MANGO

Demand Estimator

FME Datathon 2025

The Problem

How much should **Mango produce** for the **upcoming season**?

If we produce too little → lost sales
If we produce too much → excess stock

Goal

Predict the full **demand** for each **product** **before** **production** begins

Initial Dataset

Dataset includes **4 seasons** (2 years) **+ validation season**

- Product family, category, color, price
- 95339 observations
- Image embedding (512-D vector)
- Phase-in, phase-out, life cycle
- Number of stores, number of sizes
- Weekly sales and weekly demand
- Historical production quantities

ID	id_season	image_embeddding	weekly_demand	Production
01	86	“0.467, 0.822, ..., -0.533"	69	4556
01	86	“0.467, 0.822, ..., -0.533"	112	4556
02	88	“0.345, -0.299, ..., 0.983"	90	12267
03	88	“-0.445, 0.869, ..., 0.351"	1130	137780

33 VARIABLES

Aggregating Weekly Demand

We **convert weekly-level rows** into **one row per product:**

ID	CATEGORY	WEEKLY_DEMAND
1	ABRIGO	70
1	ABRIGO	30
1	ABRIGO	50
2	PANTS	10
2	PANTS	5
2	PANTS	9

+

+

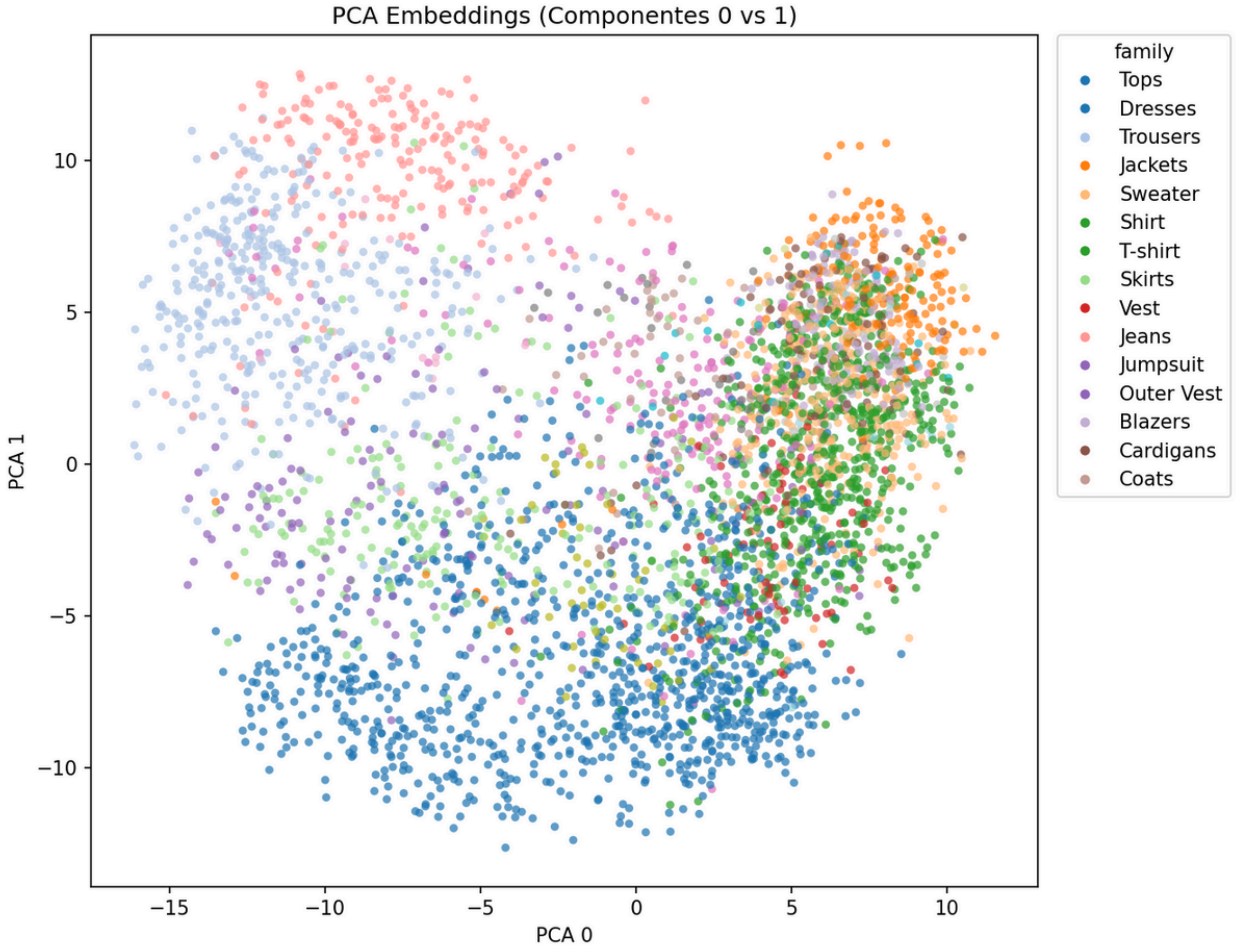
ID	CATEGORY	TOTALDEMAND
1	ABRIGO	150
2	PANTS	24

Dimensional Reduction

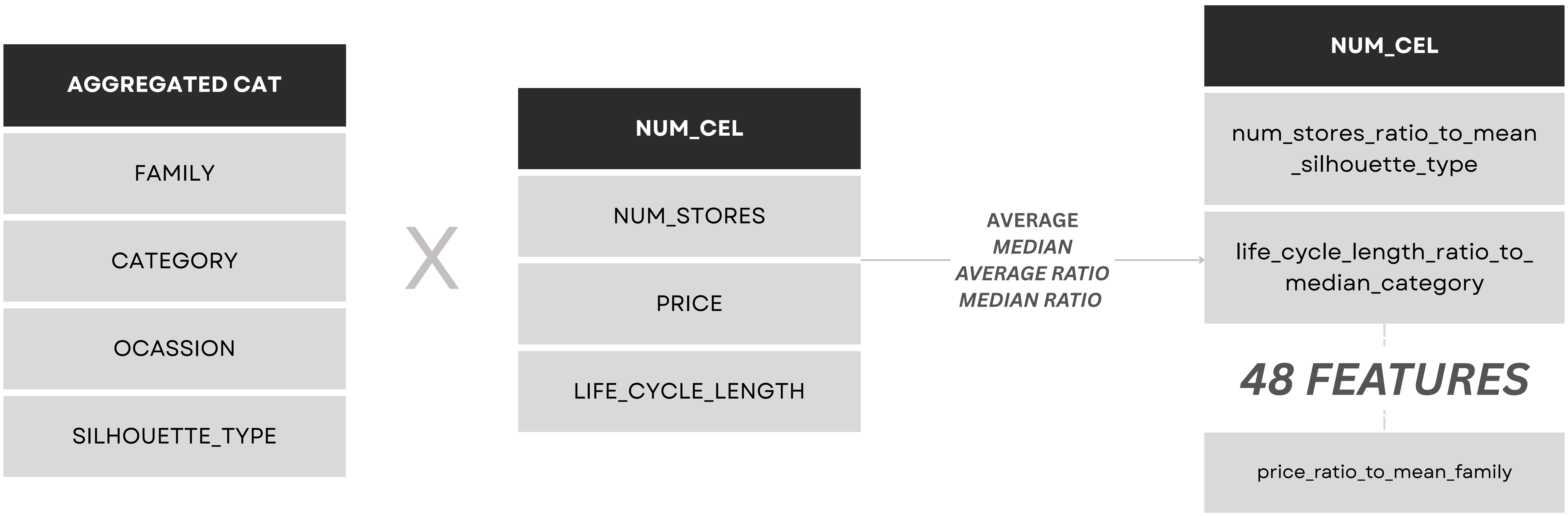
Image embeddings are **512-D.**

We apply **PCA** → **20 dimensions**
(94% variance retained)

ID	Category	pca_01	pca_02	pca_20
1	JACKET	[0.23, ..., 0.54]	[0.23, ..., 0.54]	[0.23, ..., 0.54]
2	JACKET	[0.23, ..., 0.54]	[0.23, ..., 0.54]	[0.23, ..., 0.54]



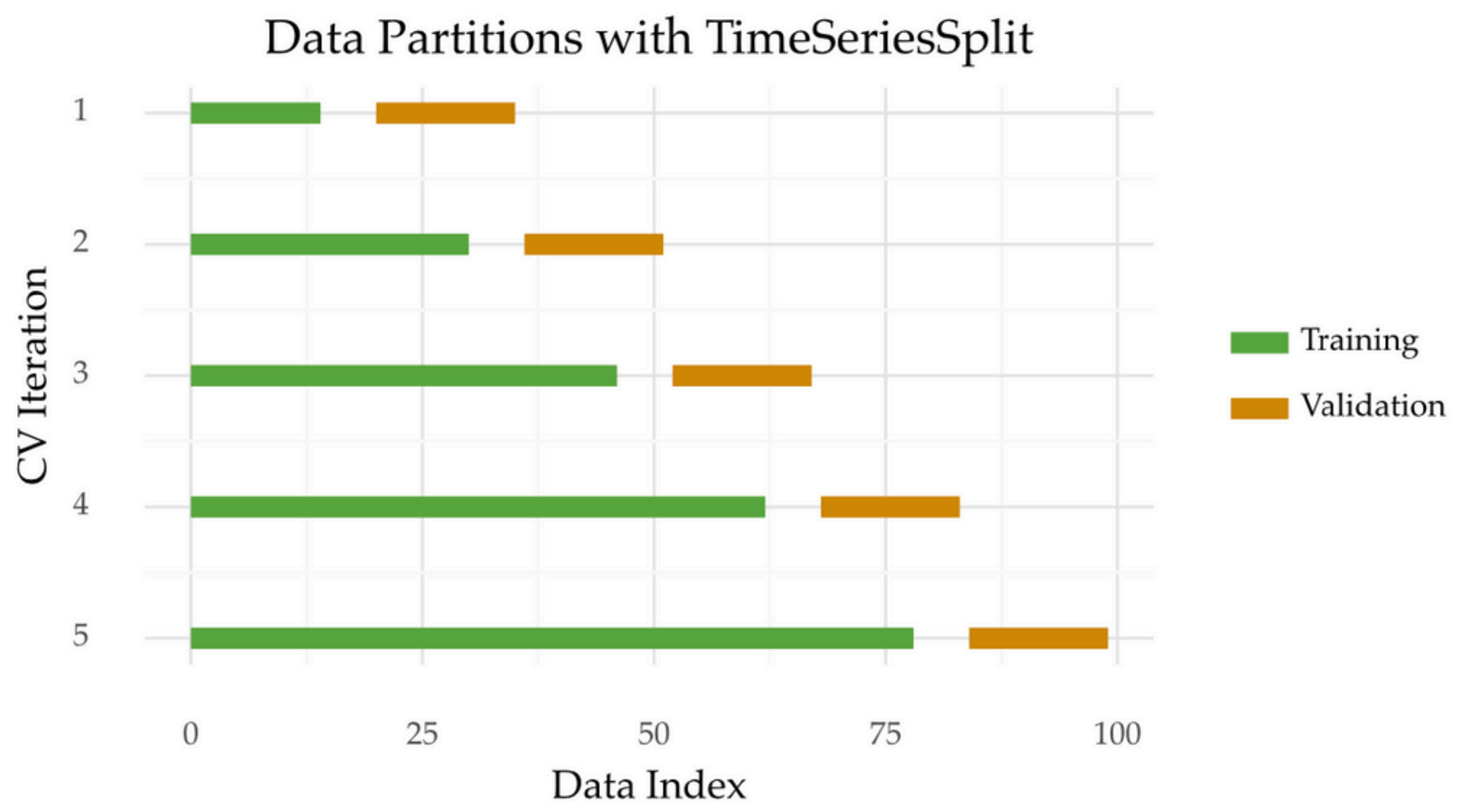
Feature Engineering

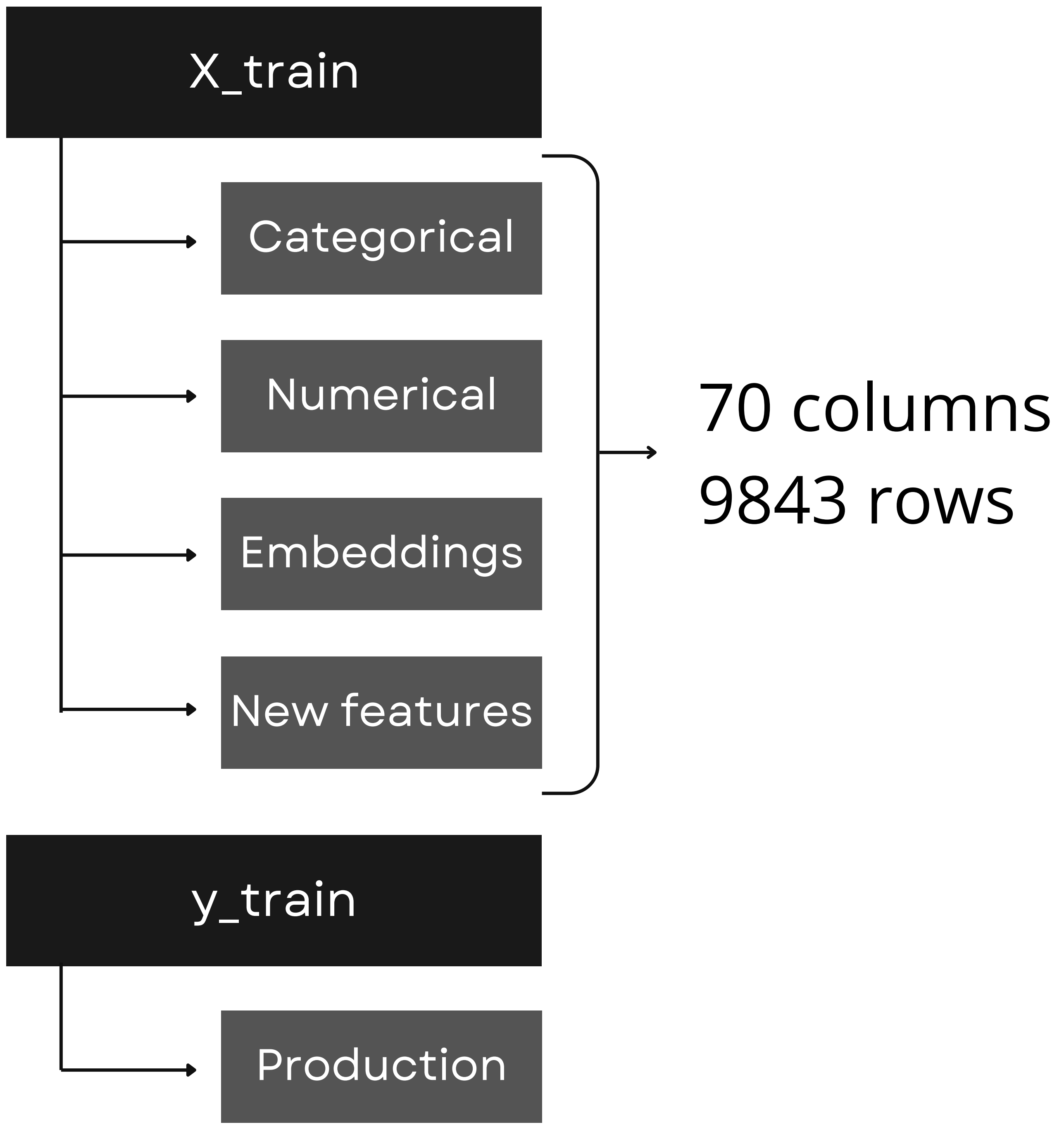


Data Partition for Training

We build:
X_train → all engineered features
y_train → aggregated total demand (sum of weekly_demand)

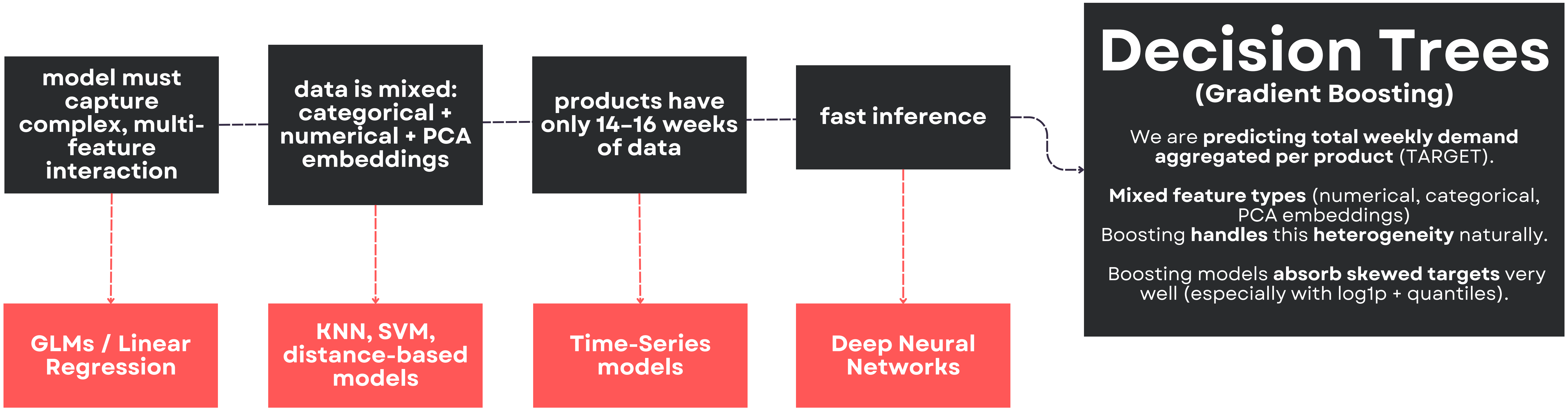
We order the dataset by season_id and apply a **temporal split**:
Train on **older seasons** → **validate** on the **latest season**.
Mimics real forecasting.





Model Selection

To choose the right model for this supervised regression, we start from the nature of Mango’s dataset and eliminate all model families that cannot handle its properties.



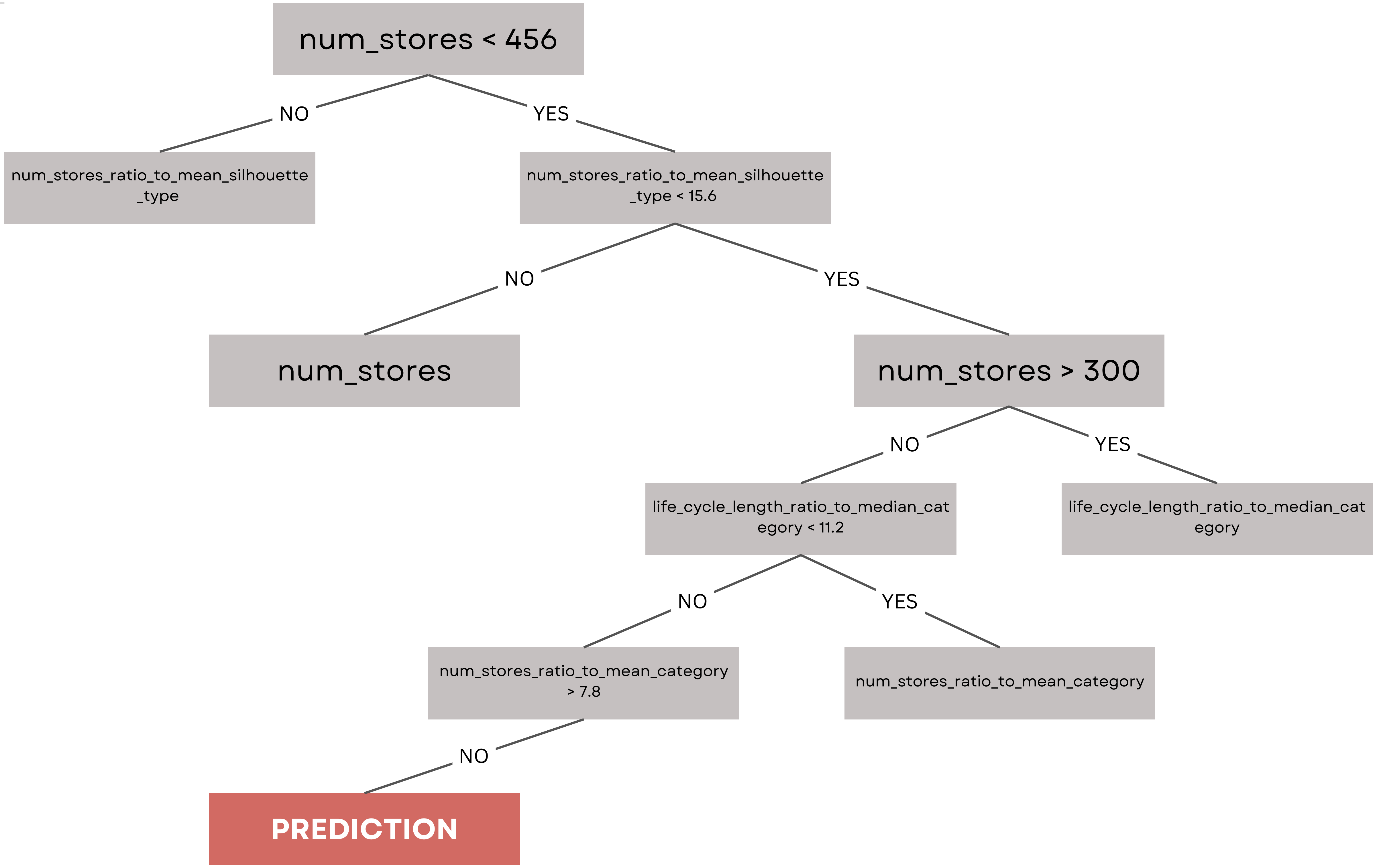
Tree boosting model selection

ASPECT	CATBOOST	LightGBM/XGBoost
Categorical handling	Native ordered enconding	Manual encoding
Noise and missing data	Robust and stable	Sensitive
Feature heterogeneity	Handles mix cleanly (catgs, numerics, embeddings...)	Needs more preprocessing

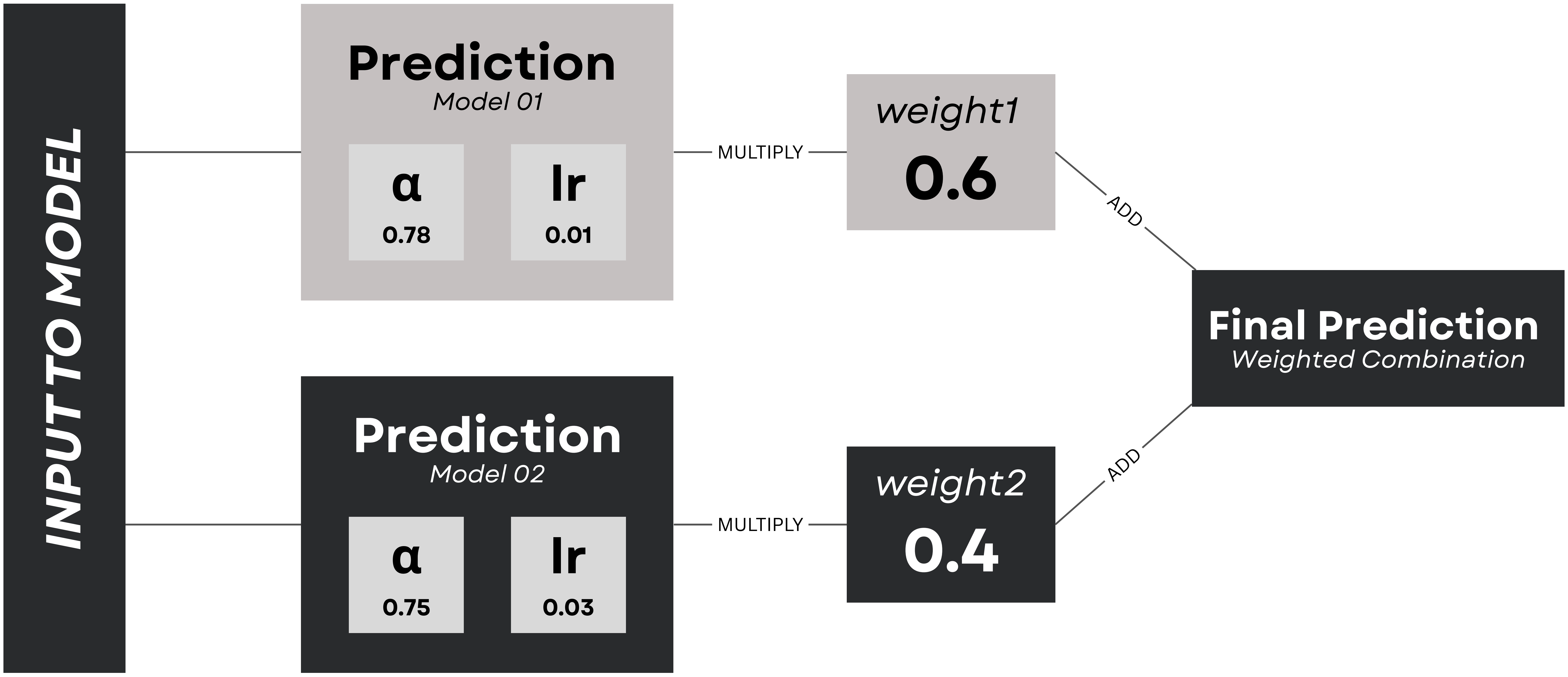
CatBoost Example

Each **split isolates different** product **behaviors** (stores, price ratios, life cycle…).

CatBoost naturally **captures non-linear interactions relevant** for **demand**.



CatBoost Model Concept



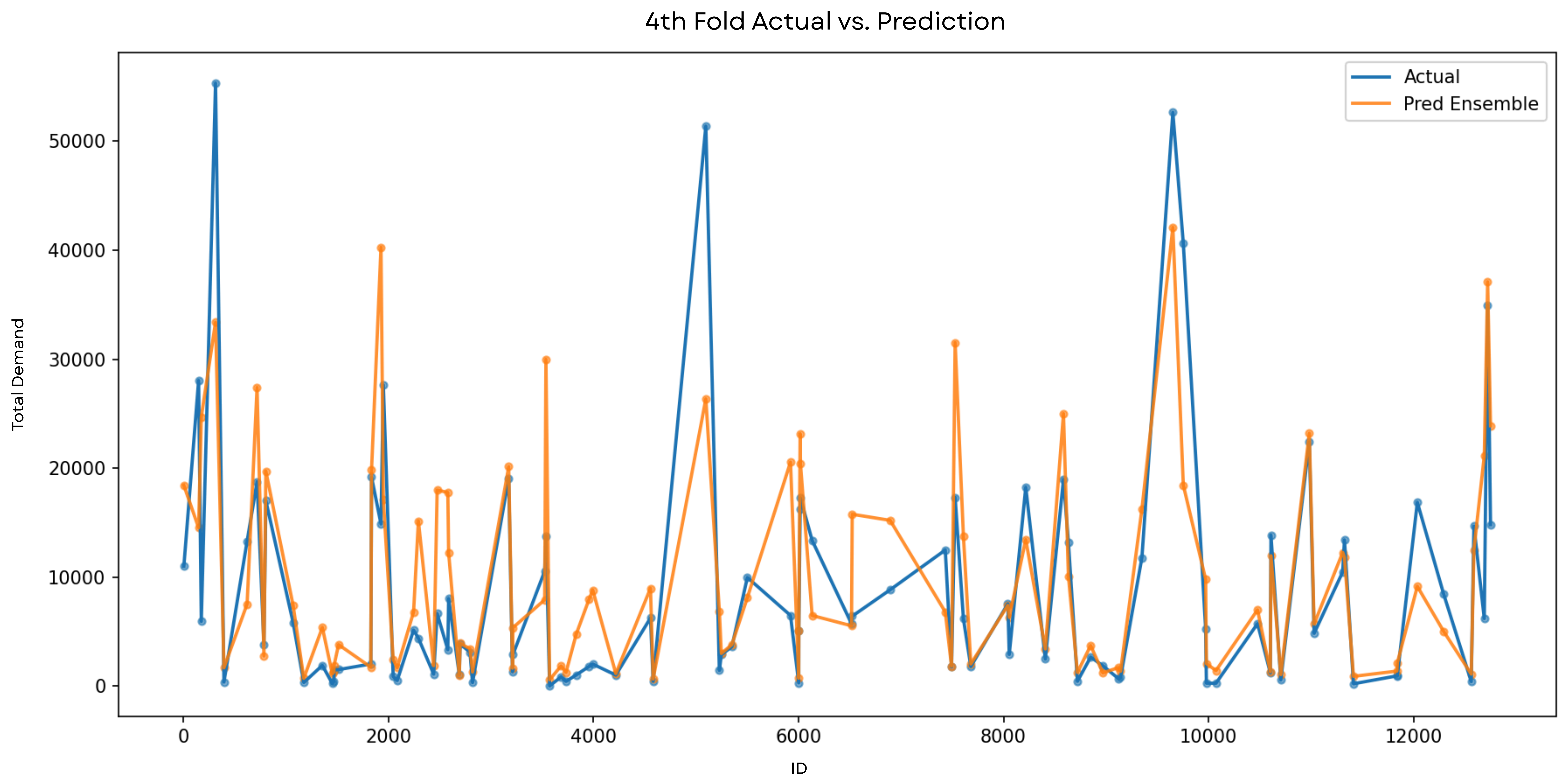
Kaggle Score

55.579

Public Score

59.243

Private Score


4th Fold Actual vs. Prediction
Actual
Pred Ensemble
Total Demand
ID
0
10000
20000
30000
40000
50000
0
2000
4000
6000
8000
10000
12000

Using the Model

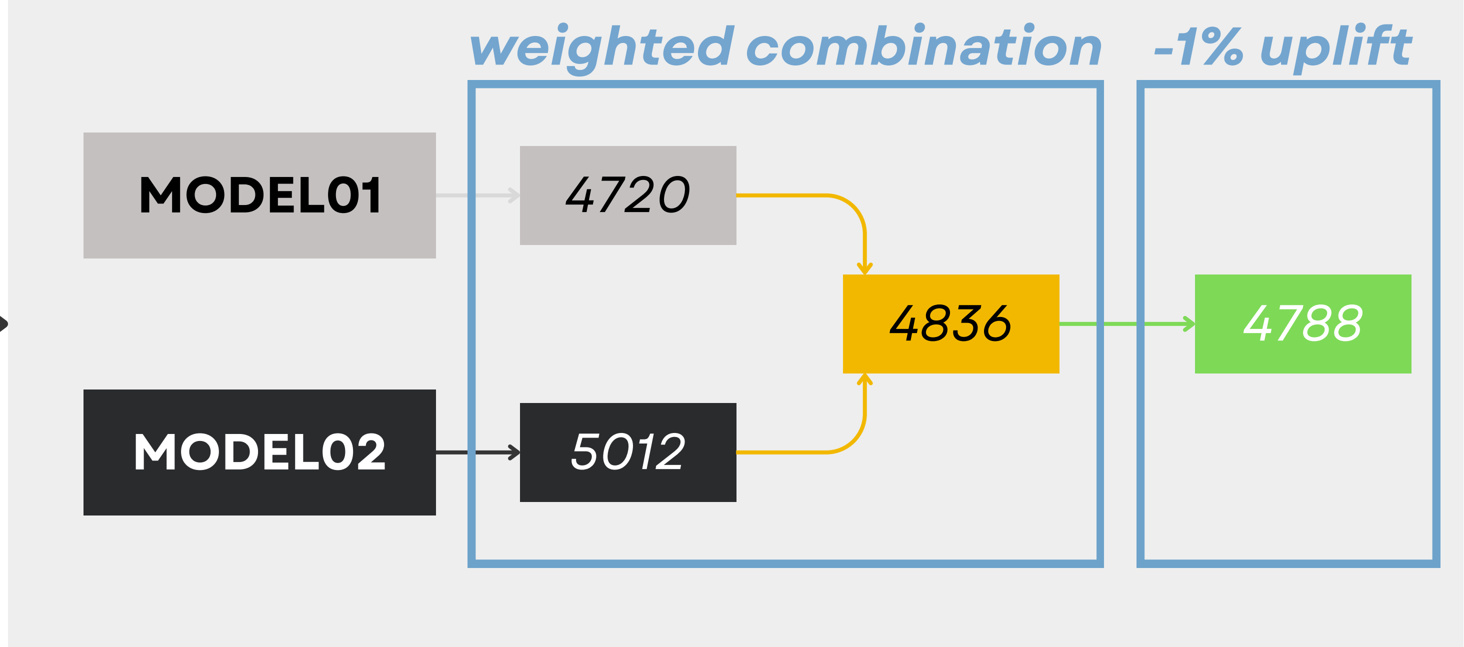
INPUT

PROCESSING

OUTPUT

Asymmetric Dress with Ruched Strap
Spring-Summer 2025

Family: Dresses
Category: Tops
Price: €45
Num_stores: 120
Num_sizes: 8
Life_cycle: 16 weeks
Season: 89 (Spring-Summer)
Image_embedding: [0.467, 0.822, ..., -0.533]
.....



4788 units

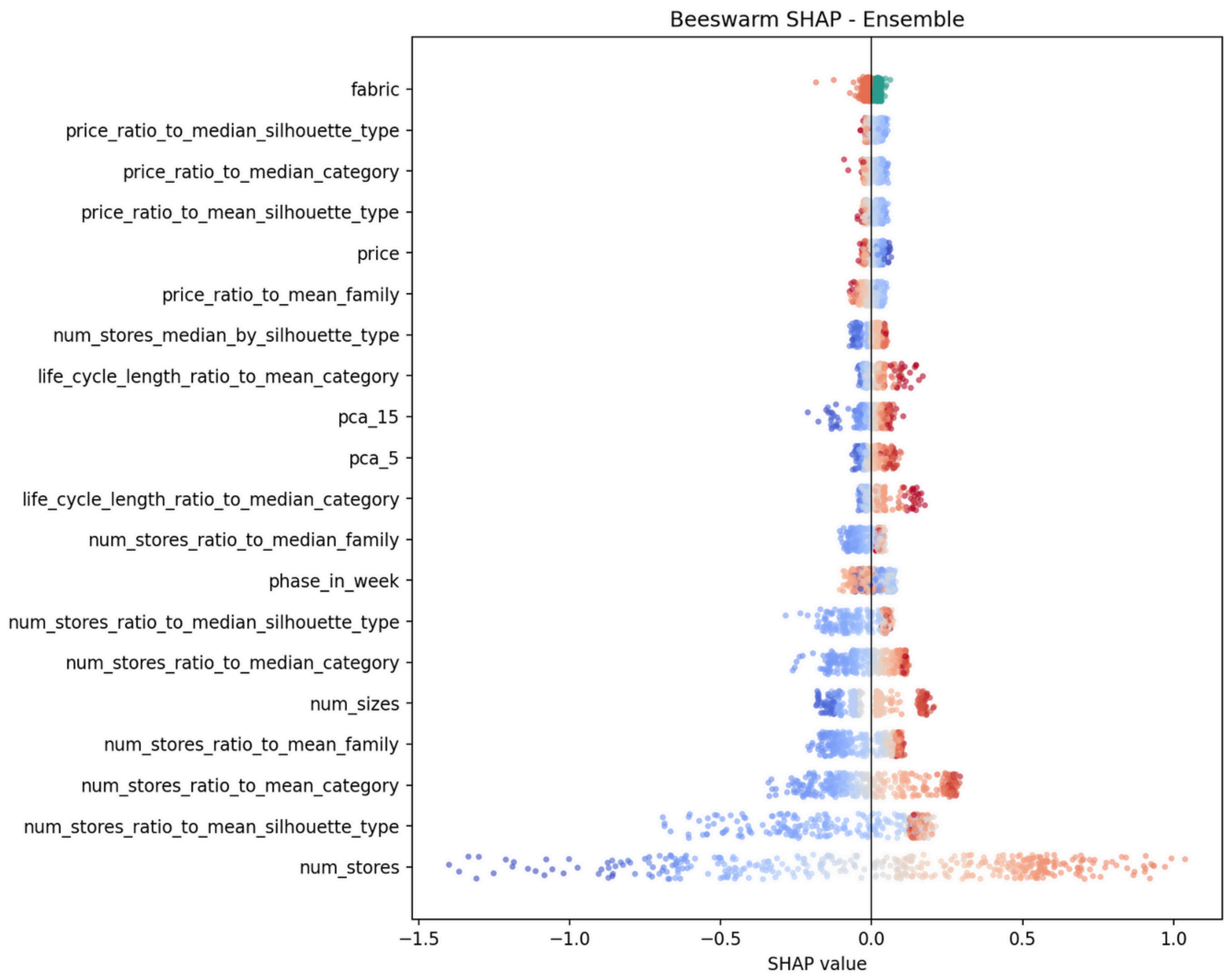
Explanability

Feature importance rankings show which variables drive predictions

Number of stores is the **dominant driver** of **demand**.

Longer planned **selling windows increase** expected **demand**; short life cycles reduce it.

Model uses **embeddings** to **capture style trends and similarities, improving predictions** on **cold-start products**.



Deployability

Our **solution** is designed to be **plug-and-play**. Once the model is **trained**, it is **exported** as a single **.cbm file** (CatBoost binary model).

- The .cbm file contains the entire model (trees, parameters, encodings).
- No external preprocessing pipeline is required.
- No need to rebuild encoders or transformers at inference time.

```
from FMEMango import PredictDemand

# Input: JSON with product characteristics
product = {
    "family": "Dresses",
    "num_stores": 120,
    "price": 45.0,
    "life_cycle_length": 16,
    "num_sizes": 8,
    "id_season": 89,
    "image_embedding": [0.467, 0.822, ..., -0.533]
}

# Predict
demand = model.predict(product)

>>> 4856
```

Future Work

Real-time Demand Updates

Implement online learning to adjust predictions based on early-season sales performance

Deep Learning for Images

Train custom vision models on product images to extract fashion-specific features beyond generic embeddings

Regional Segmentation

Build location-specific models accounting for regional preferences and climate differences

Questions?

alex.latorre@estudianat.upc.edu
https://www.linkedin.com/in/latorrealex/

robert.ennakhli@estudiantat.upc.edu
https://www.linkedin.com/in/robert-ostrozhinskiy/

nicolas.rosales@estudiantat.upc.edu
https://www.linkedin.com/in/nicolas-rosales-gomez/

fabian.birch@estudiantat.upc.edu
https://www.linkedin.com/in/fabian-birch/

https://github.com/frozono9/FME2025

MANGO

Demand Estimator

FME Datathon 2025