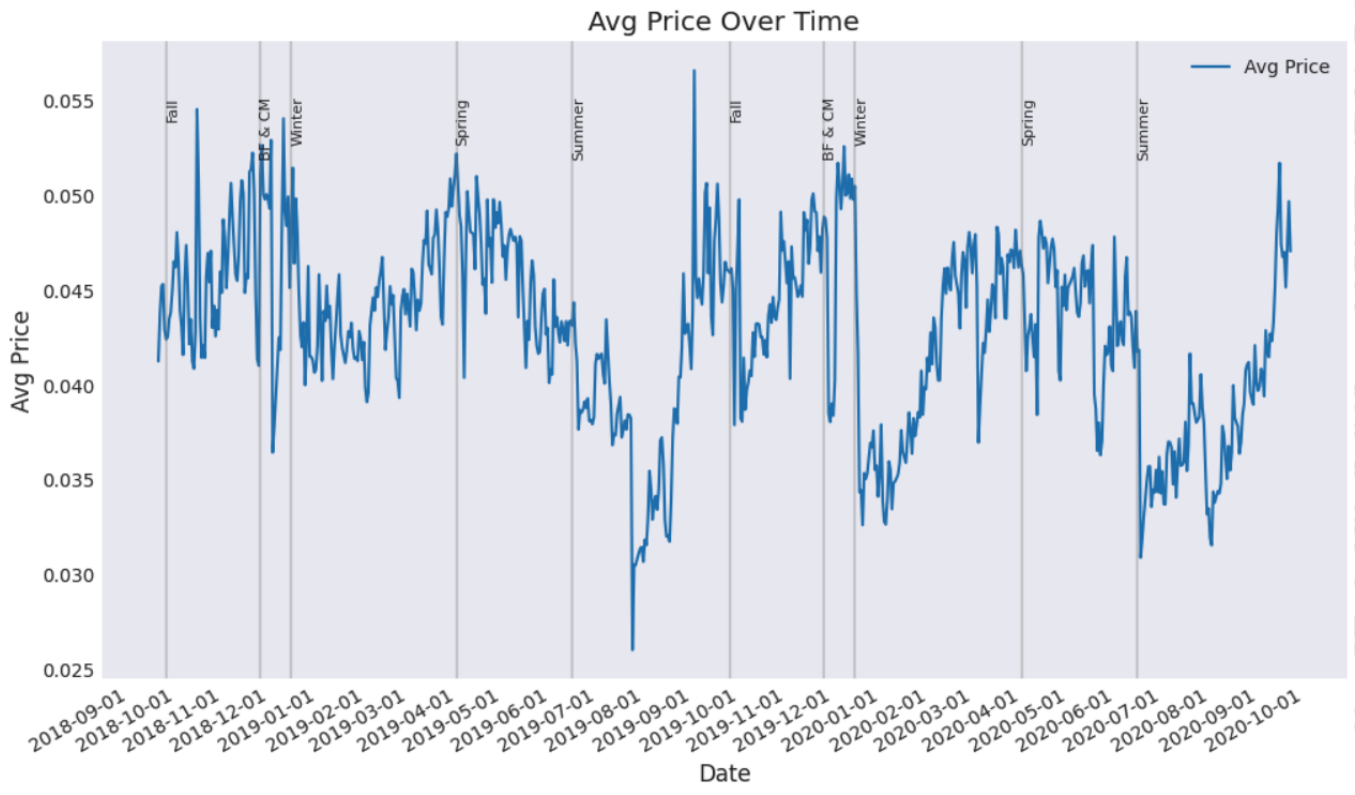
Dresses: Sales over Time

- Strong seasonal variations
- Skewed: few products gather the most sales.



Dresses: Avg Price over Time

- Sharp price discounting.
- Bell shaped prices.



Main Takeaways

The data suggests that:

- Discrete choice framework is appropriate.
- Images matter, at least for online shopping.
- Seasonal variation matters.

In the next section, I show how to work with images and text.

Representing Images and Text

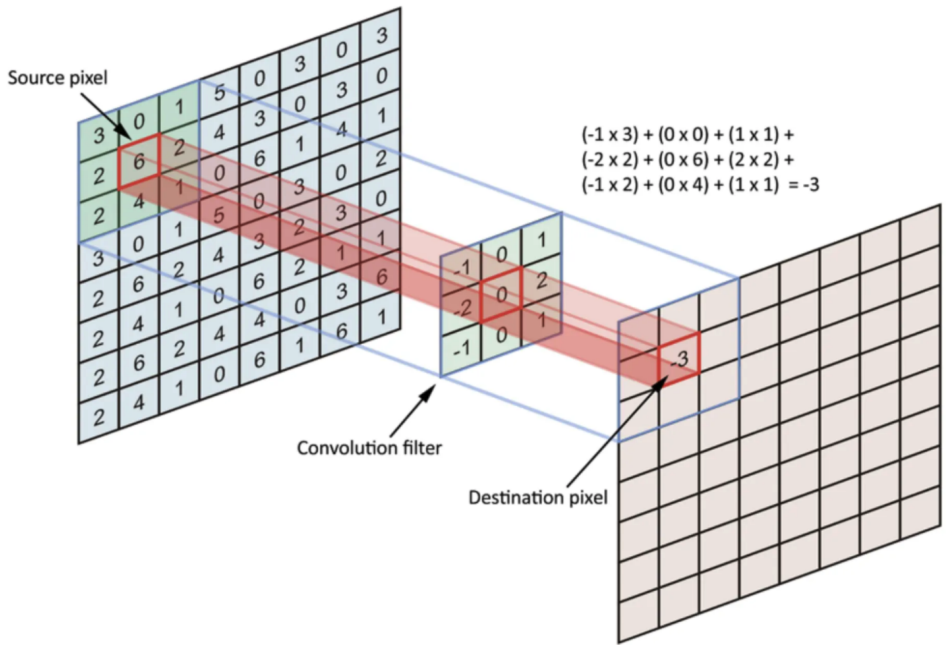
Embeddings

We can represent images and text by dense, high dimensional vectors called **embeddings**:

- We can estimate these embeddings ourselves.
- However, better to use **pre-trained embedding models**:
  - More parameters
  - More data
  - More training time
- I validate the usefulness of embedding representation:
  - They can help us predict prices and sales.
  - They cluster the product space intuitively.

Embedding Images: Convolution Operator

A 3x128x128 **image** M_j **has 49152 pixels for product** j**.** A regression like $s_j = \beta' M_j$ does not work because interactions between pixels matter[5].

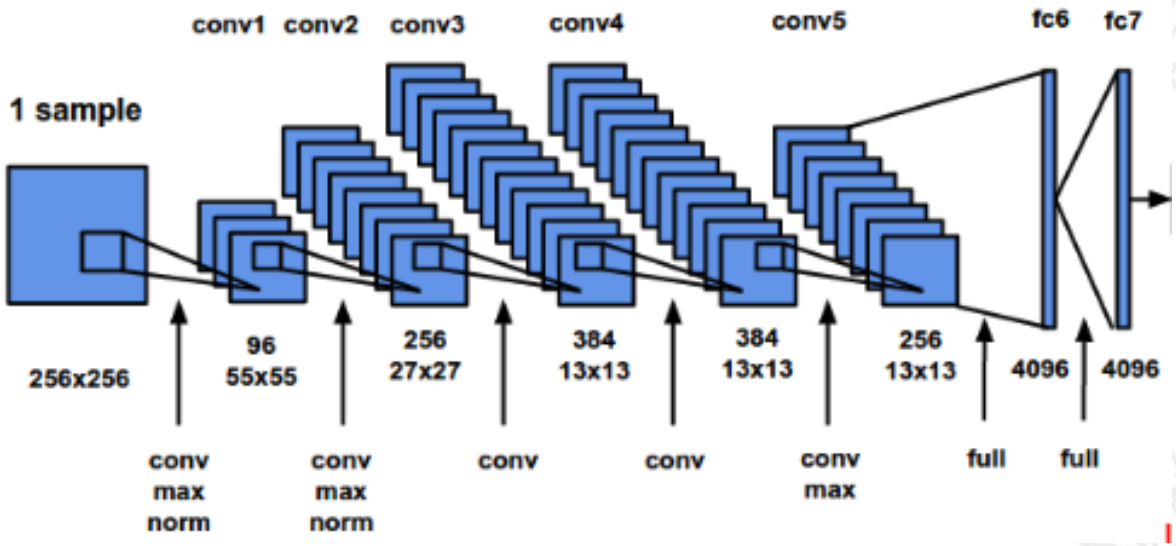


The convolution operator allows **parameter sharing** and captures **visual concepts** from **local spatial interactions** between pixels.

[5]With full interactions we get 1,207,984,128 parameters.

Embedding Images: Conv-Net Architecture

Different **filters** extract different aspects of the image (e.g., shape, texture, pattern, etc.) and refine them into higher-level concepts.



The final output is a dense high-dimensional embedding $e(M_j) \in \mathbb{R}^K$ that represents the attributes from the image[6].

[6] Can be refined even further.

Embedding Images: Image Regression

Estimate CNN on raw image pixels M_j to predict sales share s_j:

$$s_j = g^{\text{CNN}}(M_j) + \epsilon_j$$

and get an out-of-sample R2 of approximately 0.35.

Manually changing images, changes the prediction.

Contrastive Language-Image Pre-training (CLIP)

- I use **Fashion CLIP**[7], a model that matches image-text pairs to other image-text pairs that fall under the same category.
- It encodes both images and texts into **embedding** vectors in a shared latent space.
- For product j, having image M_j and text T_j, I get,

$$e_M(M_j) \in \mathbb{R}^{512}$$

$$e_T(T_j) \in \mathbb{R}^{512}$$

[7] Chia et al., 2022, Nature Scientific Reports: Trained on 700,000 pairs from Farfetch, one of the largest fashion luxury retailers in the world.

Validation 1: Predictive Performance of Image Embeddings

Image embeddings can predict sales and prices.

$$y_j = f(e_M(M_j)) + \epsilon_j$$

Model f	R2 Scores for Log Shares		R2 Scores for Log Prices	
	Train	Test	Train	Test
OLS	0.69	0.27	0.69	0.33
Ridge	0.68	0.31	0.69	0.37
Random Forests	0.87	0.43	0.86	0.41
Deep Nets	0.87	0.44	0.62	0.46
Boosting Machine	0.88	0.45	0.89	0.48
Ensemble	0.89	0.47	0.79	0.49

Validation 2: Predictive Performance of Text Embeddings

Text embeddings can predict sales and prices.

$$y_j = f(e_T(T_j)) + \epsilon_j$$

Model f	R2 Scores for Log Shares		R2 Scores for Log Prices	
	Train	Test	Train	Test
OLS	0.59	0.0	0.76	0.39
Ridge	0.57	0.08	0.74	0.47
Random Forests	0.77	0.34	0.85	0.63
Deep Nets	0.72	0.20	0.68	0.56
Boosting Machine	0.78	0.34	0.87	0.67
Ensemble	0.77	0.33	0.79	0.64

Validation 1: Predictive Performance of Combined Embeddings

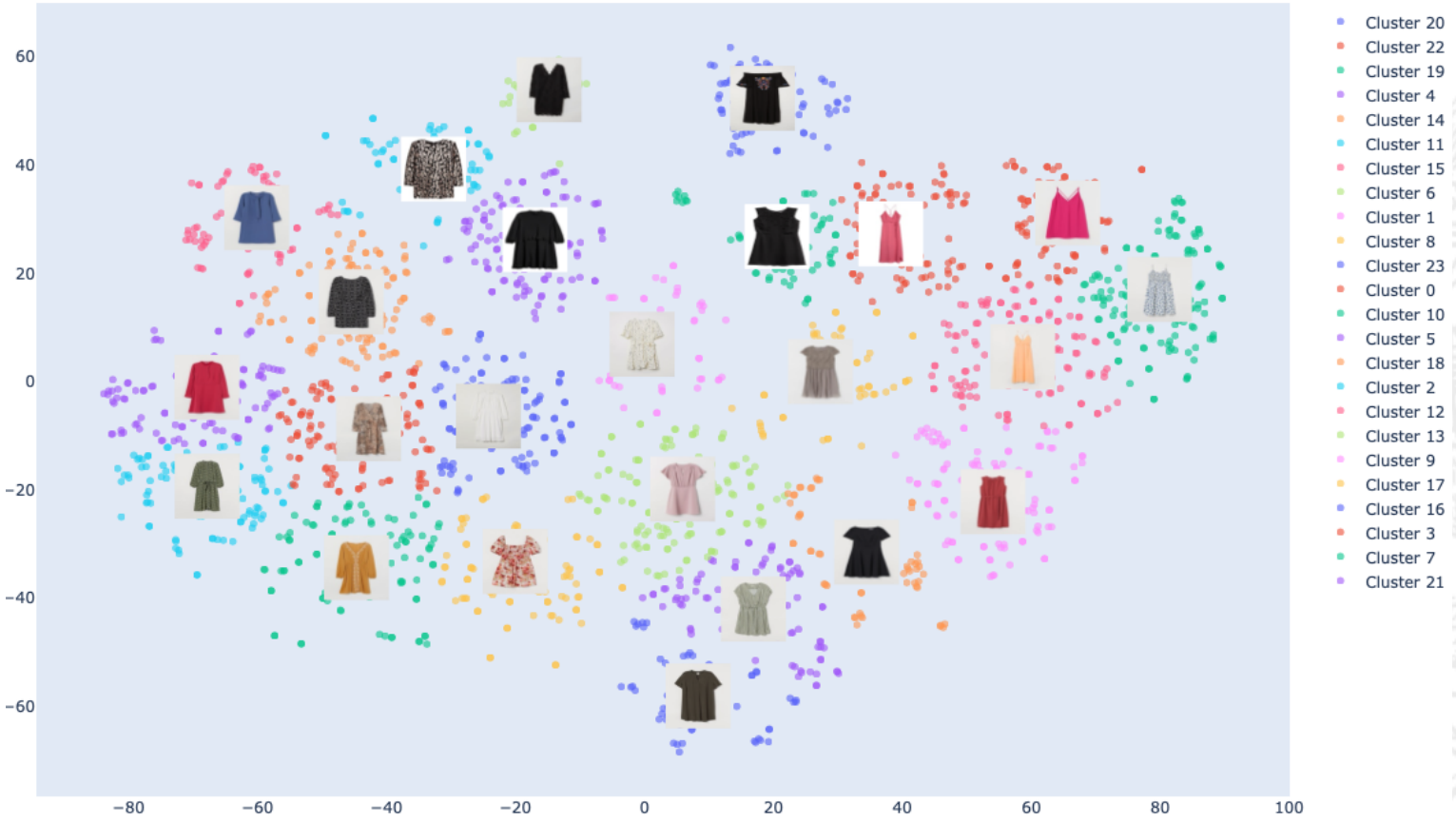
Combining information does even better.

$$y_j = f(e_M(M_j), e_T(T_j)) + \epsilon_j$$

Model f	R2 Scores for Log Shares		R2 Scores for Log Prices	
	Train	Test	Train	Test
OLS	0.85	0.04	0.88	0.26
Ridge	0.83	0.26	0.87	0.42
Random Forests	0.88	0.47	0.90	0.59
Deep Nets	0.94	0.38	0.77	0.49
Boosting Machine	0.90	0.49	0.92	0.63
Ensemble	0.93	0.51	0.92	0.67

Validation 4: Visualizing the Product Space

I use tSNE[8] to reduce to two dimensions: $e_{tSNE}(e_M(M_j), e_T(T_j)) \in \mathbb{R}^2$ and then detect 24 clusters[9].



[8] t-distributed Stochastic Neighbour Embedding

[9] Each dot is a dress, and each image is the most representative dress in that cluster.

Validation 5: Within Cluster Variation

Word Cloud for Cluster 1 (122 articles)

Word Cloud for Cluster 13 (57 articles)

Word Cloud for Cluster 16 (81 articles)

Validation 5: Within Cluster Variation

Word Cloud for Cluster 23 (119 articles)

Word Cloud for Cluster 10 (147 articles)

Word Cloud for Cluster 8 (99 articles)

Main Takeaways

This section shows that:

- Embeddings represent images and text well:
  - They are predictive of sales and prices
  - They cluster products intuitively.
- OLS cannot extract insights from embeddings.
- Deep networks do better because they allow complex interactions.

In the next section, I incorporate embeddings into choice models.

Consumer Choice Models

Notation

Key terms:

- y_i Product purchases by consumer i (Categorical)
- p_j Avg Price of product j
- $X_j = [e_M(M_j), e_T(T_j)]$ Text and Image Embeddings of product j
- D_i Demographics of consumer i
- α, α^{DNN} Price Parameter
- g^{DNN} Deep Neural Network that measures aesthetic taste

Consumer Choice Model

For i-th consumer the utility given by the j-th product is given by,

$$u_{ij} = h_1(p_j, D_i; \theta_1) + h_2(X_j, D_i; \theta_2) + \epsilon_{ij}$$

and discrete choice is,

$$y_i = \underset{j \in \{1,2,\ldots,J\}}{\text{argmax}} \; h_1(p_j, D_i; \theta_1) + h_2(X_j, D_i; \theta_2) + \epsilon_{ij}$$

If ϵ_{ij} is EV(1) and IID,

$$\mathbb{P}(y_i = j) = \frac{e^{h_1(p_j, D_i; \theta_1) + h_2(X_j, D_i; \theta_2)}}{\sum_k e^{h_1(p_k, D_i; \theta_1) + h_2(X_k, D_i; \theta_2)}}$$

Estimation

This leads to the loglikelihood:

$$\ell(\theta) = \sum_i \sum_j 1(y_i = j) \log \left(\frac{e^{h_1(p_j, D_i; \theta_1) + h_2(X_j, D_i; \theta_2)}}{\sum_k e^{h_1(p_k, D_i; \theta_1) + h_2(X_k, D_i; \theta_2)}} \right)$$

- Estimation: get $\ell'(\theta)$ (backdrop) + hill climbing
- StdErr: Hessians + Influence functions (Farell et al., 2018, 2021)
- No outside good! Model limited to consumers who do purchase.

Functional Forms

I estimate models from simple to complex:

- $u_{ij} = \alpha \log p_j + \beta' X_j + \epsilon_{ij}$
- $u_{ij} = \alpha \log p_j + g^{DNN}(X_j) + \epsilon_{ij}$
- $u_{ij} = \alpha \log p_j + g^{DNN}(X_j, D_i) + \epsilon_{ij}$
- $u_{ij} = \alpha' d_i \log p_j + g^{DNN}(X_j, D_i) + \epsilon_{ij}$
- $u_{ij} = \alpha^{DNN}(D_i) \log p_j + g^{DNN}(X_j, D_i) + \epsilon_{ij}$

Results I: Controls Matter

Model	α	Std Err	p-value
$u_{ij} = \alpha \log p_j + \beta' X_j + \epsilon_{ij}$	-0.40	0.02	0.00
$u_{ij} = \alpha \log p_j + g^{DNN}(X_j) + \epsilon_{ij}$	-0.26	0.02	0.00
$u_{ij} = \alpha \log p_j + g^{DNN}(X_j, D_i) + \epsilon_{ij}$	-0.37	0.02	0.00

Table 1: Homogenous Logit Results

Results II: Heterogenous Price Sensitivity

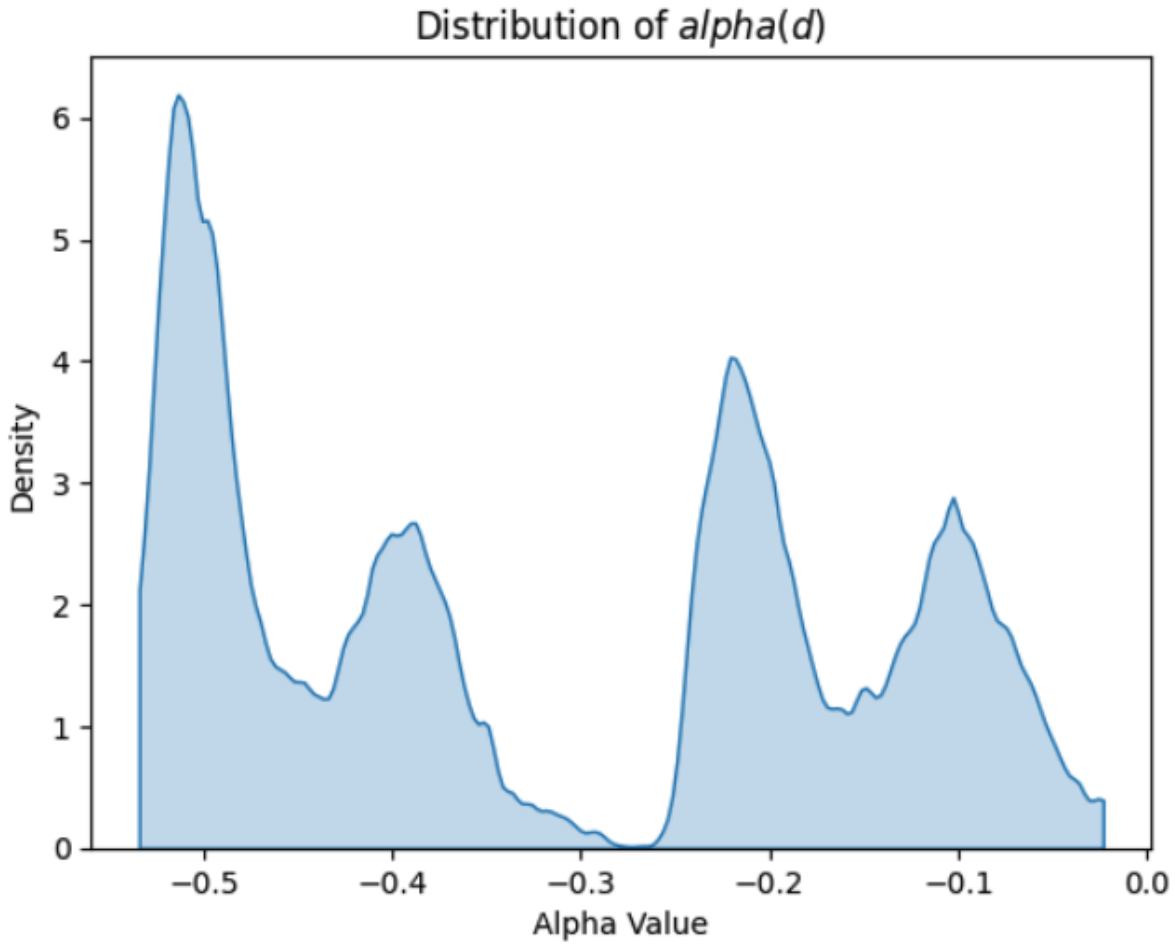
$$u_{ij} = (\alpha' D_i) \log p_j + g^{DNN}(X_j, D_i) + \epsilon_{ij}$$

Standardized Variable	Coefficient
Intercept	-0.4249
Active Communications	-0.1551
Age	-0.0428
Club Member Active	-0.1427
Club Member Left	-0.1374
Club Member Processing	0.0976
Following Newsletter	0.3864
News Freq Monthly	0.0928
News Freq None	0.2663
News Freq Regularly	0.0343

Results III: Heterogenous Price Sensitivity

Neural networks capture a larger heterogeneity in price sensitivity.

$$u_{ij} = \alpha^{DNN}(D_i) \log p_j + g^{DNN}(X_j, D_i) + \epsilon_{ij}$$



Results IV: Heterogenous Consumer Aesthetics

Using the g function, I rank order all products for two groups of people.

Figure 1: Top ranked dresses for 95th quantile by age (55-60)

Figure 2: Top ranked dresses for 5th quantile by age (18-21)

Main Takeaways

Some insights from choice models:

- Price sensitivities can range widely:
  - **A 50% percent discount increases probability of purchase for some by 25% and others by only 5%.**
- **Different demographics have very different aesthetic tastes.**

Key Extensions

Some enrichments:

- Richer demographics.
- Brand loyalty.
- Seasonal variation.
- Unobserved heterogeneity.
- Price endogeneity.

Seasonal Variation and Unobserved Types

Utility for j-th product for i-th customer of type k in period t be,

$$U_{ijtk} = \alpha_k(D_i, S_t) \log p_{jt} + g_k(X_j, D_i, S_t) + \epsilon_{ijtk}$$

p_{jt} Prices
S_t Seasonal Dummies
k Unobserved Type

Seasonal Variation and Unobserved Types

Allowing unobserved types of consumers helps account for missing demographic information[10]:

$$\mathbb{P}(y_i = j) = \sum_k \pi_k \frac{e^{\alpha_k(D_i, S_t) \log p_{jt} + g_k(X_j, D_i, S_t)}}{\sum_{j'} e^{\alpha_k(D_i, S_t) \log p_{j't} + g_k(X_{j'}, D_i, S_t)}}$$

- Additional parameters to estimate: π_k, α_k, g_k
- Choice of k: when the likelihood increases tapers off.

[10] Heckman and Singer 1984

Handling Price Endogeneity via Control Functions[13]

Given instruments Z_{jt}[11] we have,

$$u_{ijt} = \alpha \log p_{jt} + g(X_j, S_t) + \xi_{jt} + \epsilon_{ijt}$$

where ξ_{jt}[12] is the omitted variable. We model ξ_{jt},

$$\log p_{jt} = q(X_j, S_t, Z_{jt}) + v_{jt}$$

$$\xi_{jt} = \gamma(v_{jt})$$

This gives us,

$$\mathbb{P}(y_i = j) = \frac{e^{\alpha \log p_{jt} + g(X_j, S_t) + \xi_{jt} + \epsilon_{ijt}}}{\sum_{j'} e^{\alpha \log p_{j't} + g(X_{j'}, S_t) + \xi_{j't} + \epsilon_{ij't}}}$$

[11] Lagged prices, lagged shares, BLP, differentiation, costs, Wagelfold, Hausman.
[12] Promotional activity (feature and display), word-of-mouth effects
[13] Villas-Boras and Winer 1999, Petrin and Train 2010

Simulations

Discounting Policies

- Optimal Discounts:
  - Given budget B, set of customers arriving T_i, assuming costs c_j, how do we select discounts w_{ij}:

$$\mathbf{w} = \text{argmax}_{w_{ij}} E[\sum_{i \in T_i} \sum_{j} 1(y_i = j)(p_j - w_{ij} - c_j)$$

$$\sum_{i} \sum_{j} w_{ij} = B$$

  - If nonlinear logit is true, what is the profit lost in using simple logit?
  - What is the welfare cost of discounts?
  - Can price discrimination actually help some section of consumers?

Design Policies

- Optimal Number of Designs:
  - How many designs does H&M really need to stay within 95% of total profits?

$$\pi = \max E[\sum_{i \in T_i} \sum_{j} 1(y_i = j)(p_j - c_j)]$$

  - Does H&M have too many designs?
  - What if it just sold a core group of products year-round?

New Design

- Can AI design dresses?
  - Use optimization techniques to learn in latent space X_j, to maximize probability of purchase for a given consumer:

$$X_j^* = \text{argmax}_{X_j} \mathbb{P}(y_i = j)$$

  - What would be the total profit gain?
  - What would AI-generated dresses reduce the diversity of options?

Next steps

- Enrich the model:
  - Better demographics.
  - Brand loyalty.
  - Seasonal variation.
  - Unobserved Heterogeneity.
  - Price endogeneity.
- Conduct simulations:
  - Discounting.
  - Number of designs.
  - Using AI to develop designs.

Appendix

Extended Literature

- **Demand/Choice Models**: Train (2009), Gandhi and Houde (2019), Conlon and Gortmaker (2020), Berry and Haile (2021), Petrin and Train (2010)
- **Deep Learning Architechtures**: Younesi et al., (2024), Vaswani et al., (2017), Kingma et al., (2015), LeCun et al., (2015), He et al., (2016), Sun et al., (2020).
- **Machine Learning in Econometrics**: Chernozhukov et al (2018, 2022), Farrell et al (2018, 2021), Ludwig & Mullainathan (2023).
- **Retail Fashion Industry**: McCormick et al (2014), Wen et al., (2019), Bhardwaj and Fairhurst (2009)
- **Applied Work**: Quah & Williams (2021), Han et al., (2021), Giovanni et al., (2021), Zhang et al (2022), He et al (2023), Janssens et al (2021), Zhang (2024)

Retail Fashion: Industry

- Monopolistic competition - many small to medium firms, differentiated products and few entry barriers.
- Fashion industry is characterized by:
  - large number of incomparable products.
  - short product lifespans.
  - large number of seasonal products.
  - collections with non-replinishable inventory.
  - widespread promotions and discounts.
  - some amount of brand loyalty but not too much.

Retail Fashion: Industry Evolution

Since the 1980s there have been critical changes[14]

- **Product Design**: End of mass-production (Levi's 501, White Tees) and move towards trendy and stylish apparel. This led to increase in mark-downs due to failure to sell inventory during season.
- **Fashion Seasons**: Shrinking of time between runway to delivery. Incoporation of 3-6 midseasons to the traditional fashion calender (Spring/Summer + Autumn/Winter).
- **Supply Chain**: Cost reduction from outsourcing. Quick Response and Just-in-Time strategies to combat longer lead times, complex supply chains.
- **Consumers**: More aware of fashion trends, prices, and options. Increasingly interested in cheaper but fashionable clothes.

[14] Bhardwaj and Fairhurst 2009

Retail Fashion: Pricing

- Common methods to determine pricing:
  - Keystone markup - if costs are known then a simple markup is applied (2x, 4x).
  - Backwards pricing - find what consumers are willing to pay and then work back to determine cost and materials.
  - Promotion / Discounting - adjust prices to deplete inventory as season comes to an end.
- With better data, faster development to delivery times, smarter inventory management, there is more dynamic pricing.
- Fast fashion (Zara, HM), operate at scale, and have made average prices lower, but luxury brands (Dior, Chanel) have increased prices.

Dresses: Worst Sellers

ID: 879646003
Share: 0.00%

Short dress in a jacquard weave with a slight sheen. Double-layered stand-up collar and an opening at the back with concealed hook-and-eye fasteners at the back of the neck. Short, raglan puff sleeves with narrow elastication at the hems, and a seam at the waist with a narrow drawstring that ties at the back. Lined.

ID: 524855001
Share: 0.00%

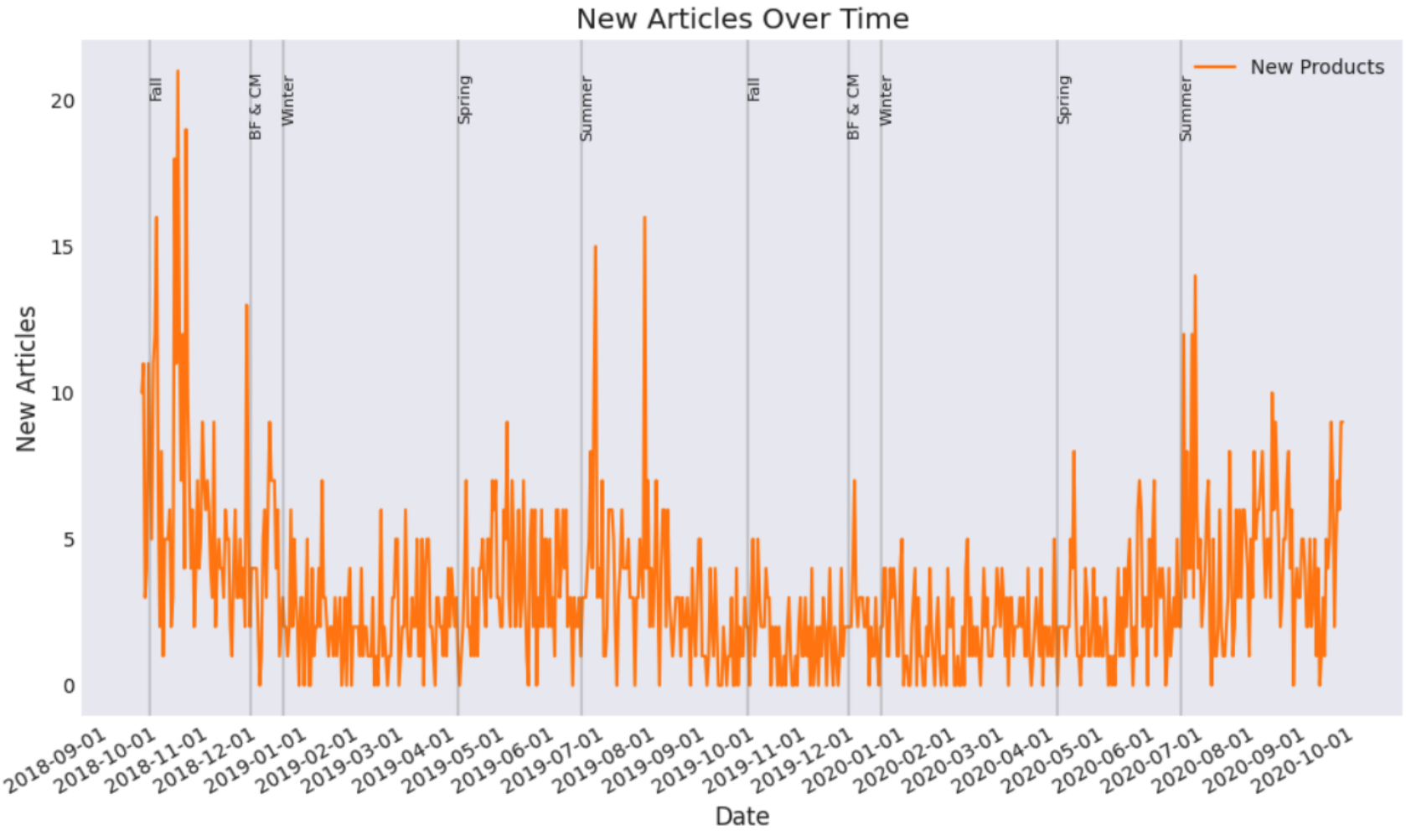
Pleated mesh dress that is transparent at the top with a small frilled collar and opening with a button at the back of the neck. Concealed zip in the side, a seam at the waist and a double-layered skirt. Lined.

ID: 577399004
Share: 0.00%

Shirt dress in a crêpe weave with a collar, button placket and short dolman sleeves. Yoke with a pleat at the back, and slits in the sides. Unlined.

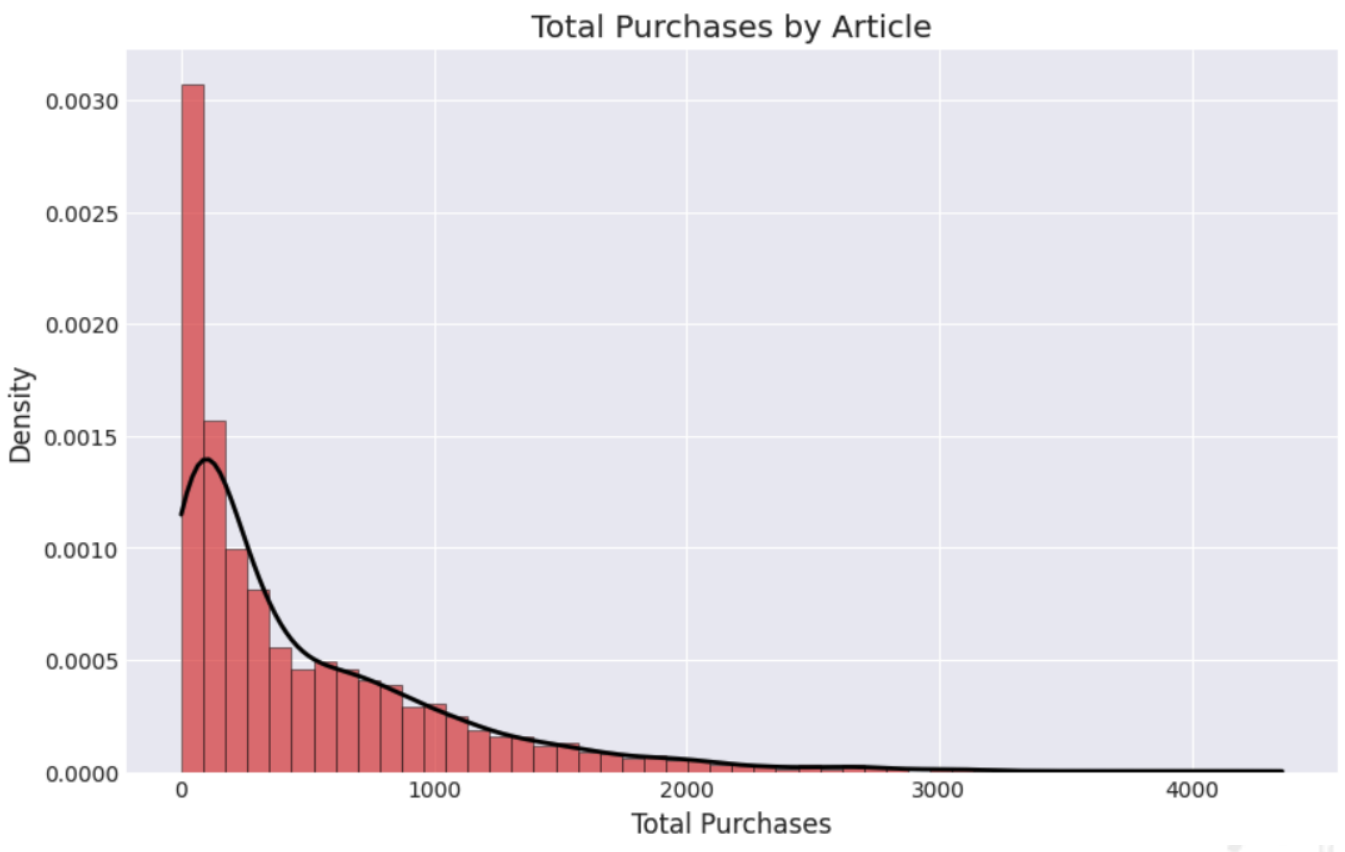
Dresses: New Articles Launched

New dresses show up continously over time.



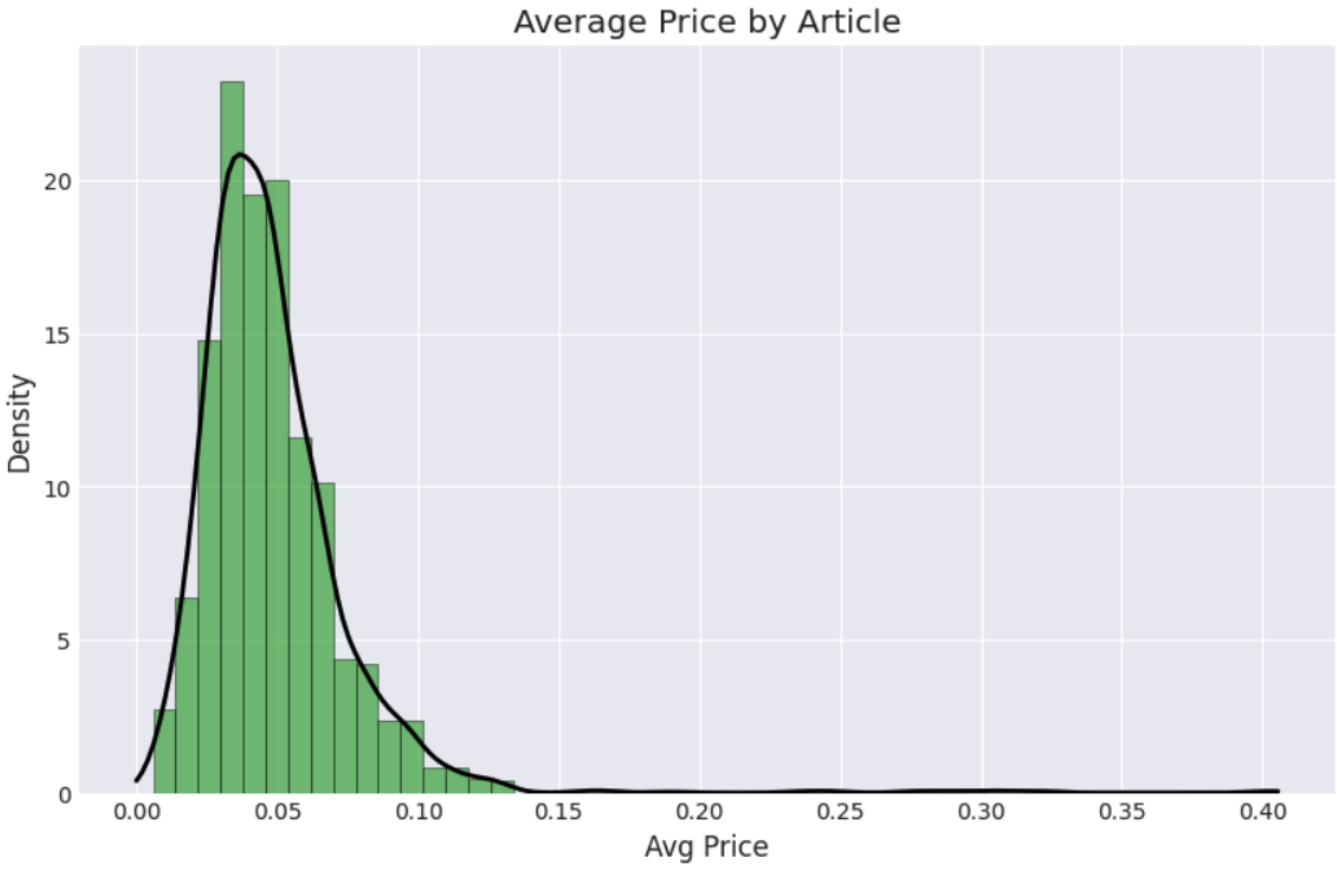
Dresses: Distribution of Sales

A few dresses pick up a majority of the sales share.



Dresses: Distribution of Prices

The price measure in the data has been scaled down.



Model 1: Image Regression with CNN

I estimate a CNN model on raw image pixels to predict sales share:

$$s_j = g(M_j; \theta)$$

and get an out of sample R2 of approximately 0.35.

Occlusion can reveal how inputs map to outputs.

RF Model 1: Counterfactuals

I use Dalle to generate images using the best seller as base to generate new designs. The predicted market shares are:

0.024%

0.003%

0.004%

0.003%

0.005%

0.008%

$$s_j = g(M_j; \theta)$$

Additional Reduced Form Evidence

- Data $\{M_j, p_j, s_j\}_{j=0}^{J}$ is split into training (80%) and test (20%) datasets
- Models tried: Linear Regression, Ridge, Random Forests, Neural Nets, Light Gradient Boosting, Ensemble
- Feature Sets:
  - 512 Image-embeddings: $e(M_j)$
  - 495 TDIDF: $t(T_j)$
  - 410 Sentence BERT: $b(T_j)$
  - 1417 Image-embeddings + BERT + TDIDF: $e(M_j), b(T_j), t(T_j)$
  - 50 Image-embeddings + BERT + TDIDF (factors): $f[e(M_j), b(T_j), t(T_j)]$
- Regularization: L2 parameter 5.0 for Ridge and Neural Nets, Min_samples_leaf = 10 for random forests and min_data_leaf=100 for LGBM. Voting (Ensemble) averages neural net and LGBM.

Product Usage

Features from customer demographics aggregated to product level.

Table 2: Log Sales and Prices on Product Usage

Model	R2 Scores for Log Sales		R2 Scores for Log Prices	
	Train	Test	Train	Test
OLS	0.03	0.04	0.02	0.02
Ridge	0.01	0.01	0.00	0.00
Random Forests	0.41	0.20	0.33	-0.01
Deep Nets	0.00	0.01	-0.10	-0.08
LGBM	0.38	0.17	0.27	-0.02
Voting	0.25	0.15	0.16	0.01

Product Categories

Large number of organizational dummies and other dummies related to product color, texture, etc.

Table 3: Log Sales and Prices on Product Categories

Model	R2 Scores for Log Sales		R2 Scores for Log Prices	
	Train	Test	Train	Test
OLS	0.62	-7.05e+22	0.16	-3.79e+24
Ridge	0.59	0.61	0.14	-0.04
Random Forests	0.58	0.60	0.17	0.01
Deep Nets	0.64	0.62	0.07	-0.01
LGBM	0.41	0.40	0.10	-0.03
Voting	0.56	0.56	0.09	-0.01

Image-Embeddings

Image-Em

Table 4: Log Sales and Prices on Image Features

Model	R2 Scores for Log Sales		R2 Scores for Log Prices	
	Train	Test	Train	Test
OLS	0.68	0.22	0.69	0.29
Ridge	0.65	0.38	0.66	0.42
Random Forests	0.78	0.43	0.76	0.41
Deep Nets	0.66	0.45	0.55	0.45
LGBM	0.89	0.46	0.88	0.42
Voting	0.82	0.49	0.76	0.47

BERT

Table 5: Log Sales and Prices on BERT Features

Model	R2 Scores for Log Sales		R2 Scores for Log Prices	
	Train	Test	Train	Test
OLS	0.72	0.38	0.27	-0.44
Ridge	0.38	0.29	0.06	-0.03
Random Forests	0.83	0.59	0.43	-0.11
Deep Nets	0.35	0.26	0.02	-0.01
LGBM	0.92	0.69	0.52	-0.19
Voting	0.73	0.55	0.34	-0.04

TFIDF

Table 6: Log Sales and Prices on TFIDF Features

Model	R2 Scores for Log Sales		R2 Scores for Log Prices	
	Train	Test	Train	Test
OLS	0.73	-1.54e+20	0.27	-3.78e+19
Ridge	0.57	0.46	0.12	-0.05
Random Forests	0.74	0.57	0.33	-0.06
Deep Nets	0.56	0.45	0.03	-0.01
LGBM	0.76	0.56	0.35	-0.07
Voting	0.69	0.53	0.22	-0.01

All Features

Table 7: Log Sales and Prices on All Features

Model	R2 Scores for Log Sales		R2 Scores for Log Prices	
	Train	Test	Train	Test
OLS	0.99	-0.77	0.95	-4.61
Ridge	0.88	0.72	0.75	0.39
Random Forests	0.86	0.65	0.77	0.39
Deep Nets	0.83	0.70	0.58	0.45
LGBM	0.95	0.71	0.90	0.43
Voting	0.91	0.74	0.78	0.47

All Features Compressed

Factor analysis is used to compress features to 50 dimensions.

Table 8: Log Sales and Prices on All Features Compressed to 50

Model	R2 Scores for Log Sales		R2 Scores for Log Prices	
	Train	Test	Train	Test
OLS	0.56	-1.17e+10	0.39	-3.70e+11
Ridge	0.56	0.53	0.39	0.35
Random Forests	0.75	0.49	0.66	0.33
Deep Nets	0.83	0.59	0.43	0.36
LGBM	0.83	0.55	0.76	0.34
Voting	0.84	0.60	0.63	0.37

Attention-Based Embeddings

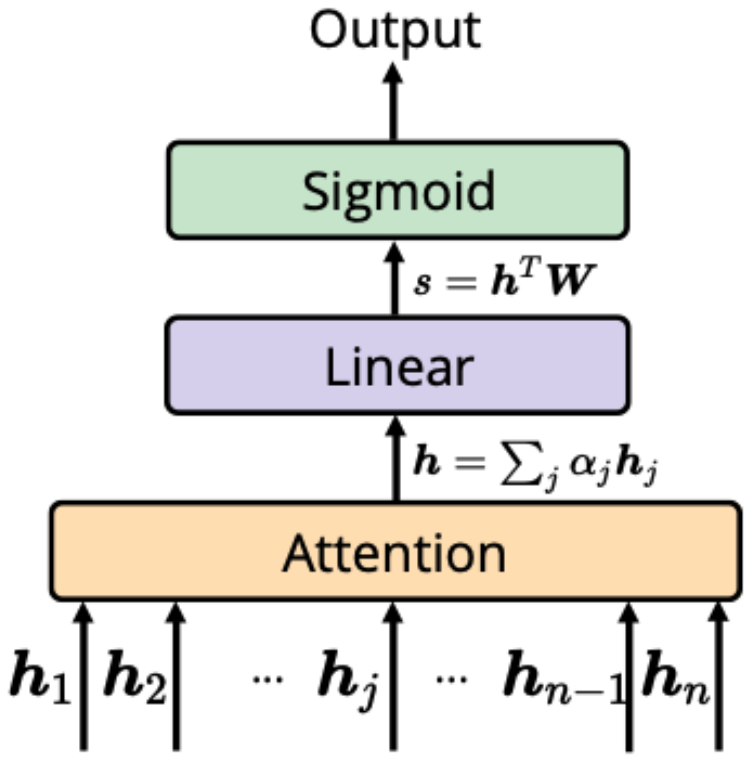
Moving beyond word counting, we need to understand how words interact in a given context:

- **Text Description:** $T = (w_1, w_2, w_3, ..., w_N)$
- **Word Embeddings:** $e_W(T) = (h_1, h_2, h_3, ..., h_N)$
- **Attention Weight:** $\alpha_n = \frac{\exp\{h_n^T V/\lambda\}}{\sum_{n'} \exp\{h_{n'}^T V/\lambda\}}$ tells us the importance of word w_n in determining the "context" of the entire description T.
- **Context:** $h = \sum_n \alpha_n h_n$ A high dimensional representation of the entire text T, which allows us to differentiate the text along multiple "attributes".
- **Scaler Prediction:** $s = h^T W$ picks up relevant contexts.

Trainable parameters: query weights V, context weights W.

Attention-Based Embeddings

The high dimensional context layer $h = e(T)$ will now represent our entire text T.



We can train this ourselves, or use a pre-trained model.

Model 3: Text Regression with Word Counts

I generate dummies from the text based on n-grams and estimate a Ridge regression on log sales to get an out of sample **R2: 0.37**.

Top 10 β

1. crêpe weave neck
2. neckline
3. viscose
4. sheen opening
5. wide flounce hem unlined
6. short fitted dress velour
7. buttons short
8. calf length dress
9. neck rounded
10. sleeves smocking

Bottom 10 β

1. straight
2. tie
3. dress
4. short dress satin
5. unlined
6. elasticated cuffs
7. appliqués
8. striped
9. seam waist slit
10. shoulder dress

Model 3: Variation by Age

Keyword significance varies across age groups, emphasizing diverse product preferences. "Young" is below 21, and "old" is above 60.

All ($R^2 = 0.37$)

- crêpe weave neck
- neckline
- viscose
- sheen opening
- wide flounce hem unlined

Young ($R^2 = 0.38$)

- short dress
- cuffs
- long wide sleeves
- opening narrow ties
- opening narrow

Old ($R^2 = 0.54$)

- viscose
- viscose weave neck
- wide flounce hem unlined
- buttons short
- shoulder straps buttons

Model 3: Machine Learning Methods

Moving beyond linear models, I find that nonlinear models can better extract generalizable signals.

Table 9: R2 Scores for Models on Log Shares

Model	R2 Train	R2 Test
OLS	1.00	0.00
Ridge	1.00	0.66
Random Forests	0.76	0.59
Deep Nets	0.98	0.75
Boosting Machines	0.77	0.59
Ensemble	0.93	0.74

Due to large covariate set ($\geq 10,000$), OLS completely overfits.

Model 4: Sentence BERT

Sentence-BERT uses attention mechanisms, along with other techniques, to create 410 dimensional sentence embeddings $e_B(T_j)$.

Table 10: R2 Scores for Log Sales on Sentence-BERT Features

Model	Train	Test
OLS	0.73	0.40
Ridge	0.73	0.42
Random Forests	0.83	0.61
Deep Nets	0.95	0.66
Boosting	0.92	0.68
Ensemble	0.96	0.72

Model 4: Visualization

I discover natural "topics" in the text embedding through clustering and find representative images for them.

tSNE and Kmeans Clustering

I compress the 512-dim image embeddings into 2 dimensions via t-SNE and find clusters:

- **t-SNE (t-distributed Stochastic Neighbor Embedding):**
  - Reduces dimensions by minimizing the Kullback-Leibler divergence between two distributions:
  - $C = KL(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$
  - p_{ij} - probability of picking j as neighbor of i in high-dimensional space.
  - q_{ij} - probability in the low-dimensional embedding.
- **K-Means Clustering:**
  - Objective is to partition n observations into k clusters by minimizing within-cluster variances:
  - $J = \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2$
  - μ_i is the mean of points in S_i.

Autoencoders

We want to compress $e_O(M_j), e_B(T_j)$ without losing too much information (e.g. tSNE):

- **Autoencoders**: Estimate a neural network to predict $e_O(M_j), e_B(T_j)$ from itself, and use hidder layer outputs.
  - Input data: $\mathbf{x} = [e_O(M_j), e_B(T_j)]$
  - Encoder function: $\mathbf{f}$ (high dim to low dim)
  - Decoder function: $\mathbf{r}$ (low dim to high dim)
- **Reconstruction Loss**: Mean squared error (MSE)

$$L(\mathbf{x}, \mathbf{f}, \mathbf{r}) = \|\mathbf{x} - r(f(\mathbf{x}))\|^2 \tag{1}$$

- **Latent Space**: The encoder maps the input to the latent space $f(\mathbf{x})$ (hidden representation), which the decoder then uses to reconstruct the input.

I compress the image + text embeddings into 100 dimensions: $X_j = f(e_O(M_j), e_B(T_j))$ for structural modelling.

Market Demand

For J products and N individuals, we have utilities given by,

$$u_{ij} = \alpha \log p_j + g(X_j) + \xi_j + \epsilon_{ij}$$

$$X_j = f(e_O(M_j), e_B(T_j))^{15}$$

this gives rise to market demand,

$$\log s_j = \log s_0 + \alpha \log p_j + g(X_j) + \xi_j$$

y_{ij}	Purchase of j product by consumer i	e_O, e_B	Embedding Layers
p_j	Price of product j	α	Price Elasticity
M_j	Image of product j	f	Autoencoder Layer
T_j	Text of product j	ϵ_{ij}	IID Gumbel Error
s_j	Sales share of product j	ξ_j	Market demand shock

[15] I use autoencoders for compressing embeddings into 100 dimensions

Market Demand: Assumptions

- Utility is linearly separable and ϵ_{ij} is IID.
- Endogenous Prices: $\xi_j \not\perp p_j | (M_j, T_j)$
- Exogenous Characteristics: $\xi_j \perp (M_j, T_j)$
- We have $Kx1$ instruments such that $E[\xi_j Z_{jk}] = 0$
  - Diffentiation IVs: Distances between X_j
  - BLP IVs: Sums of other product characteristics X_{-j}.
- Instrument logic: Proximity to competing products affects sales through reduced markups (prices) but does not enter utility directly.

Market Demand: Estimation

Here we draw from the debiased machine learning literature (Chernozhukhov et al., 2016, 2018):

- Partially Linear Regression (PLR):
  - $y_j = \theta T_j + g(X_j) + \epsilon_j$
  - $E[\epsilon_j | T_j, X_j] = 0$
- Partially Linear Instrumental Variables (PLIV)
  - $y_j = \theta T_j + g(X_j) + \epsilon_j$
  - $E[\epsilon_i | Z_j, X_j] = 0$
- Estimation:
  - ML + regularization used to approximate g.
  - First stage: Partial out $[X, Z]$ from T and y using dataset A
  - Second stage: Regress residuals on data B to estimate α

Market Demand: Results

y_j: log shares, $\log s_j$ T_j: log prices, $\log p_j$ X_j: Autoencoded Embeddings

Table 11: Coefficient Table

Model	OLS	PLR	IV	PLIV
α	-0.54	-0.41	-0.84	-0.52
Std. Error	0.08	0.08	0.17	0.18
t-value	-7.13	-4.95	-4.88	-2.89
p-value	0.00	0.00	0.00	0.00
LB 0.025	-0.69	-0.57	-1.18	-0.88
UB 0.975	-0.39	-0.25	-0.50	-0.17
N	2319	2319	2319	2319

First stage with LGBMRegressor showed train R2 0.83, test R2 0.67; PLR & PLIV used 5-fold cross splitting and LGBM to estimate g with 100 min_data_leaf.

Consumer Choice: Inference

Farell et al., (2019, 2021) show that inference can be done by deep nets:

- Parametric Loss: $\theta(x) = \text{argmin}_\theta E[L(Y, T, \theta(X))]$
  - $\theta(x) = \text{argmin}_\theta E[L(Y, T, \theta(X))]$
  - Exp. Hessian: $\Lambda(x) = E[L_{\theta\theta}|X, T]$
- Inference Target: $\mu = E[H(X, \theta(X); T = t^*)]$
- Influence Function: $\Psi(y, t, x) = H - H_\theta \Lambda(x)^{-1} L_\theta$
  - Measures the change in target μ for a infinitesimal change in observation (y, t, x).
- Estimation:
  - Deep neural networks $\hat{\theta}(x)$ that minimize empirical loss.
  - $\hat{H}_\theta, \hat{L}_\theta, \hat{\Lambda}(x_i)$ obtained by automatic differentiation.
- Inference:
  - From $\hat{\theta}(x_i), \hat{\Lambda}(x_i)$ get $\hat{\Psi}(y_i, t_i, x_i)$.
  - $\hat{\mu} = \frac{1}{n} \sum_i \hat{\Psi}(y_i, t_i, x_i)$
  - $\hat{\text{Avar}}(\hat{\mu}) = \frac{1}{n} \sum_i (\hat{\Psi}(y_i, t_i, x_i) - \hat{\mu})^2$

Deep Logit: Simulation Study I

For $J = 1000$, I generate images $M_j(Jx3x1024x1024)$ from the MNIST Digits dataset. I generate u_{ij} according to,

$$u_{ij} = \log(1 + 0.1\text{label}(M_j)) + \epsilon_{ij}$$

where label gives the actual digit value of the image of the digit.

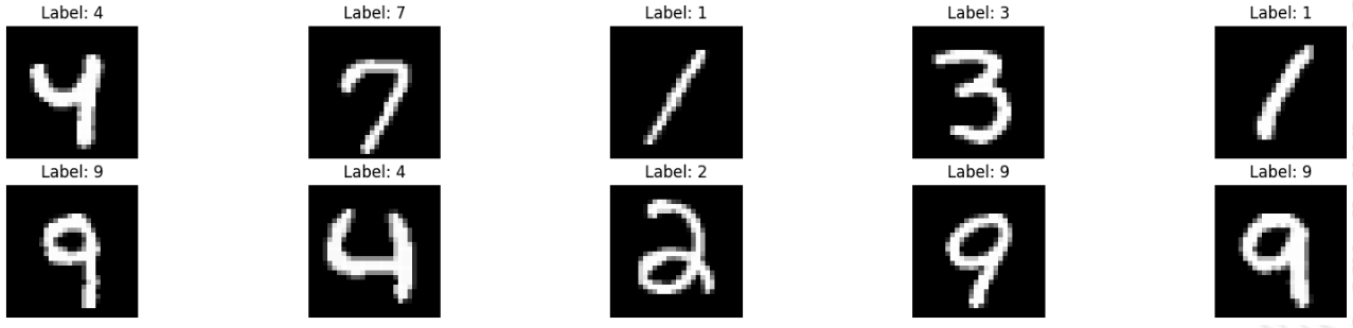


Figure 3: Product Images

Deep Logit: Simulation Study II

I estimate model and check the average scores of $\hat{g}(M_j)$ when label$(M_j) = j$

$$u_{ij} = g(M_j) + \epsilon_{ij}$$

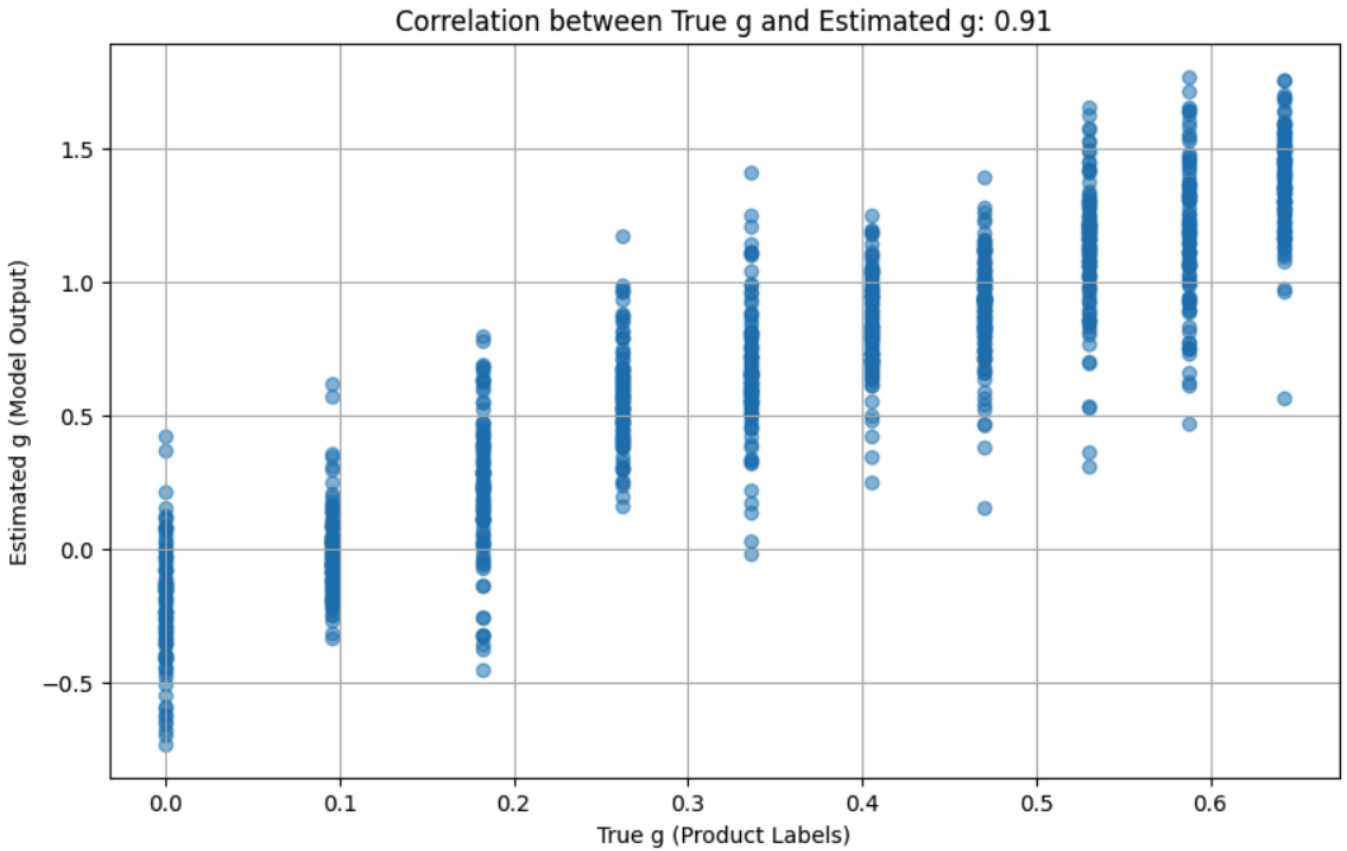


Figure 4: Product Images

References i

Steven T. Berry and Philip A. Haile.

Foundations of demand estimation.

In Kate Ho, Ali Hortaçsu, and Alessandro Lizzeri, editors, *Handbook of Industrial Organization*, volume 4 of *Handbook of Industrial Organization, Volume 4*, pages 1–62. Elsevier, January 2021.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins.

Double/Debiased Machine Learning for Treatment and Causal Parameters, December 2017.

arXiv:1608.00060 [econ, stat].

References ii

Victor Chernozhukov, Whitney Newey, Vıctor M. Quintas-Martınez, and Vasilis Syrgkanis.

RieszNet and ForestRiesz: Automatic Debiased Machine Learning with Neural Nets and Random Forests.

In *Proceedings of the 39th International Conference on Machine Learning*, pages 3901–3914. PMLR, June 2022.

Giovanni Compiani, Ilya Morozov, and Stephan Seiler.

Demand Estimation with Text and Image Data, 2023.

Max H. Farrell, Tengyuan Liang, and Sanjog Misra.

Deep Learning for Individual Heterogeneity: An Automatic Inference Framework, July 2021.

arXiv:2010.14694 [cs, econ, math, stat].

References iii

Max H. Farrell, Tengyuan Liang, and Sanjog Misra.

Deep Neural Networks for Estimation and Inference.

Econometrica, 89(1):181–213, 2021.

Sukjin Han, Eric H. Schulman, Kristen Grauman, and Santhosh Ramakrishnan.

Shapes as Product Differentiation: Neural Network Embedding in the Analysis of Markets for Fonts, March 2024.

arXiv:2107.02739 [cs, econ].

Jens Ludwig and Sendhil Mullainathan.

Machine Learning as a Tool for Hypothesis Generation.

NBER Working Papers, March 2023.

References iv

Lorenzo Magnolfi, Jonathon McClure, and Alan T. Sorensen.

Triplet Embeddings for Demand Estimation, October 2023.

Thomas W Quan and Kevin R Williams.

Extracting Characteristics from Product Images and its Application to Demand Estimation.

Kenneth E. Train.

Discrete Choice Methods with Simulation.

Cambridge University Press, June 2009.

Google-Books-ID: 4yHaAgAAQBAJ.

Chengjun Zhang.

Essays on Consumer Heterogeneity and Personalized Discounts in an Online Market.

PhD Thesis, Georgetown University, February 2024.