SKU Demand Forecasting (Aritzia Style)

1. Project Goal

Original assumption: Forecasting would identify which products are "successful."

Corrected goal: Reduce uncertainty in inventory planning for already high-demand SKUs by forecasting *how much* and *when* demand will occur.

This reframing aligned the project with real retail practice (forecasting supports operations, not product discovery).

2. Dataset Choice & Constraints

Dataset: Kaggle – *Store Item Demand Forecasting* (train.csv)

Why this dataset works:

- Daily, multi-year sales history
- Store × item structure → realistic retail setup
- Clean time series suitable for forecasting

Key limitation (accepted, not ignored):

- Dataset is synthetic and homogeneous
- All items are already "successful"
  → Not suitable for "winner vs loser" analysis, but appropriate for demand forecasting

3. Scope Reduction

Initial reality:

- 10 stores × 50 items = **500 time series**
- ~913k rows, ~5 years per series

Decision: Pilot on **1 store + 5 SKUs**

Why:

- 500 Prophet models = slow, noisy, hard to interpret
- Industry practice is to pilot before scaling

SKU selection method:

Chose top-variance SKUs within one store to maximize signal:

```
df[df["store"] == 1].groupby("item")["sales"].std().sort_values(ascending=False)
```

Mapped to real Aritzia-style products for storytelling:

Super Puff Shorty, Effortless Pant, Sculpt Knit Tank, Wilfred Sweater, TNA Sweatfleece Hoodie

4. Key ETL steps:

- Filter to pilot scope
- Rename SKUs
- Fill missing dates with 0
- Sort by SKU and date

5. EDA Findings

What EDA confirmed:

- Strong, stable annual seasonality
- Similar demand shapes across SKUs
- No structural breaks or anomalies
- High daily noise → aggregation needed

What EDA did *not* show:

- No dramatic SKU-specific seasonal shifts
  → Dataset is calendar-driven by design

Important realization:
EDA's job here was to confirm forecastability, not discover "winning" products.

6. Model Choice: Prophet

Why Prophet:

- Data shows smooth trend + strong seasonality
- Goal = interpretability + risk assessment
- Not optimizing RMSE, but supporting planning decisions

Explicit decision not to use XGBoost:

- No additional features (price, promos, holidays)
- Would increase complexity without changing conclusions

7. Forecasting Logic

Forecast horizon: 90 days
Key question: Will future demand exceed what we already handled last year?
Benchmark definition:

```
last_year_peak = last_year_df["y"].max()
```

= max units sold on *any single day* last year
= worst historical demand stress

Comparison:

- Forecasted peak (expected)
- Forecasted peak (upper bound) vs. Last year's peak

8. Risk Interpretation

For all 5 SKUs:

- Forecasted peak < last year’s peak
- Upper bound < last year’s peak

Conclusion: Low short-term risk of exceeding historical capacity.

Operational meaning:

- No need to escalate inventory
- Existing planning assumptions likely sufficient
- Forecasting prevented unnecessary overreaction

	SKU	Last Year Peak	Forecast Peak (Expected)	Forecast Peak (Upper)	Risk Level
0	Effortless Pant	155	106.3	118.8	Low
1	Sculpt Knit Tank	155	100.5	113.4	Low
2	Super Puff Shorty	154	105.8	117.7	Low
3	TNA Sweatfleece Hoodie	150	98.1	109.6	Low
4	Wilfred Sweater	139	99.4	111.2	Low