Style Synth:

ML Fashion Recommendation System

Courtney Green, Li-Wen Hu, Satomi Ito, Nandini Kodali, Sophia Rutman

Agenda

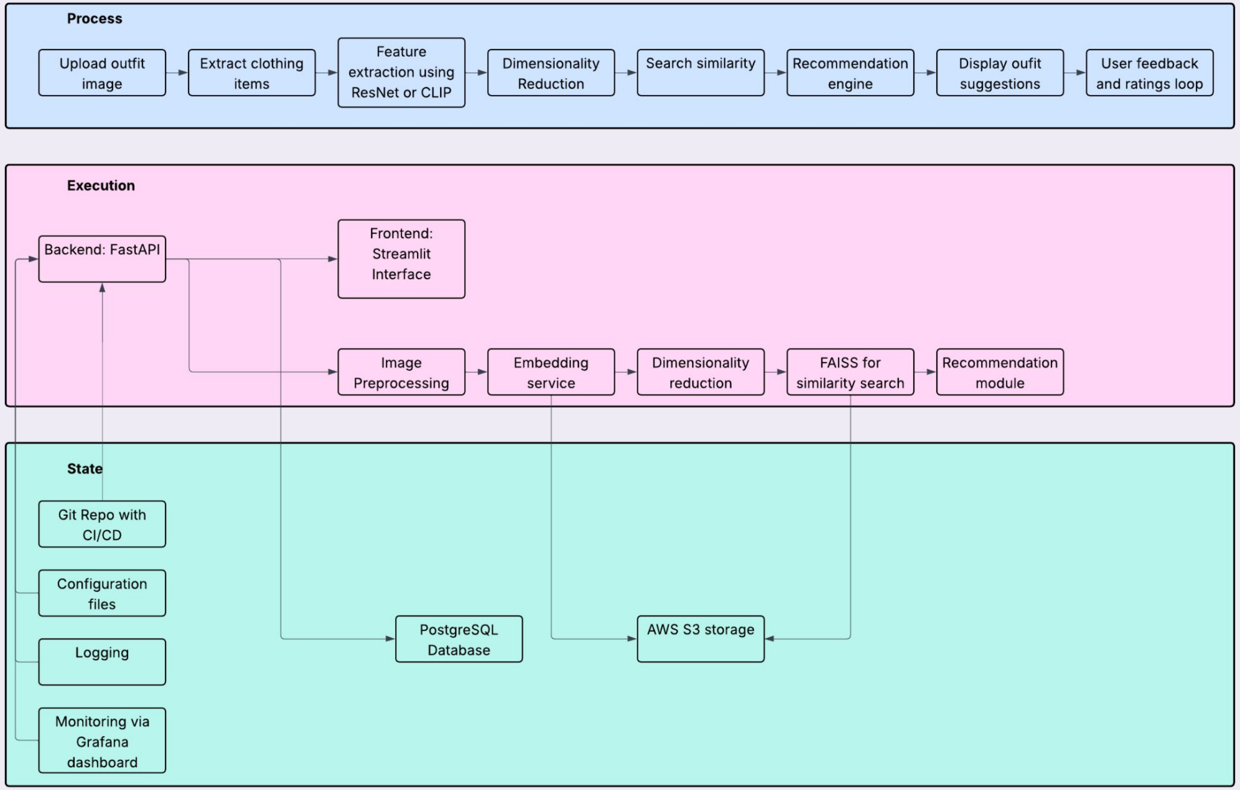
1

Introduction

2

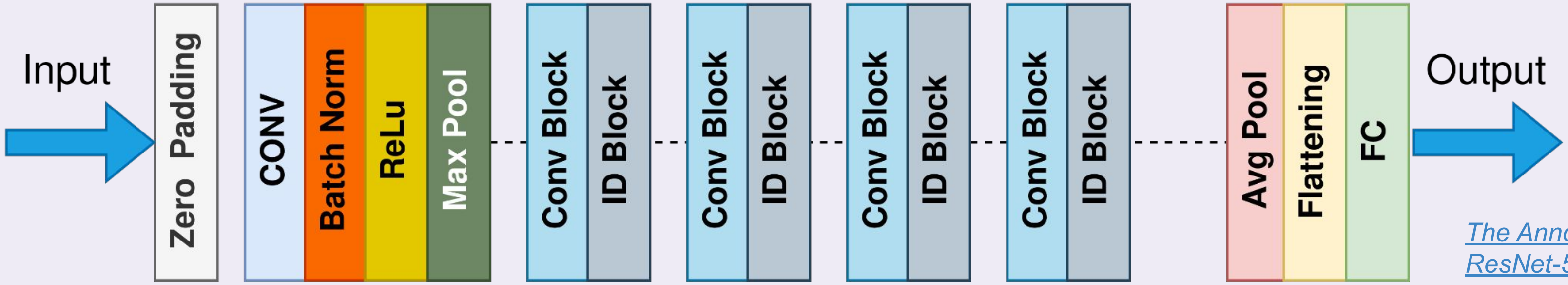
Computer Vision

Solution Architecture



End-to-end ML pipeline with FastAPI backend and Streamlit frontend. Uses ResNet for feature extraction, PCA for dimensionality reduction, and FAISS for similarity search. Data persisted in PostgreSQL and S3, with Grafana monitoring.

ResNet-50 Feature Extractor



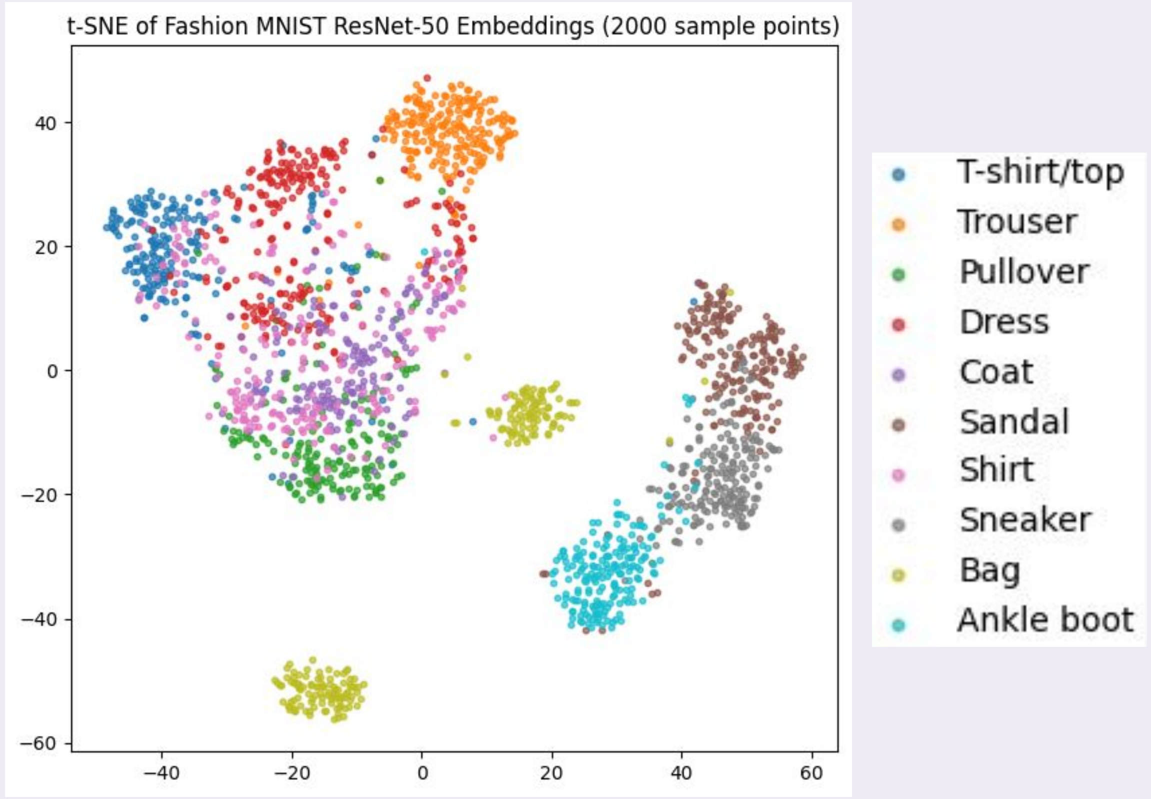
The Annotated ResNet-50, Towards Data Science

- Captures low-level visual features like edges
- Batch normalization stabilizes training and inference
- Max pooling reduces spatial size for efficiency

- Learn mid- to high-level visual features of images
- Residual connections preserve information across layers
- Form the core representation used for similarity and clustering

- Aggregates spatial information across the entire image
- Convert feature maps into a fixed-length vector
- Produces a 2048-dimensional image embeddings

Embedding Quality Check

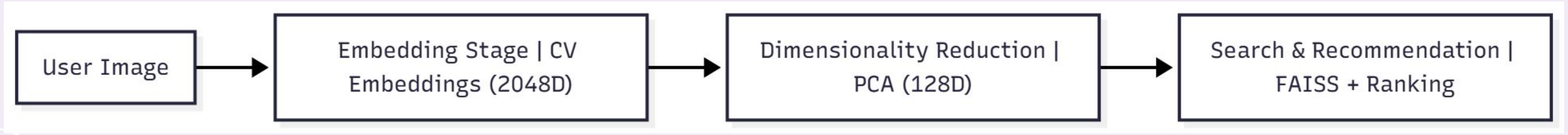


3

Recommendation Engine

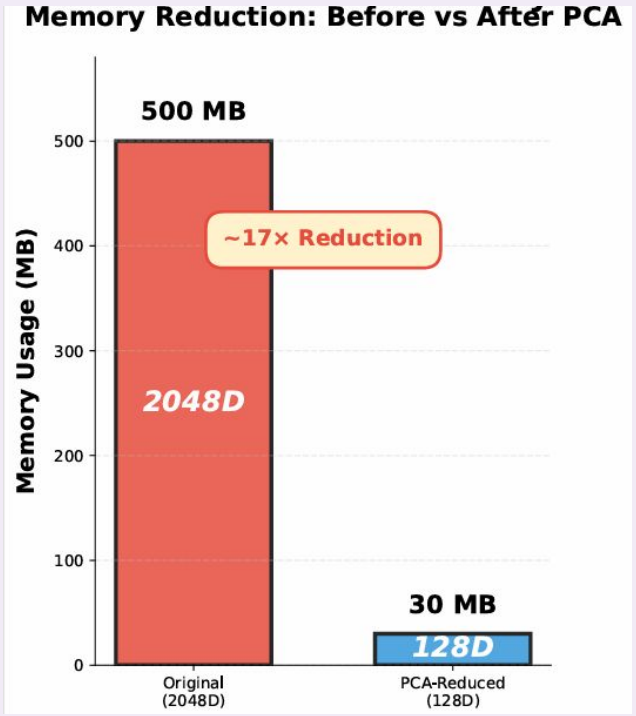
Fashion Recommendation Engine

- Uses ResNet-50 embeddings from the CV pipeline (2048D)
- Applies PCA for dimensionality reduction (2048D → 128D)
- Uses FAISS for fast similarity search
- Implements ranking, filtering, and metadata-aware recommendations
- Designed to be production-ready (save/load, tests, benchmarks)


User Image
Embedding Stage | CV Embeddings (2048D)
Dimensionality Reduction | PCA (128D)
Search & Recommendation | FAISS + Ranking

Dimensionality Reduction with PCA

- Input: 2048D embeddings from ResNet-50
- PCA reduces to configurable dimensions
- ~95% variance retained at 128D
- Memory: ~500MB → ~30MB (≈17× reduction)
- Enables faster search and lower resource usage

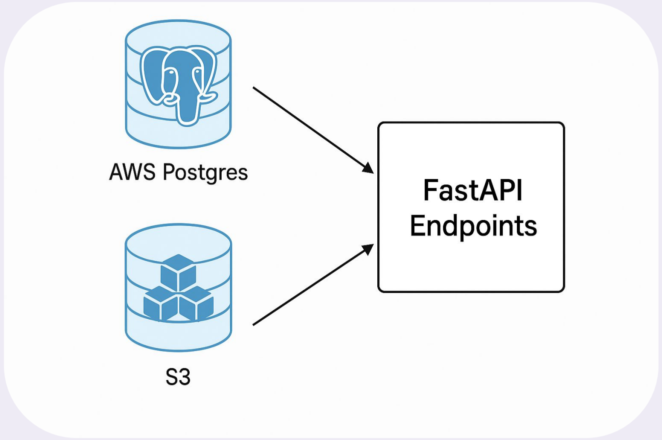


FAISS & Recommendation Logic

- FAISS index with L2 or cosine similarity
- Index build time: < 1 second
- Search speed: ~10,000+ queries/second
- Recommendation logic:
    a. Distance-based ranking
    b. Class filtering (e.g., only tops, only trousers)
    c. Exclusion filtering (avoid specific items)
    d. Optional metadata-aware filtering

4

Backend Web Server



FastAPI endpoints:

1. wardrobe/upload
2. wardrobe/items
3. outfits/saved
4. predict
5. outfits/generate
6. outfits/save
7. outfits/{outfit_id}
8. wardrobe/item/{item_id}
9. wardrobe/clear-all

PostgreSQL Schema

wardrobe_items
item_id | **PK**
image_url
category
metadata

image_url: Publicly Accessible S3 URL
category: top, bottom, shoes, outerwear
Metadata: user inputs of brand, occasion, and any other notes they chose to include

embeddings
id | **PK**
item_id | **FK**
embedding
created_at

embedding: /predict generates entry into embeddings, using pretrained outfit recommender

saved_outfits
id | **PK**
outfit_id | **PK**
items
occasion
season
created_at

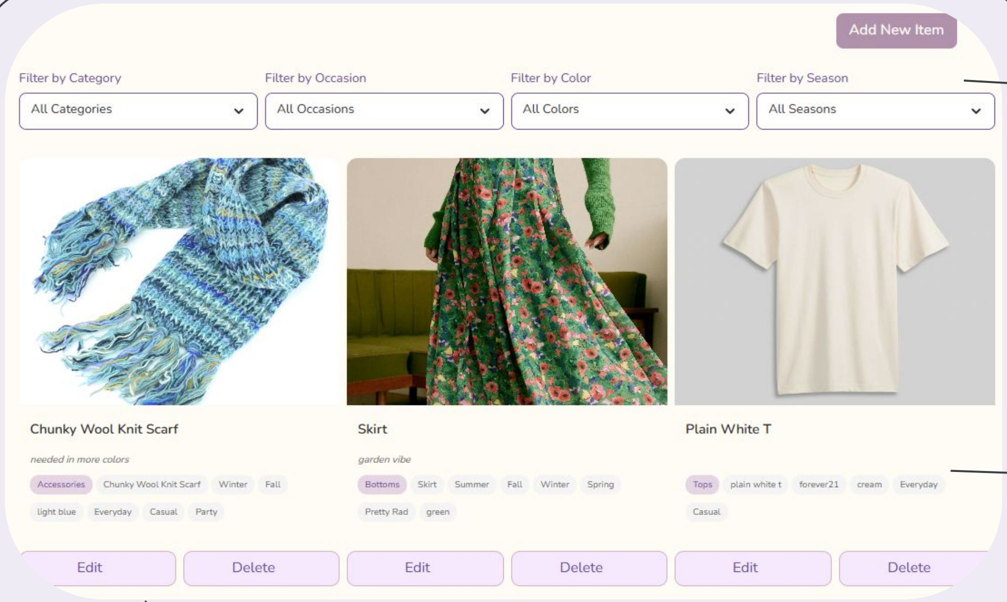
items: list of wardrobe items

5

Frontend

My Wardrobe

Add New Item
Allows user to add new items to their wardrobe.



Filter by Category: All Categories
Filter by Occasion: All Occasions
Filter by Color: All Colors
Filter by Season: All Seasons

Chunky Wool Knit Scarf
needed in more colors
Accessories · Chunky Wool Knit Scarf · Winter · Fall · light blue · Everyday · Casual · Party

Skirt
garden vibe
Bottoms · Skirt · Summer · Fall · Winter · Spring · Pretty Rad · green

Plain White T
Tops · plain white t · forever21 · cream · Everyday · Casual

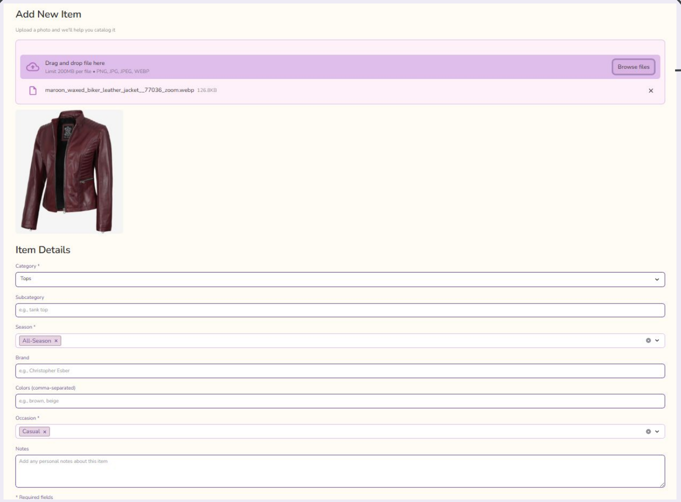
Filtering
Filter wardrobe items by category, occasion, color, or season for easy browsing.

Tags
Label items with category, season, occasion, colors, and brand for easy sorting and outfit matching.

Edit & Delete
Modify and/or remove uploaded clothing items.

Upload New Items

Upload clothing photos with metadata (category, season, brand, colors, occasion). Images are processed to generate embeddings for AI-powered outfit recommendations.


Add New Item

Upload a photo and we'll help you catalog it

maroon_waxed_biker_leather_jacket__77036_zoom.webp 126.8KB

Item Details

Category *: Tops

Subcategory: e.g., tank top

Season *: All-Season

Brand: e.g., Christopher Esber

Colors (comma-separated): e.g., brown, beige

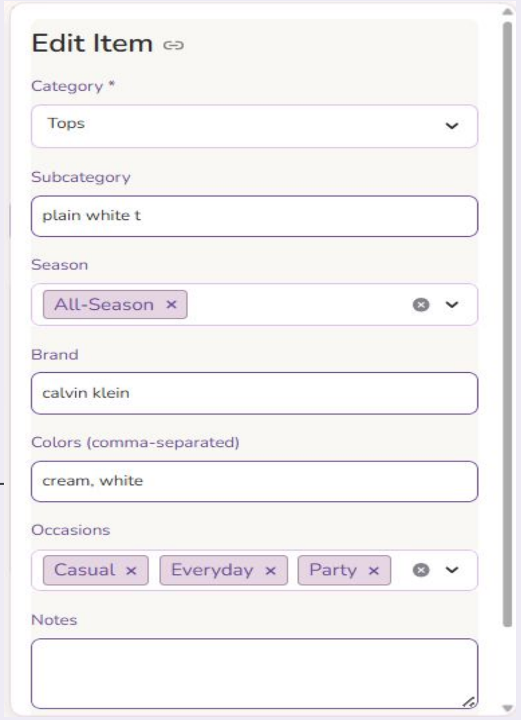
Occasion *: Casual

Notes: Add any personal notes about this item

* Required fields

Wardrobe Management

Full control over wardrobe items. Users can edit metadata like category, brand, colors, and occasions to improve outfit recommendations or safely remove items with deletion confirmation.


Edit Item

Category *: Tops

Subcategory: plain white t

Season: All-Season

Brand: calvin klein

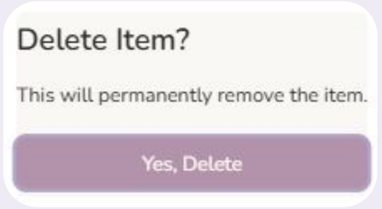
Colors (comma-separated): cream, white

Occasions: Casual, Everyday, Party

Notes:

My Wardrobe

11 items in your collection


Delete Item?

This will permanently remove the item.

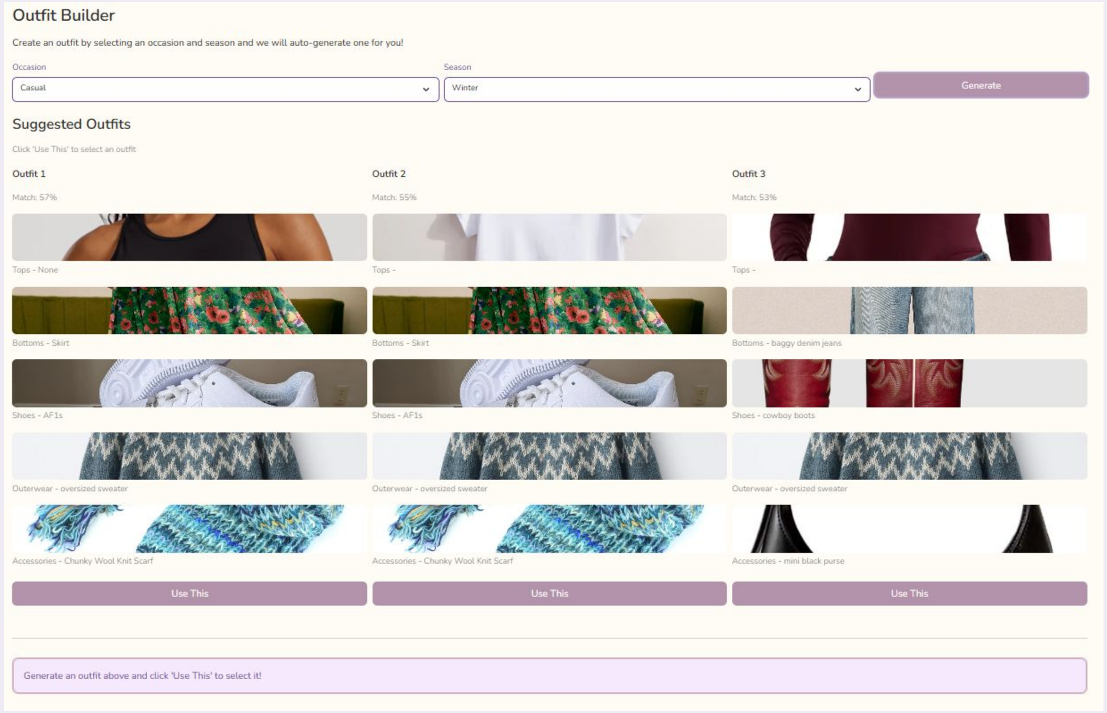
Outfit Builder

Suggested Outfits
3 ML-ranked outfit options based on occasion & season

Match Score
Embedding-based compatibility score using FAISS similarity search

Match: 58%

Your Outfit
Preview with item removal.



Outfit Builder

Create an outfit by selecting an occasion and season and we will auto-generate one for you!

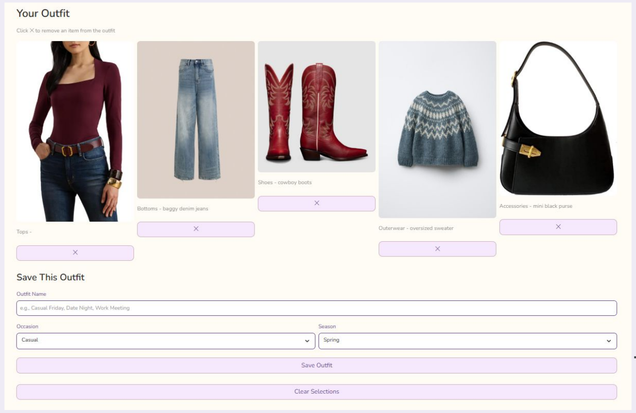
Occasion: Casual
Season: Winter

Suggested Outfits

Click 'Use This' to select an outfit

Outfit 1	Outfit 2	Outfit 3
Match: 57%	Match: 55%	Match: 53%
Tops - None	Tops -	Tops -
Bottoms - Skirt	Bottoms - Skirt	Bottoms - baggy denim jeans
Shoes - AF1s	Shoes - AF1s	Shoes - cowboy boots
Outerwear - oversized sweater	Outerwear - oversized sweater	Outerwear - oversized sweater
Accessories - Chunky Wool Knit Scarf	Accessories - Chunky Wool Knit Scarf	Accessories - mini black purse

Generate an outfit above and click 'Use This' to select it!



Your Outfit

Click ✕ to remove an item from the outfit

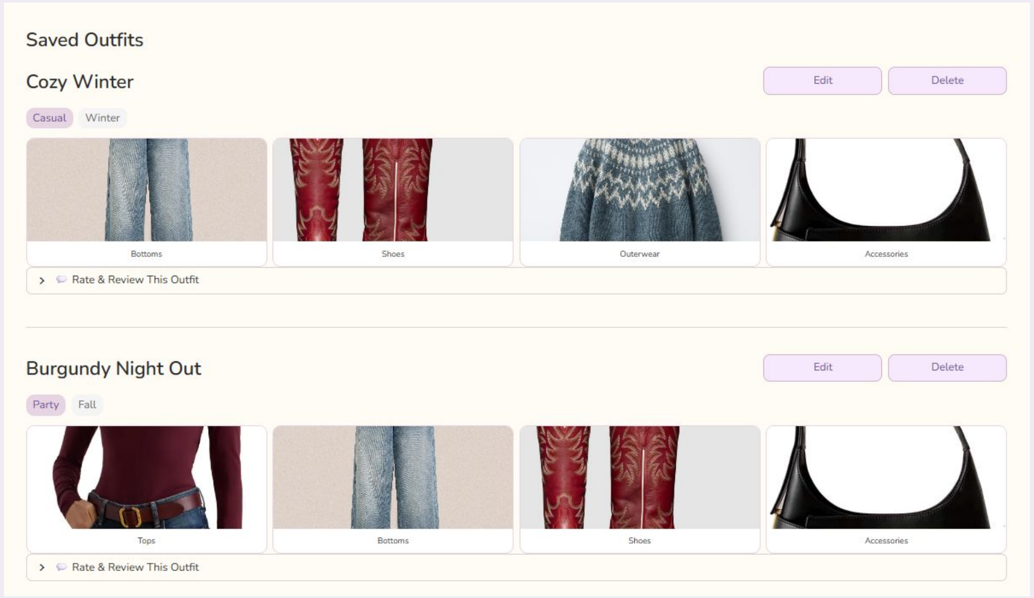
Tops -
Bottoms - baggy denim jeans
Shoes - cowboy boots
Outerwear - oversized sweater
Accessories - mini black purse

Save This Outfit

Outfit Name: e.g., Casual Friday, Date Night, Work Meeting
Occasion: Casual
Season: Spring

Save Outfit
Persist outfits with name, occasion & season tags

Saved Outfits



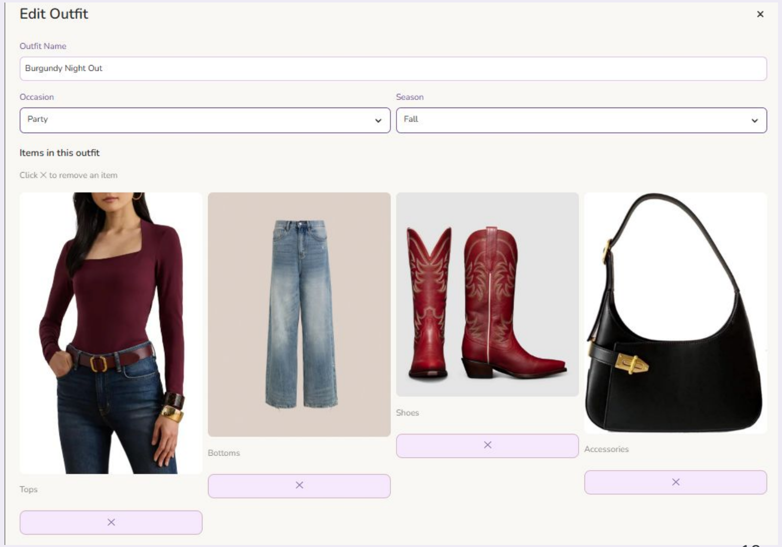
Saved Outfits

Cozy Winter

Burgundy Night Out

Edit Auto-Generated Outfit

Users can customize ML-generated outfits by editing the name, occasion, season, and removing individual items. This allows personalization of recommendations before or after saving.



Edit Outfit

Outfit Name: Burgundy Night Out

Occasion: Party

Season: Fall

Items in this outfit

Click ✕ to remove an item

Rate & Review Saved Outfits

Users can rate outfits (1-5 stars) and leave feedback comments. This feature captures user preferences for future model improvements. (Feedback collection in place; personalization coming soon)

6

Monitoring the app

Prometheus - Metrics Collection

- Prometheus: tracks our metrics
  - FastAPI application metrics
    - Request counts
    - Response times
  - Underlying ML metrics
    - Computer vision processing time
    - Recommendation engine query performance
  - Data-layer metrics
    - Postgres
    - AWS operations
  - Errors

Grafana: Visualization & Alerting

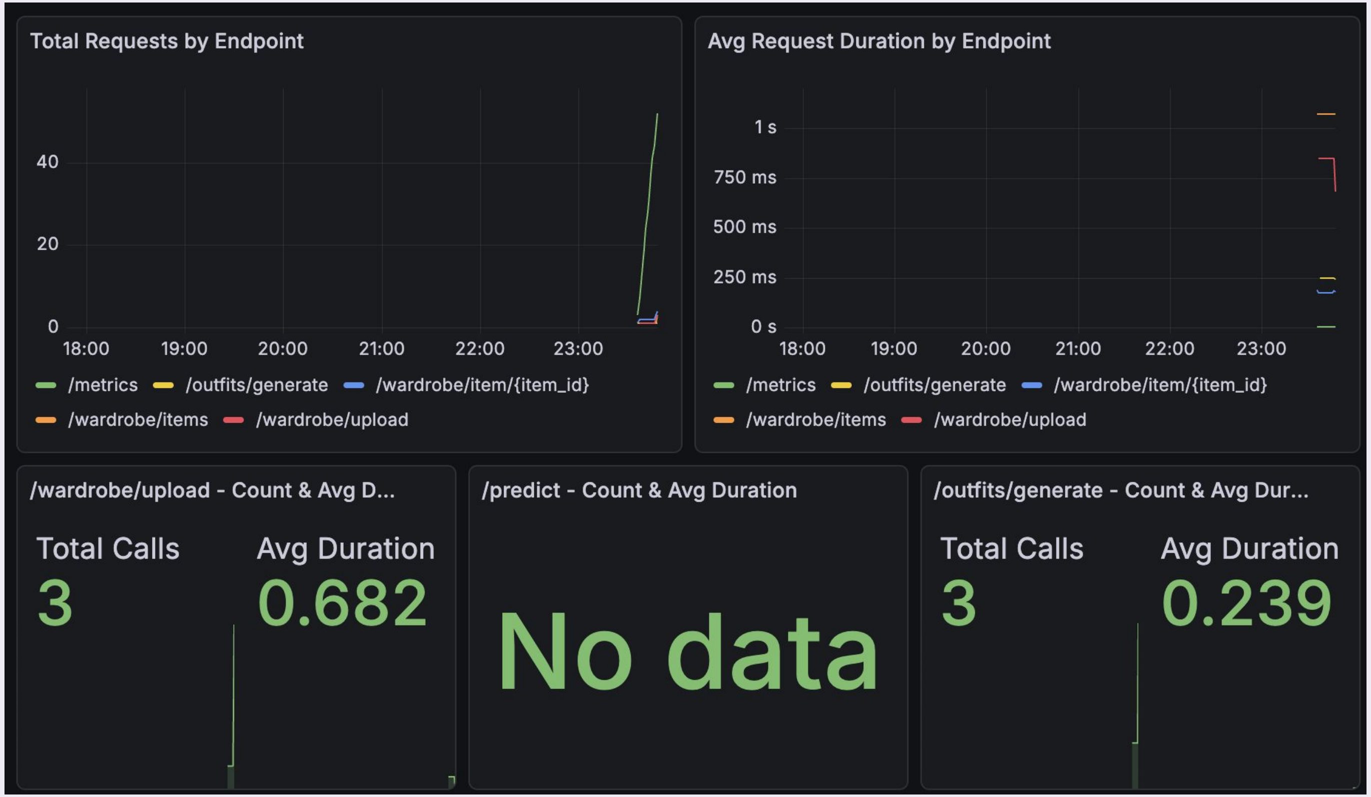
- Create custom dashboards to see API health
- Monitor recommendation engine performance
- Track user activity patterns
- Expose errors

Prometheus + Grafana: Working Together

- Monitoring Stack Architecture:
    - FastAPI -> Prometheus -> Grafana -> Users/DevOps
- Prometheus: The Data Collector & Storage
    - Scrapes metrics from FastAPI backend
    - Stores time-series data
    - Provides query language
- Grafana: The Visualization Layer
    - Connects to Prometheus as a data source
    - Transform raw metrics into visual dashboards

- **Why Use Both Together?**
    - **A standard monitoring stack that allows us to modify dashboards without disrupting data collection and scale both tools independently for production workloads**

1. API Metrics



/wardrobe/upload - Count & Avg D...

Total Calls	Avg Duration
3	0.682

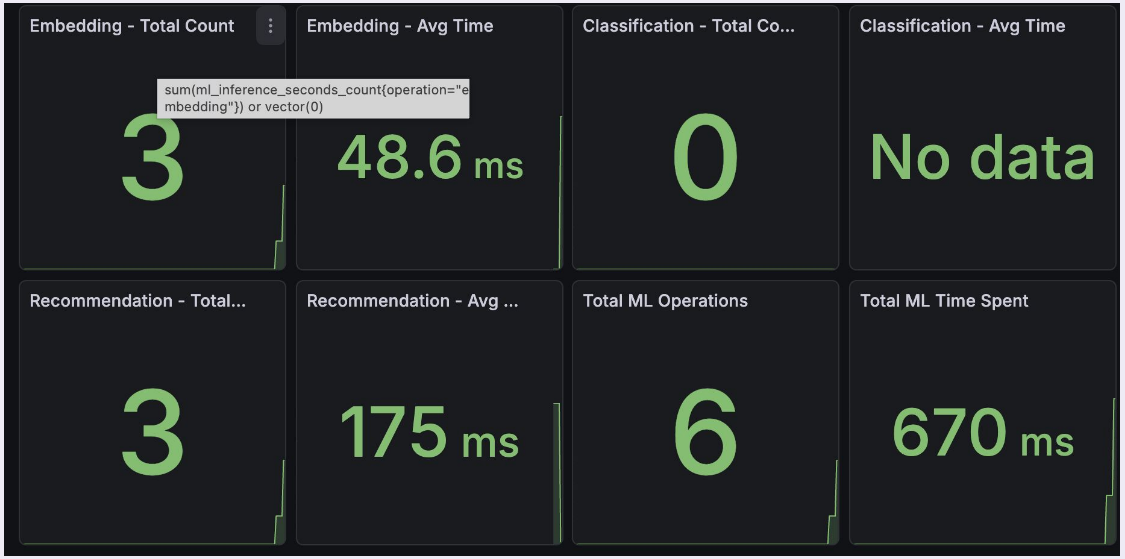
/predict - Count & Avg Duration

No data

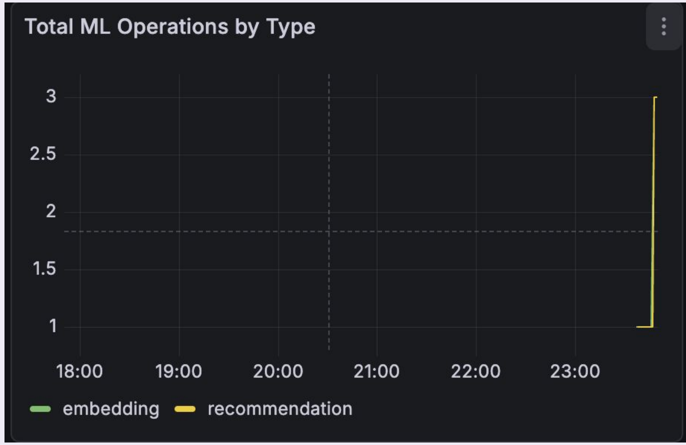
/outfits/generate - Count & Avg Dur...

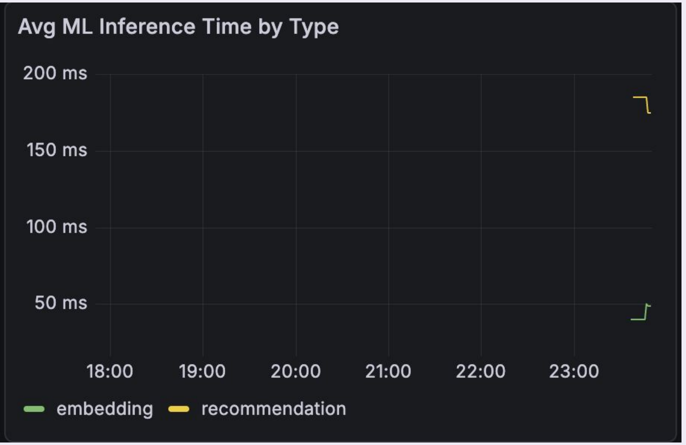
Total Calls	Avg Duration
3	0.239

2. ML Metrics

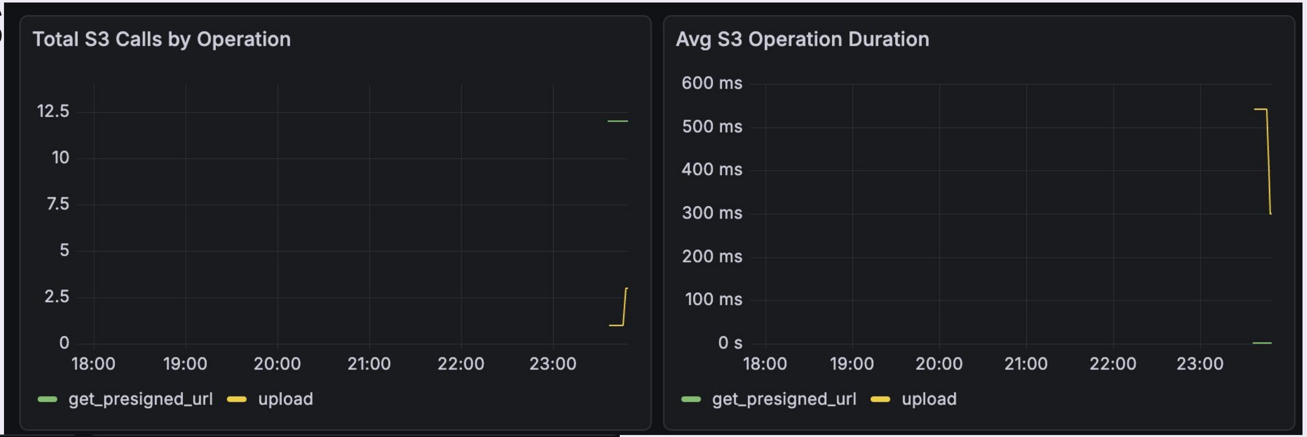


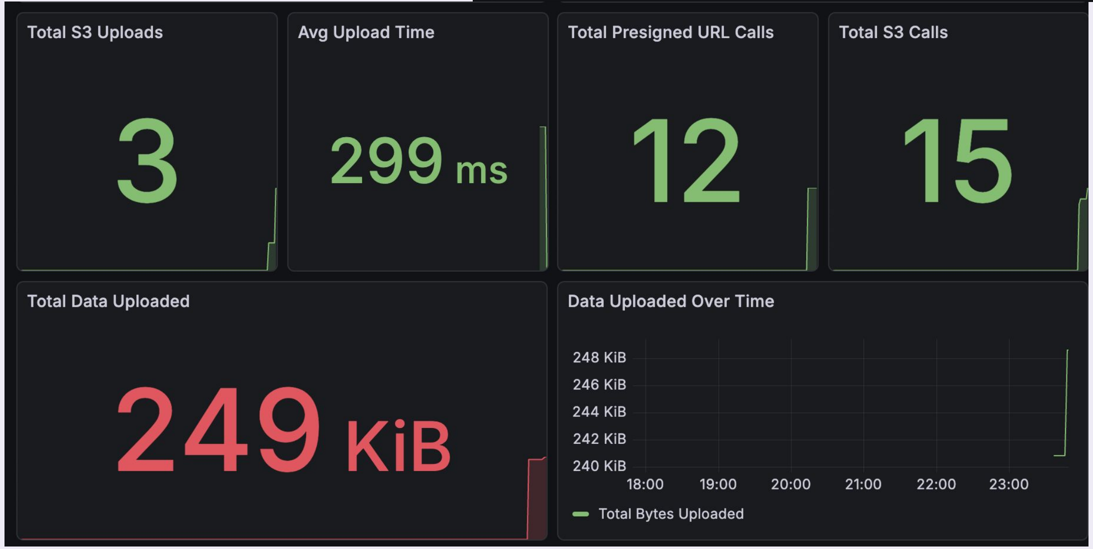
```
sum(ml_inference_seconds_count{operation="e
mbedding"}) or vector(0)
```





3. Data Metrics





Total S3 Uploads
3

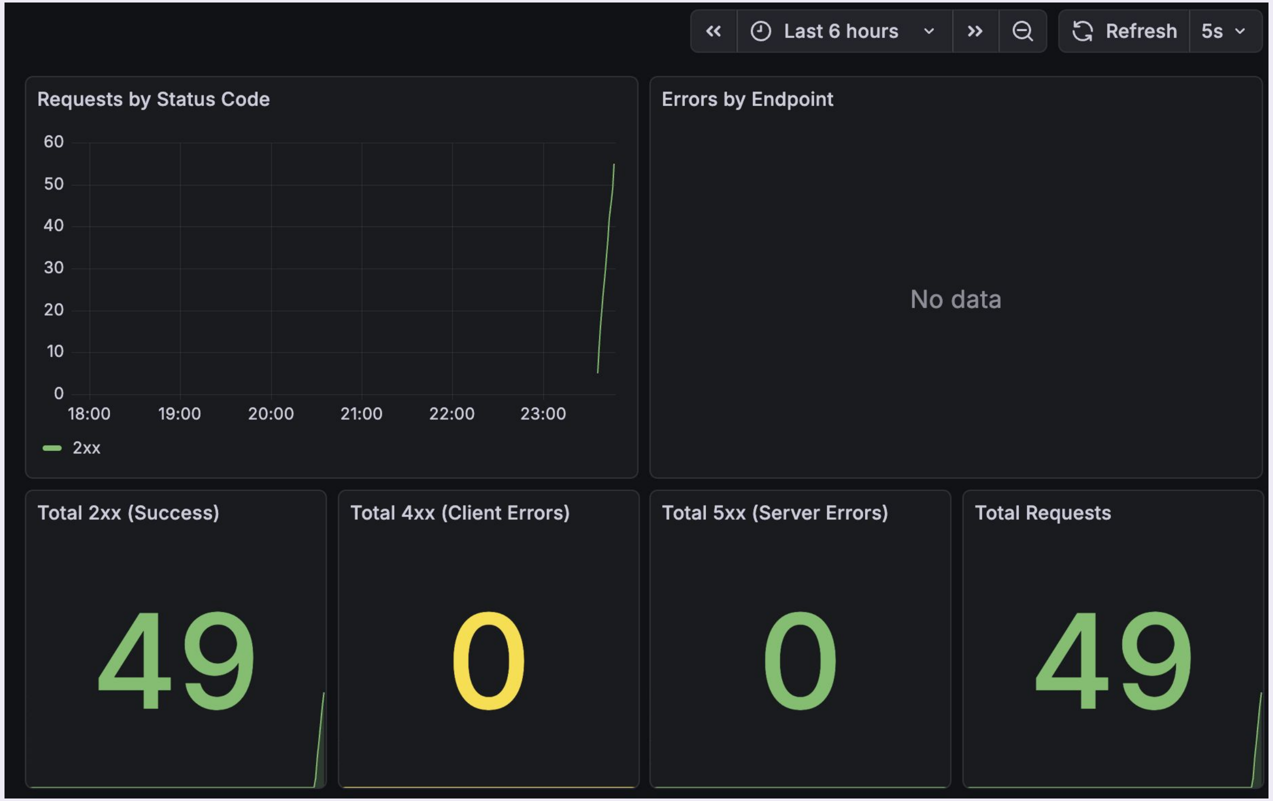
Avg Upload Time
299 ms

Total Presigned URL Calls
12

Total S3 Calls
15

Total Data Uploaded
249 KiB

4. Errors



Total 2xx (Success): 49

Total 4xx (Client Errors): 0

Total 5xx (Server Errors): 0

Total Requests: 49

7

App Demonstration

Thank You!