Pricing Analytics for Fashion Retail

My experience after 4-ish years in the trenches

Andrea Taverna, PhD CAP

Currently Decision Scientist in private sector

andrea.taverna@outlook.com

in https://www.linkedin.com/in/andrea-taverna-data-analytics

Disclaimer

Opinions stated here are my own and do not necessarily reflect the official policy or position of my past, present or future employers.

About me

- worked on novel Pricing Optimization for Zalando between 2019-2021, see paper
- between 2021-2022 MLE at Priceloop GmbH
- main expertise: large-scale decision optimization

Topics

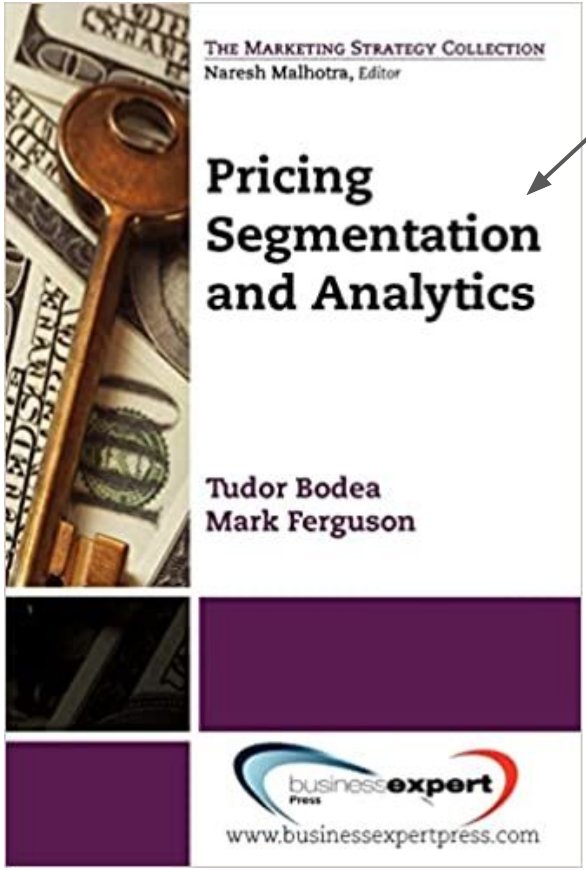
1. **Analytics Processes**

2. **Predictive Analytics,** with Data Science and Machine Learning

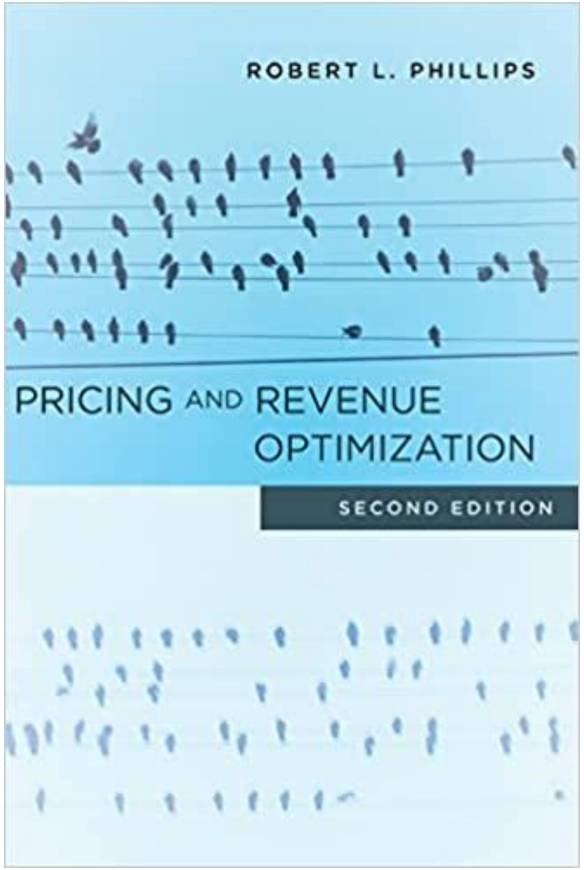
3. **Prescriptive Analytics,** with Decision Optimization

4. **Tips for guaranteed impact**

Recommended starting sources

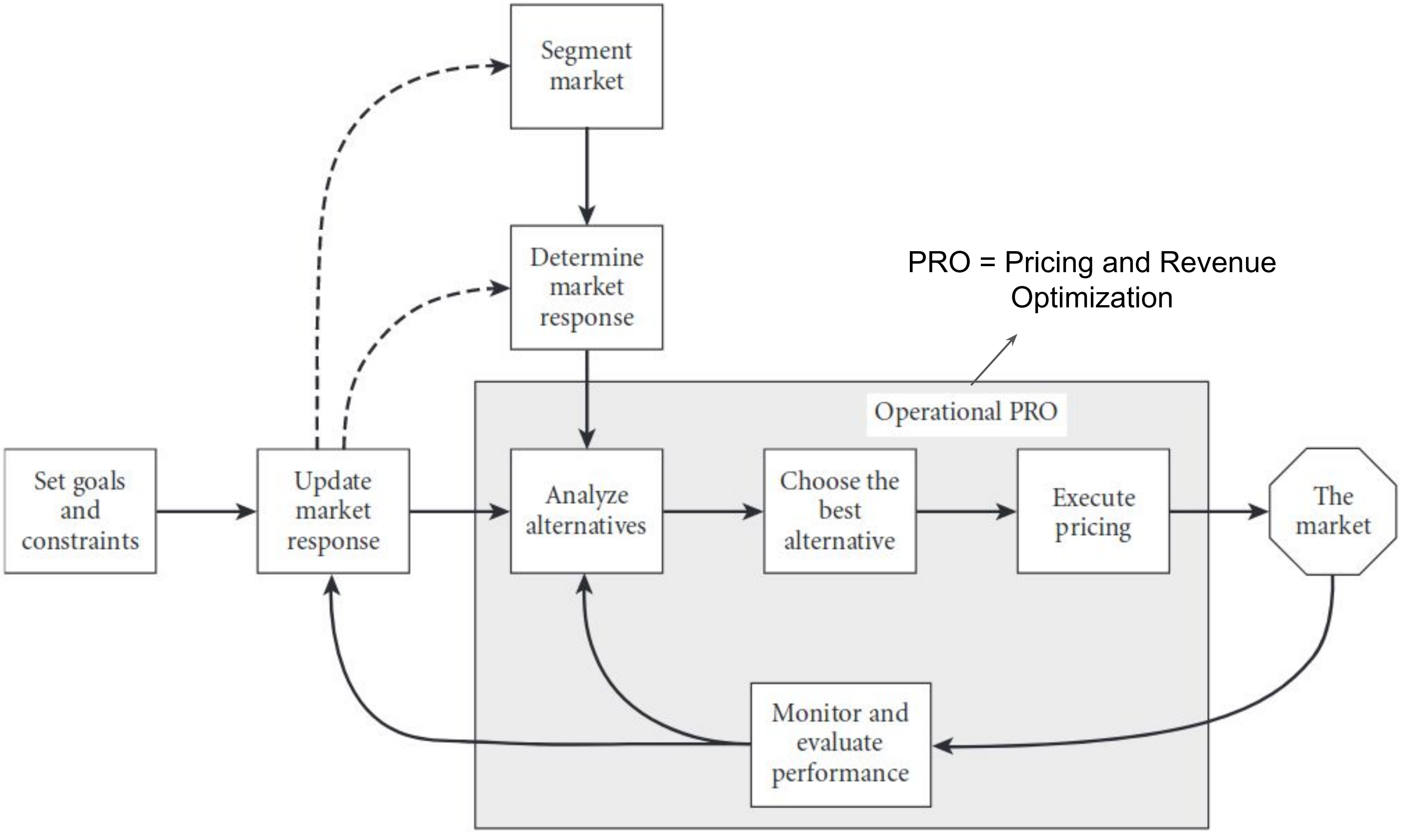


my favourite (pocket, cheap)



Analytics Processes

Pricing Analytics, the ideal version

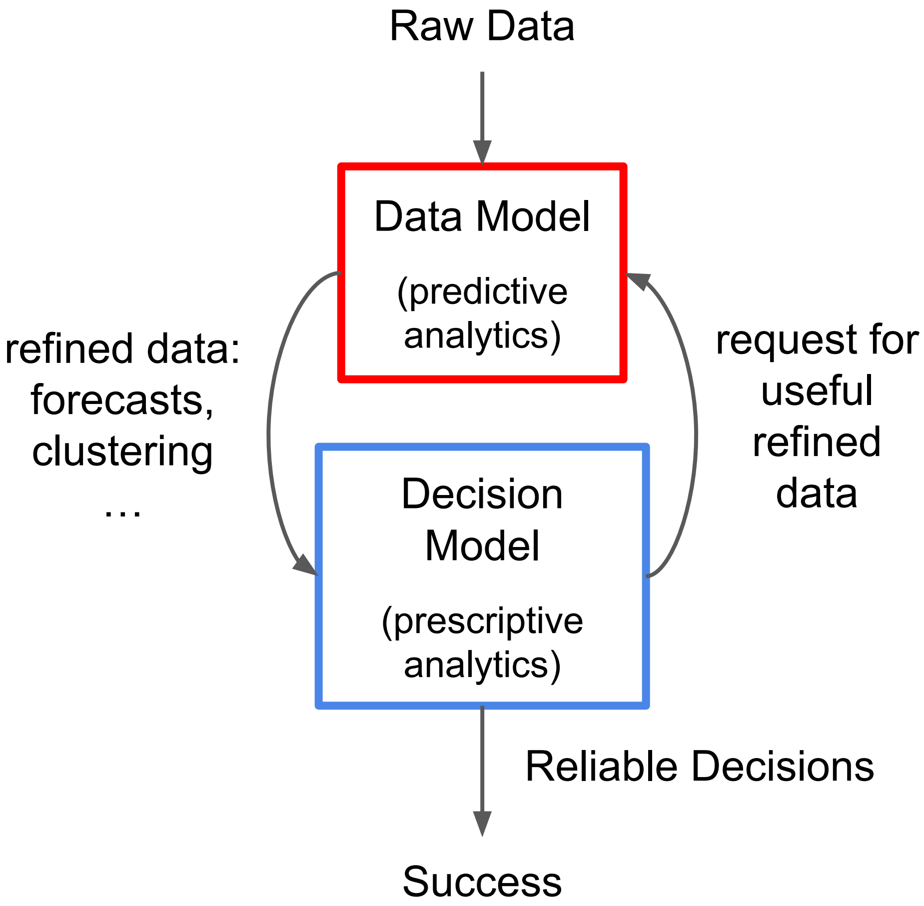


source

Analytics Processes

Analytics processes made of two sides:

- **data/predictive:** extract information from raw data
  - e.g. demand forecast
- **decision/prescriptive:** recommend decisions for users to execute
  - e.g. pricing optimization, inventory management, …
- **they work in synergy!**



Pricing Analytics should be …

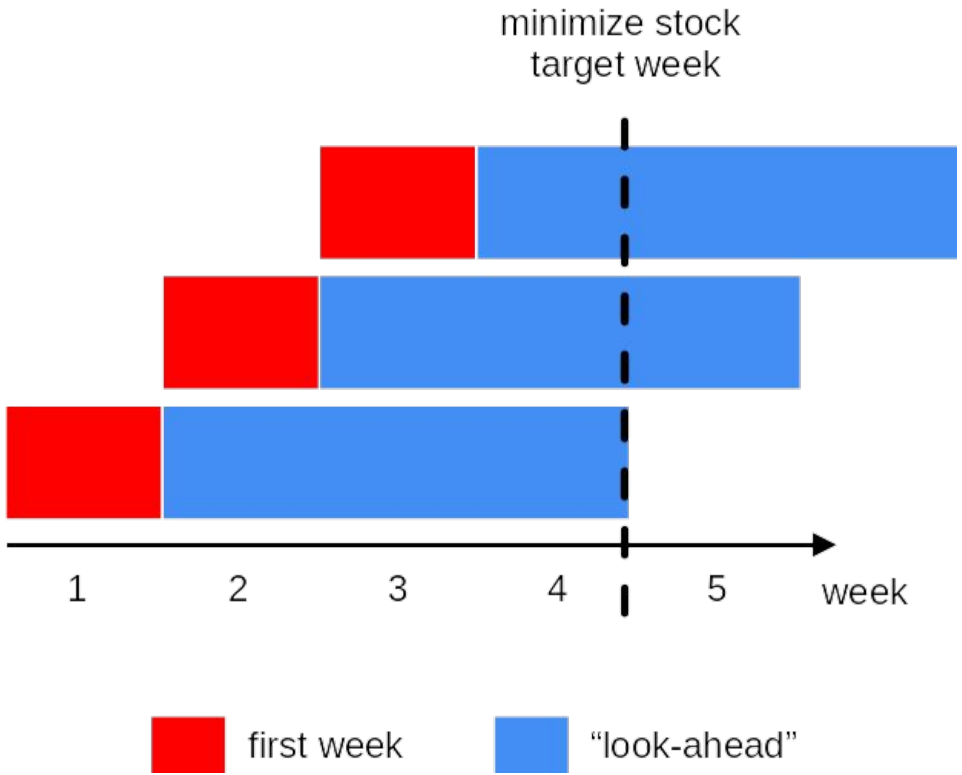
- **dynamic**, to react to changes via price adjustments
- **holistic**, integrated with other functions:
  - marketing
  - logistics (inventory, stock flows…)
  - finance
  - …
- **structured** and **consistent** over different dimensions:
  - time horizons (daily vs monthly vs yearly)
  - channels, possibly competing with each other
  - …

Example: monthly planning with weekly periods (link)

Pricing Manager wants to plan prices for the next week (week 1) such that:

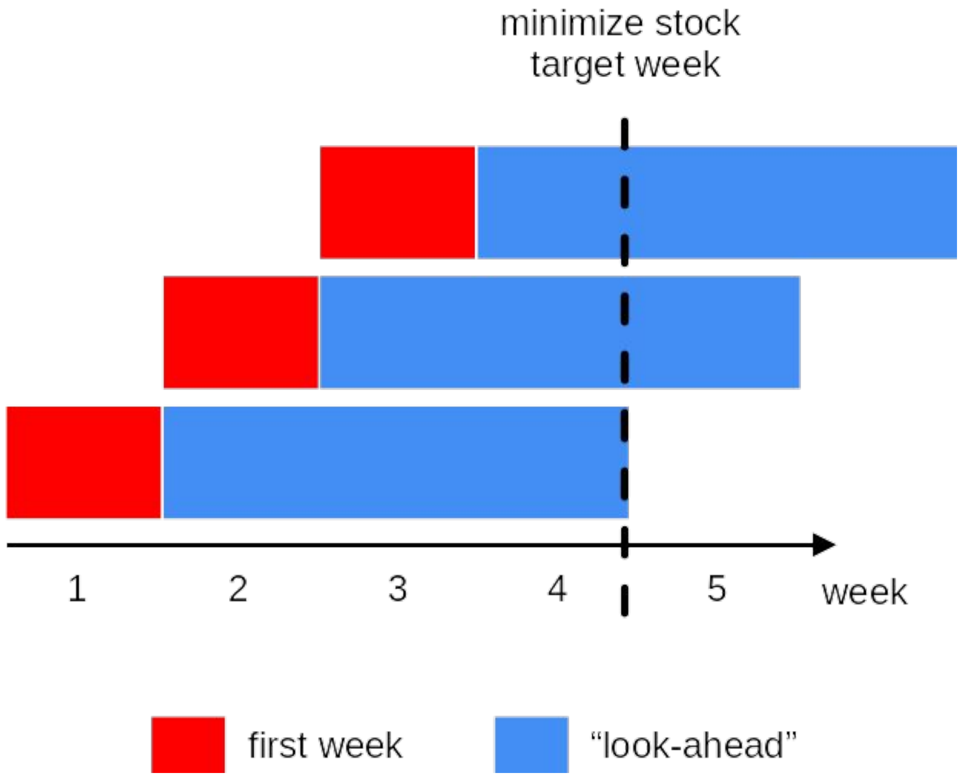
- the profit over the next 4 weeks is maximized
- the left-over stock at the end of the month (week 4) is minimized

The process is repeated again on week 2 and following, by rolling forward the 4 weeks horizon.



Example: monthly planning with weekly periods (link)

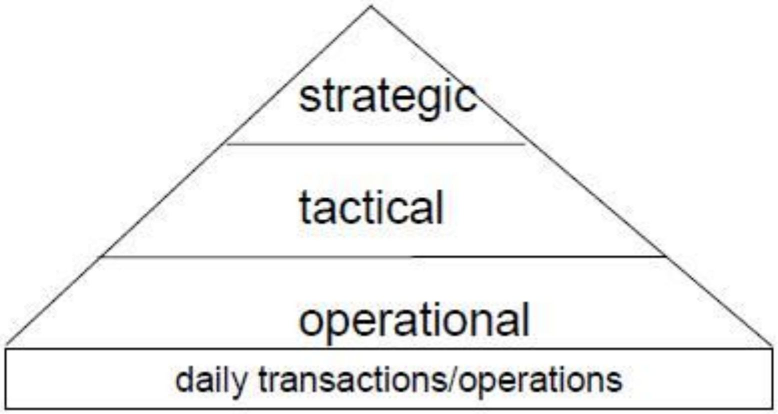
- **dynamic:** continuous planning on rolling horizon
- **holistic:** consider profit and stock
- **consistent:** plan prices for week 1 while considering the next 3 weeks as well ("look-ahead" period)
- **structured:** repeatable and nestable
  - e.g. quarterly planning with monthly periods
  - weekly planning with daily periods



Hierarchical planning (aka Anthony’s triangle)

Break down decision making in levels

1 level ⇒ 1 problem/ data product



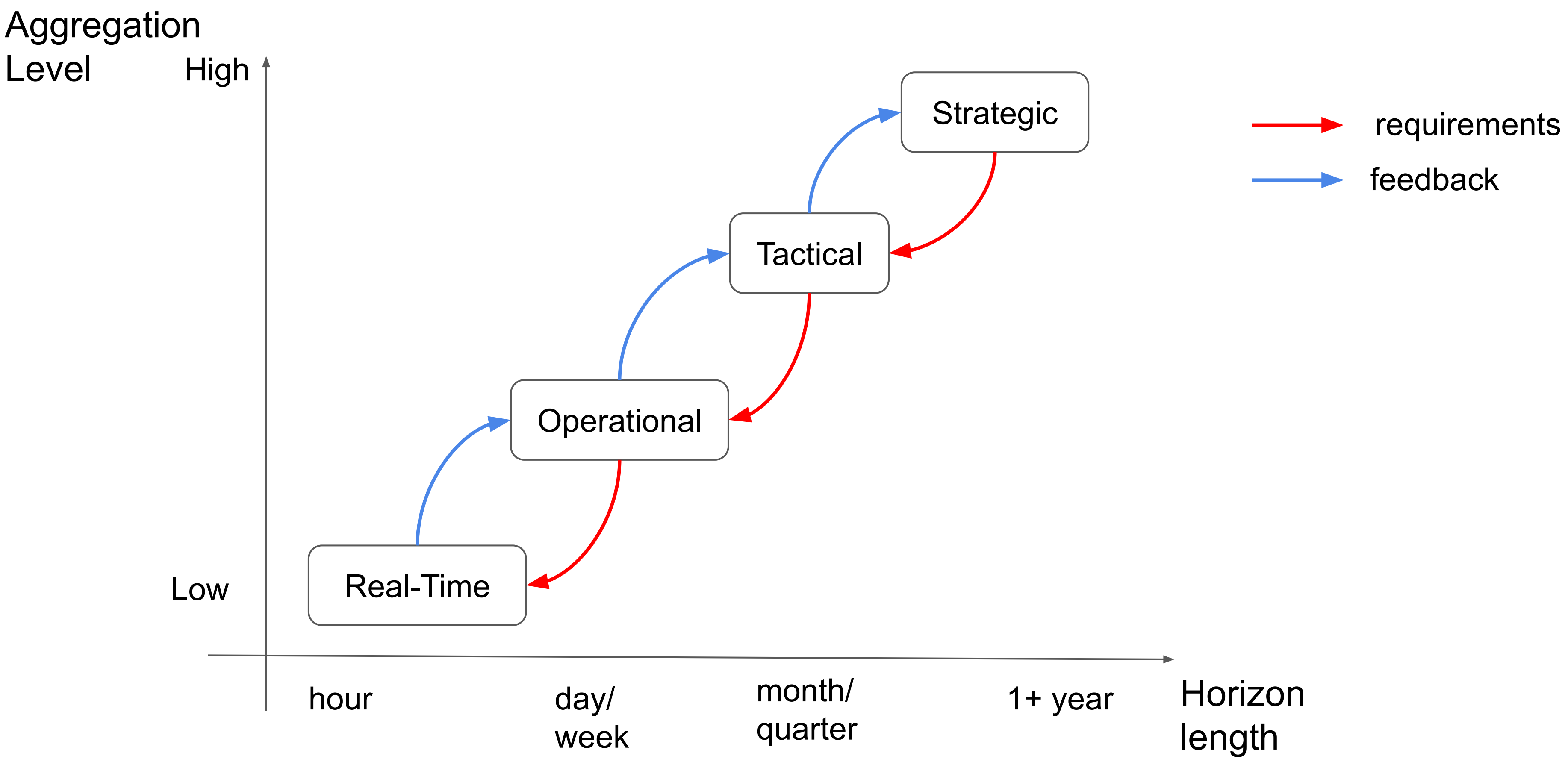
Hierarchical planning (aka Anthony's triangle)

Break down decision making in levels

1 level ⇒ 1 problem/ data product

Characteristics	*Strategic planning*	*Tactical planning*	*Operations control*
Objective	Resource acquisition	Resource utilization	Execution
Time horizon	Long	Middle	Short
Level of management involvement	Top	Medium	Low
Scope	Broad	Medium	Narrow
Source of information	(External	& Internal)	Internal
Level of detail of information	Highly aggregate	Moderately aggregate	Low
Degree of uncertainty	High	Moderate	Low
Degree of risk	High	Moderate	Low

Hierarchical planning (aka Anthony's triangle)



Programmed vs Non-programmed Decisions (Simon's)

Programmed decisions:

- taken routinely
- from a structured business process
- data should be available
- comprehensive solution can be built

Non-programmed decisions:

- specific, taken occasionally
- no corresponding business process
- data may not be fully available
- solution limited to be ad-hoc/heuristic

⇒ One model for each case!

Examples

- routine pricing planning
  - programmed
  - hierarchical: consider different horizons (week/month/quarter/year) and dimensions (category/market/channel…)

- new marketing campaign
  - probably not programmed
  - ad-hoc solution (to begin with)

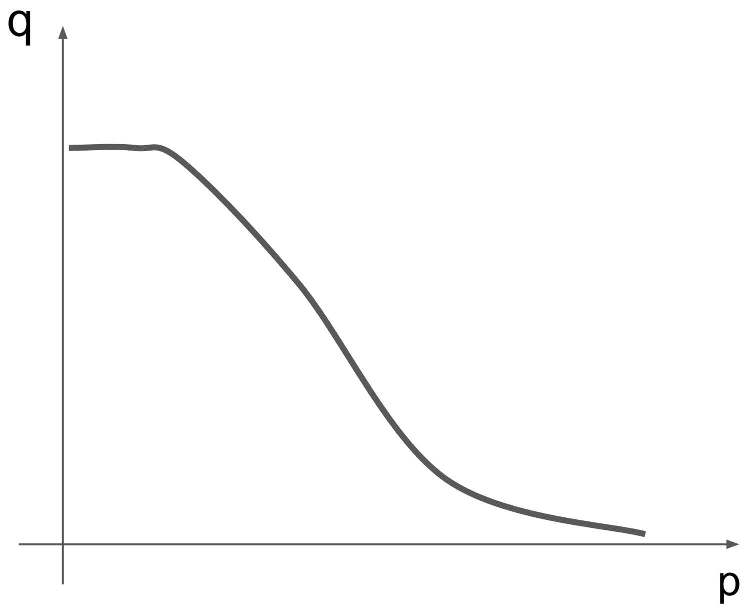
Predictive Analytics

Learning demand response

Demand response functions and elasticity

Demand q function of price p

$$q(p) \sim D(p|\theta)$$



Standard Demand Functions from Economics

- linear
- constant elasticity (exponential)
- logit
- …

When considering competing products:

- multinomial
- …

Learning Demand with Regression

Main idea: log-transform demand

- avoid negative values
- demand ~ price relation is often "exponential-like"
- caveat: can introduce bias

Better alternatives: count distributions, e.g.

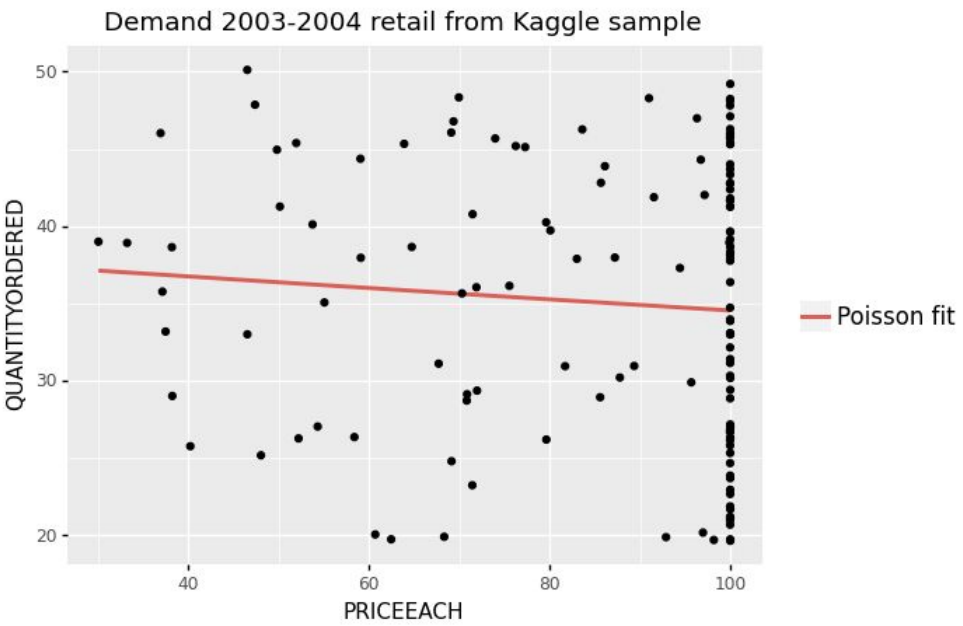
- Poisson Regression, if mean ~ variance
- Negative Binomial Regression, otherwise

available on statsmodels' GLM module

ML for Pricing can be hard: poor historical data

Historical data can be insufficient for learning.

Pricing Managers don't risk trying different prices ⇒ few price changes to learn from.

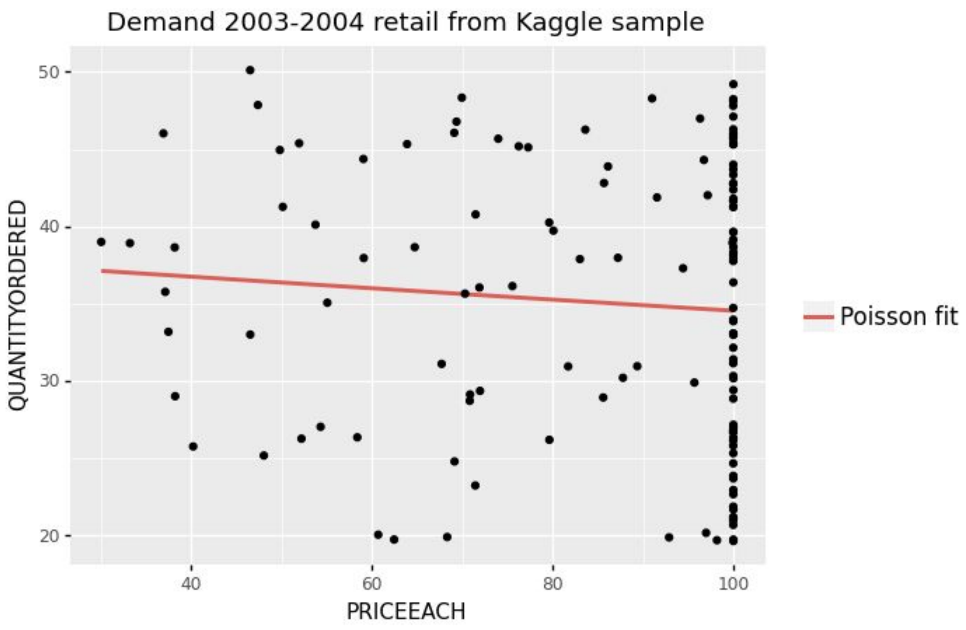


source

ML for Pricing can be hard: confounders

Lots of confounding factors:

- market conditions
- competitors’ behaviour
- auto-regressive confounders
  - past demand
  - past prices
  - past inventory
- pricing managers’ intervention
  - increase price → stock increase → sales increase



source

Quest for causality

Standard ML gives us a **probabilistic model** that tells **the demand most correlated to the price p**

$$D(p|\theta)$$

For pricing we want a **causal model** that tells **the demand if price is set to p:**

$$D(\mathrm{do}(p)|\theta)$$

(see do calculus [here])

Quest for counterfactuals

Both standard and causal ML need "counterfactuals", i.e. sales samples:

- for same item, channel, time, customer segment ...
- which differ only by price

very hard to get in practice:

- time constraints
- business constraints
- legal constraints
- …

What can we do?

Just Use ML 🤷?

Non-causal ML models can still:

- provide useful predictions even if not explicitly causal (google example)
- include some causal assumptions
- be used in synergy with causal modelling

Include causal assumptions in ML models

Examples:

- XGBoosts allows monotonicity constraints
  - prevent learning unexpected “high price ⇒ high demand” relation
- DeepAR allows to specify a Poisson or Negative Binomial output distribution

Synergy between probabilistic and causal ML
(residualization)

Given sales s_t, price p_t, confounders c_t and non-confounding covariates k_t:

1. train non-causal ML model with confounders $\tilde{s}_t = g(\mathbf{s}_{t'<t}; c_t|\sigma)$
2. compute residuals $v_t = s_t - \tilde{s}_t$
   – residuals are left without confounding factors
3. train a causal model on the residuals $\tilde{v}_t = h(p_t; k_t|\theta)$
4. obtain price-reactive forecast $\hat{s}_t = \tilde{s}_t + \tilde{v}_t$

Synergy between probabilistic and causal ML
(residualization)/ Examples

- BlueYonder's Felix Wick talk
- Thompson Sampling for Dynamic Pricing, pg. 5-6

Price Experimentation & Reinforcement Learning

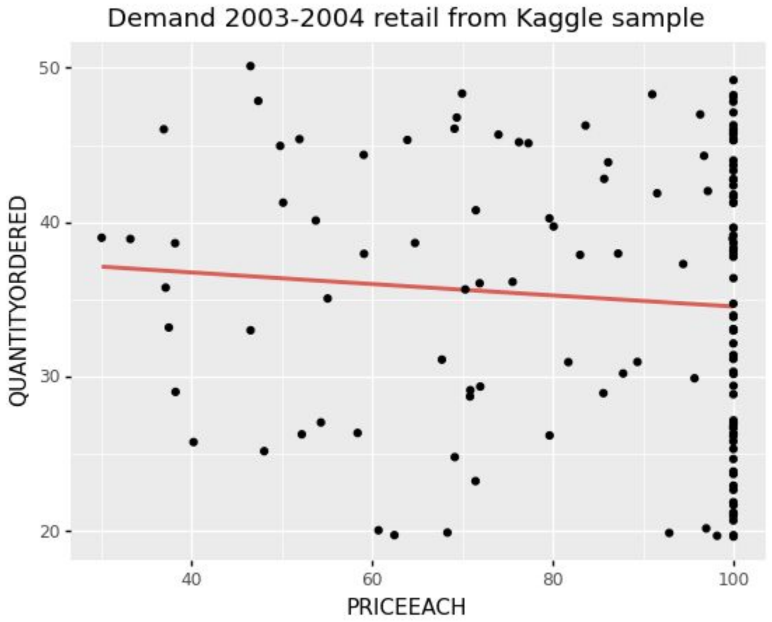
Price Experimentation to learn Demand Response

Main idea:

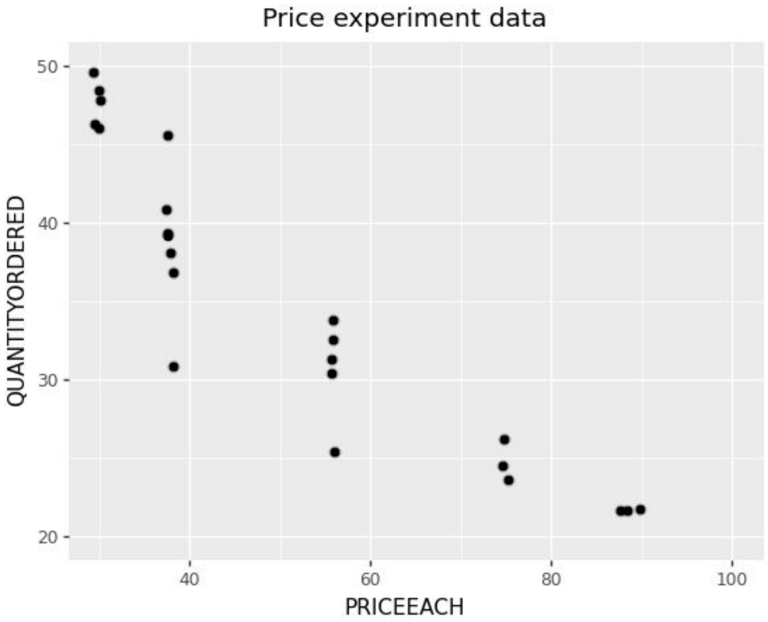
- try different prices for each product
- to acquire data for Pricing Analytics, i.e. generate counterfactuals
- in particular, to learn demand response

Price Experimentation: Example

Past data, hard to learn from



Ideal **experimental data,**
easier to learn from



Pricing Experimentation: issues /1

Confounders:

- equivalent products are priced differently, the cheaper ones will sell more
- equivalent products with same price can be influenced by visibility in shop/website or marketing campaigns
- …

Pricing Experimentation: issues /2

Fairness:

- customers in a shop/on a website should pay the same price for the same article
  - “price discrimination” is often forbidden by law
- a customer should not see the price decrease soon after her purchase
- …

Price Experimentation: possible solutions

- create reasonable control groups: equivalent products in the same group
- coordinate well with other functions (marketing, recommender systems for websites…)
- switchback testing: do not control for customers/products but for regions/time
- …

Price Experimentation with Reinforcement Learning

Price Experimentation can be seen as a Multi-Armed Bandit Problem:

- have multiple prices (arms) to try
- need to learn expected demand for each price
- want to try different prices (exploration) without sacrificing profitability (exploitation/regret minimization)

Price Experimentation with Thompson Sampling

- popular in pricing
  - see [1], [2], [3]
  - see [4] for alternatives
- Bayesian Reinforcement Learning method
  - learn the distribution *p* of demand parameters θ while maximizing reward *r*

for $t = 1, 2, \ldots$ **do**

  #sample model:

  Sample $\hat{\theta} \sim p$

  #select and apply action:

  $x_t \leftarrow \text{argmax}_{x \in \mathcal{X}} \mathbb{E}_{q_{\hat{\theta}}}[r(y_t) | x_t = x]$

  Apply x_t and observe y_t

  #update distribution:

  $p \leftarrow \mathbb{P}_{p,q}(\theta \in \cdot | x_t, y_t)$

end for

source

Price Learning with Thompson Sampling

Example: fashion flash sales

- brand new assortment goes on sale for limited amount of time
  - e.g. limited stock available
- no data for it, hard to compare with previous sales
- you want to choose price "reasonably"

for $t = 1, 2, \dots$ **do**

  #sample model:

  Sample $\hat{\theta} \sim p$

  #select and apply action:

  $x_t \leftarrow \mathrm{argmax}_{x \in \mathcal{X}} \mathbb{E}_{q_{\hat{\theta}}}[r(y_t) | x_t = x]$

  Apply x_t and observe y_t

  #update distribution:

  $p \leftarrow \mathbb{P}_{p,q}(\theta \in \cdot | x_t, y_t)$

end for

source

AB testing pricing policies

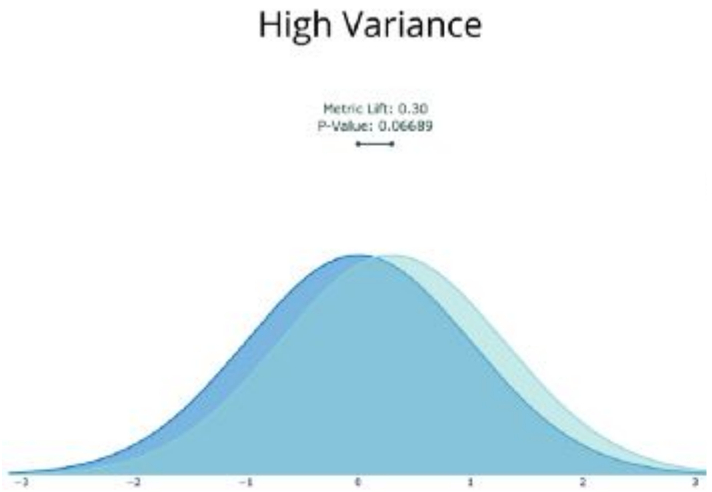
AB testing pricing policies

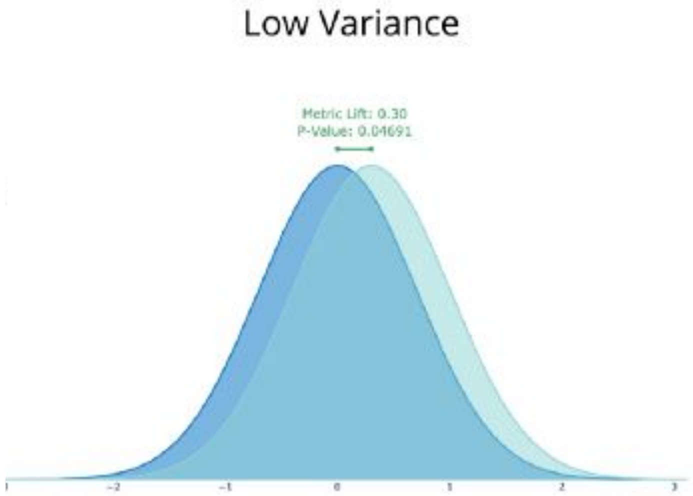
Like price experimentation: try different prices for similar products

- same issue of price fairness and controlling confounders
- different context: we have two pricing policies “Control” and “Test” and want to decide whether one is better than the other

Dealing with variance

- AB testing in practice can be very noisy and show high variance
- there is too much “stuff” going on:
  - marketing
  - customers behaviour
  - competitors
  - products visibility in website/shop
  - …
- suggestion: synthetic controls, CUPED
  - use pre-experiment data to construct ML predictive model and explain part of the variance away





Demand vs Sales

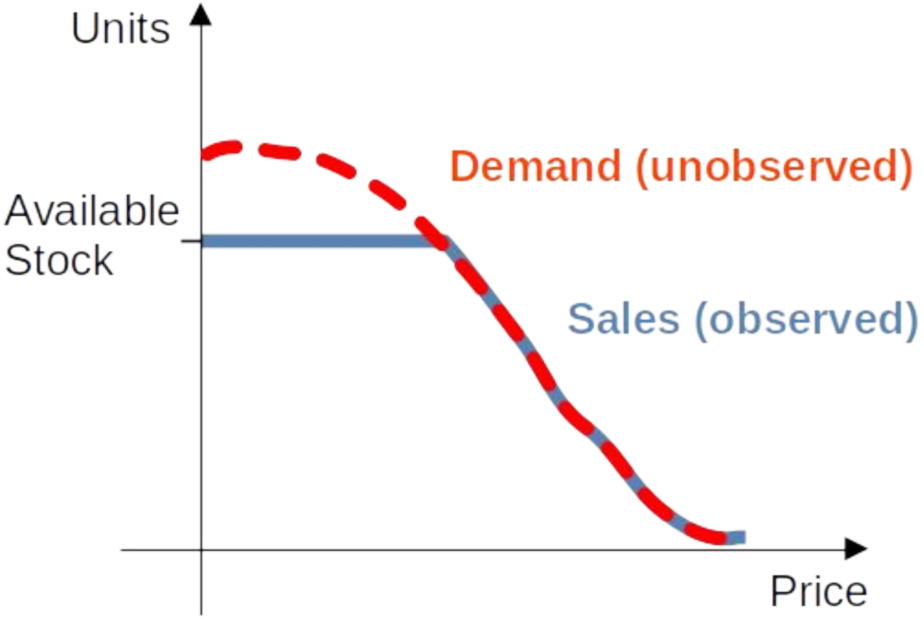
Demand vs Sales

We tried learning the relationship between demand and price.

In practice **demand can not be fully observed, but sales can.**

$$\text{Sales}_t = \min(\text{Demand}_t, \text{Stock}_t)$$

Main confounder: stock-out.



Predicting Demand from Sales: solutions

- just forecast sales, i.e. assume sales = demand
- consider methods for censored or intermittent time series
- imputation: estimate missed sales on stock-outs using non-stock-out data

YMMV

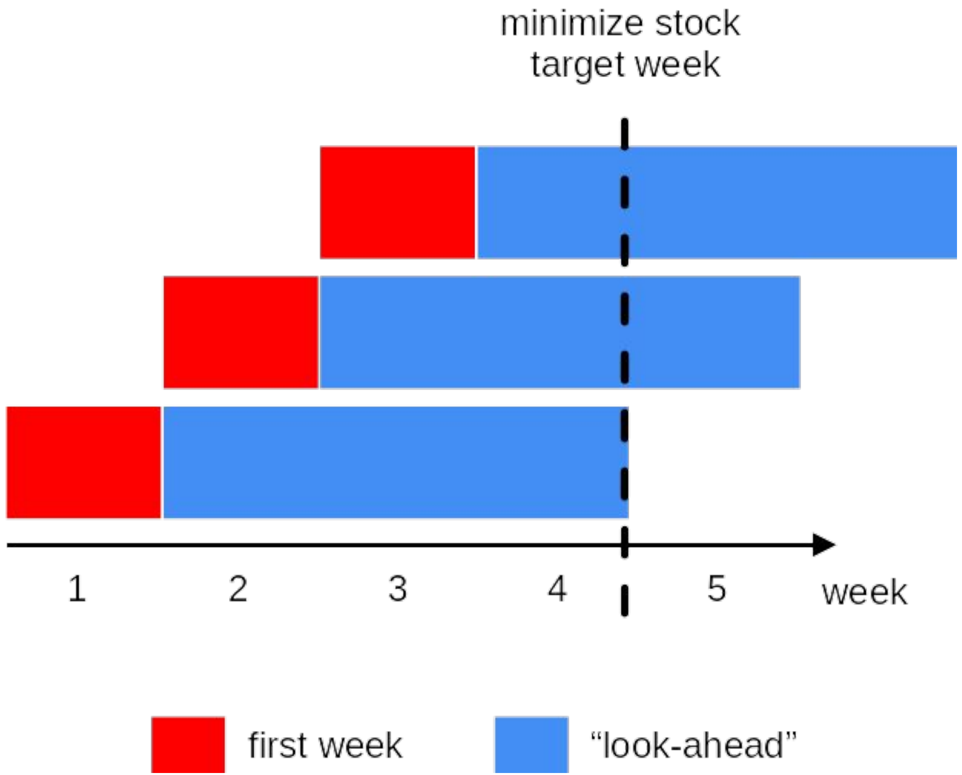
Prescriptive Analytics

Decisions Optimization

Example: monthly planning with weekly periods (link)

Pricing Manager wants to plan prices for the next week (week 1) such that:

- the profit over the next 4 weeks is maximized
- the left-over stock at the end of the month (week 4) is minimized



Planning Example - Decision Model (simplified)

Decision model as **mathematical program** (constraints + objectives):

- prices are between min and max
- last week's stock is either sold or goes into the this week's stock
- each week we sell all the stock available until either the stock is depleted or the demand is satisfied
- Objectives:
  - maximize profit
  - minimize left-over stock at t=4

$$\underline{p}_a \leq p_{at} \leq \bar{p}_a \quad \forall a \in A, t \in T$$

$$l_{a(t-1)} = S_{at} + l_{at} \quad \forall a \in A, t \in T$$

$$S_{at} = \min(Q_{at}(p_{at}), l_{a(t-1)}) \quad \forall a \in A, t \in T$$

$$\max \sum_{a \in A, t \in T} (p_{at} - c_a) S_{at}$$

$$\min \sum_{a \in A} l_{a4}$$

Indexes: a = article t = week index

Variables: p = price S = sold items l = stock

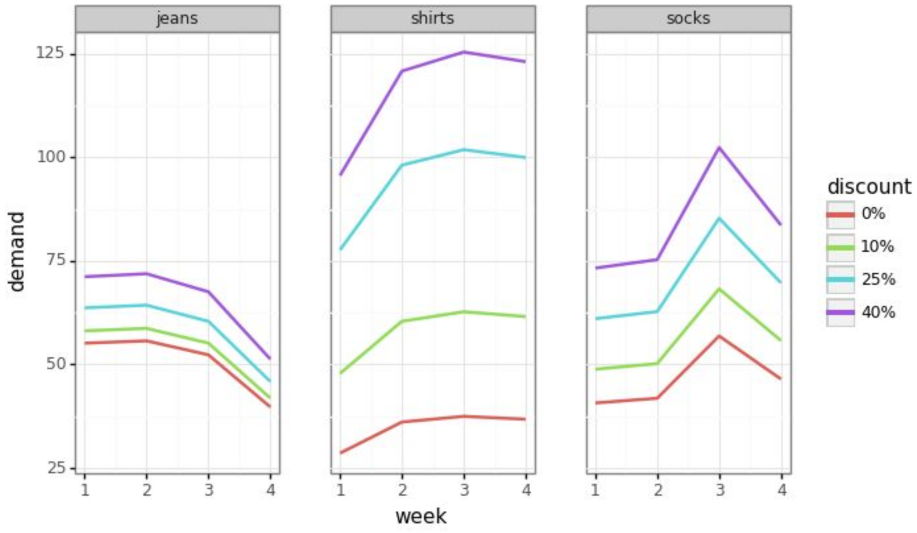
Data: Q = demand forecast c = article cost

Planning Example - Predicted Input Data

Demand forecast for each product and price.

Note: price represented as “discount” from list price:

10% discount = 90% list price



Planning Example - Implementation (link)

Model implemented in Python/PuLP.

Solved for the example of 3 items over 4 weeks.

$$\underline{p}_a \leq p_{at} \leq \bar{p}_a \quad \forall a \in A, t \in T$$

$$l_{a(t-1)} = S_{at} + l_{at} \quad \forall a \in A, t \in T$$

$$S_{at} = \min(Q_{at}(p_{at}), l_{a(t-1)}) \quad \forall a \in A, t \in T$$

$$\max \sum_{a \in A, t \in T} (p_{at} - c_a) S_{at}$$

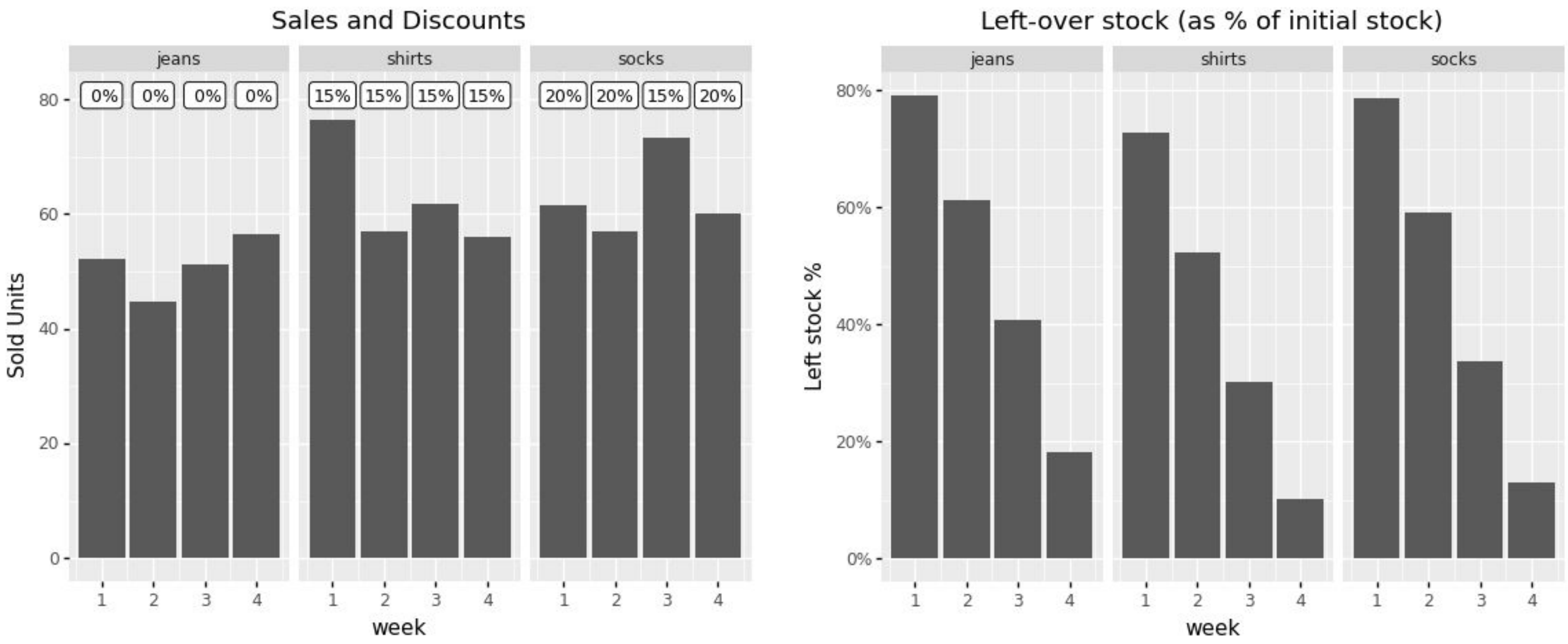
$$\min \sum_{a \in A} l_{a4}$$

Indexes: a = article t = week index

Variables: p = price S = sold items l = stock

Data: Q = demand forecast c = article cost

Planning Example - Solution (link)



Planning Example - Solution ([link]())

Solution:

- Total profit [€]: 5100
  - -2.5% from theoretical maximum profit
- Left-over stock [units]: 112
  - minimum possible left-over stock for 5100 EUR profit
  - -50% compared to theoretical maximum profit solution

- quality measure: distance from true optimum estimated
- multi-objective optimization for optimal tradeoffs:

-2.5% from theoretical maximum profit

➪

-50% left over stock

Note: 2.5% max. profit loss is chosen by the user ☝️

Zalando Use Case (link)

Big pricing problem:

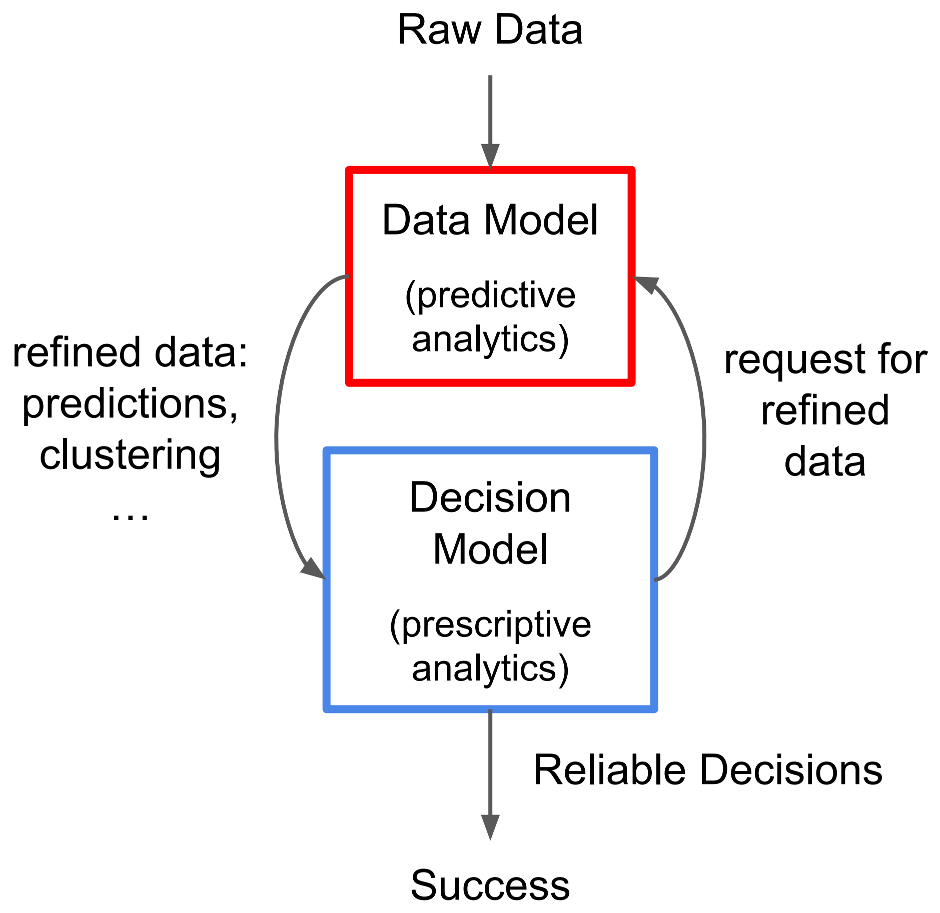
- 500k articles x 25 markets x several weeks
- user-defined targets for weekly margins (and others)
- to be solved in a few hours

⇒ It can be done 👍

Tips for guaranteed impact

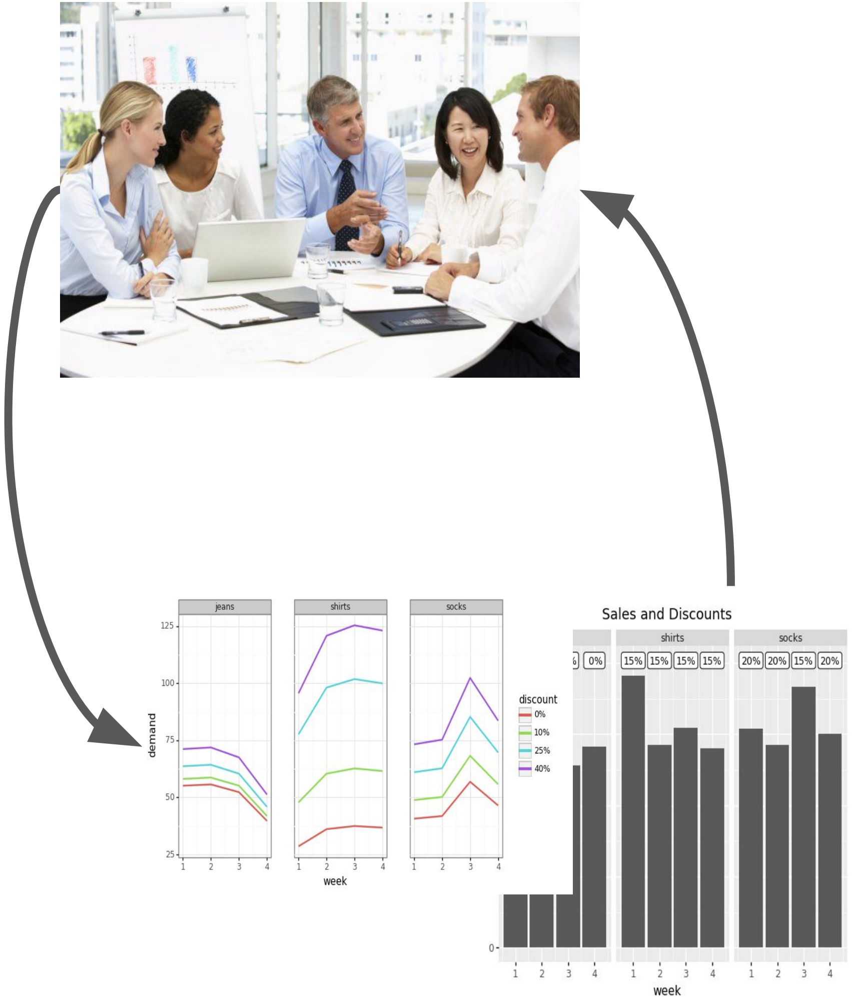
Model Decisions first, Data will follow

- **decisions first:** start modelling decisions to better understand the problem
  - see [1], [2], [3]
- **data follows:** the decision model will identify which data is needed to actually solve the problem
- keep looping between data and decision modelling until problem is clear for everyone



Playable Prototypes

- have **users play with small decision model prototype (e.g. on laptop)**
  - still “decision first”
- users can point to prototype’s output to better articulate requirements.
- very fast iteration cycle
- **final requirements much different than initial ones**
  - avoid solving the wrong problem for weeks/months/years
  - [“solving wrong problem” main cause of AI projects failure]()

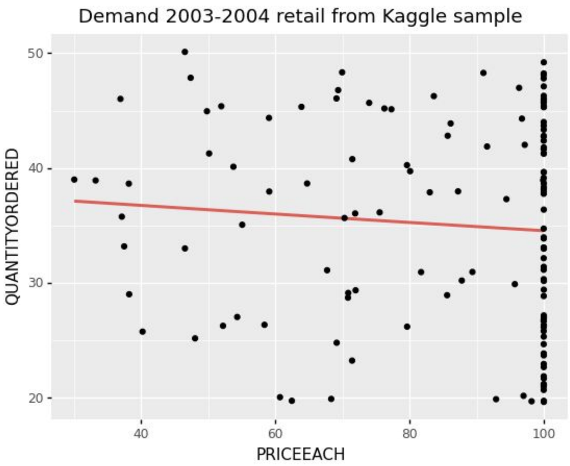


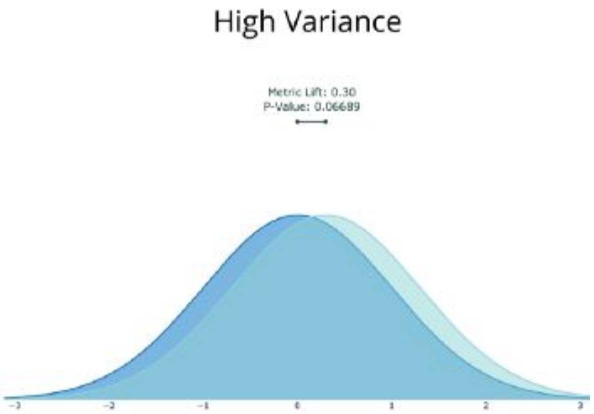
Improve people’s understanding of the problem, numbers will follow

When you do pricing analytics:

- the data is very sparse and noisy 🙁
- your forecast is not going to be accurate enough 🙁
- the AB test for your new price model is not going to show significant improvements 🙁

but…





Improve people's understanding of the problem, numbers will follow

… **your tool can still help users increase:**

- **reliability**
- **speed**
- **confidence**

of their decisioning, if:

- they can adjust forecasts to evaluate different scenarios
- they can get trustworthy optimal plans for each scenario
- thus they can easily take good decisions **together**

Analytics: value beyond numbers

*If you ask the head of Microsoft about **the most important benefit of [Analytics]**, he does not talk about [significantly improving KPIs], his quote is always that **[Analytics] drains the emotions from the decision process**.*

*Before you had **sales battling with operations about what "a number" should be**. But once you put [Analytics] in the process, **the answer is just "the [right] number"**.*
They can argue about what lead to "the number", but not "the number" itself.
This really improved the rate with which they make decisions.

-- prof. Sean Willems on PA for Large-Scale Supply Chain Optimization
https://www.youtube.com/watch?v=Via8yTYHX-A