Prima Assignment

P. Arrutia

May 2025

Problem statement

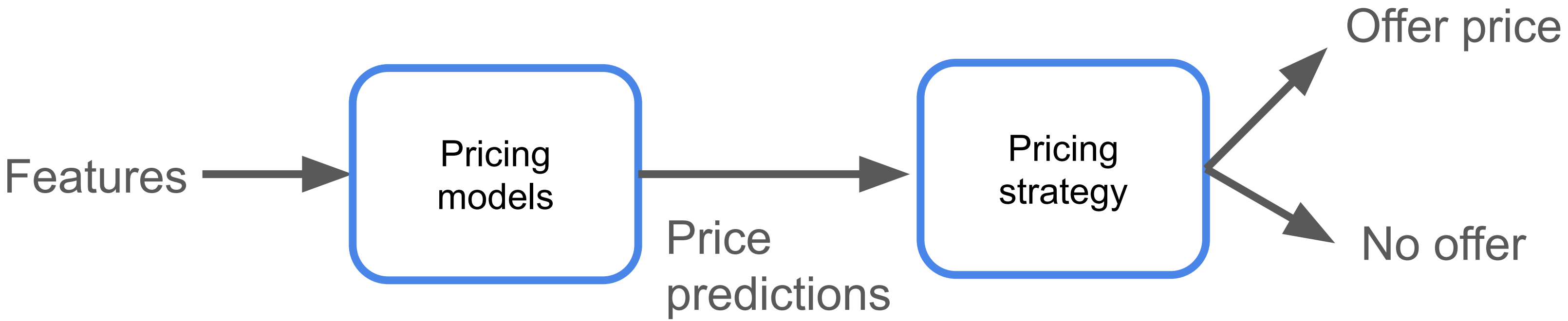
We are providing prices in price comparison website.
Rules:

1. If you are cheaper than the cheapest price of all your competitors, you sell.
2. You must sell at least 30% of the quotes.
3. Minimize the average absolute difference w.r.t. competitors’ cheapest price, calculated only on the policies you sell.

Outline of solution

Propose a 2-stage approach:

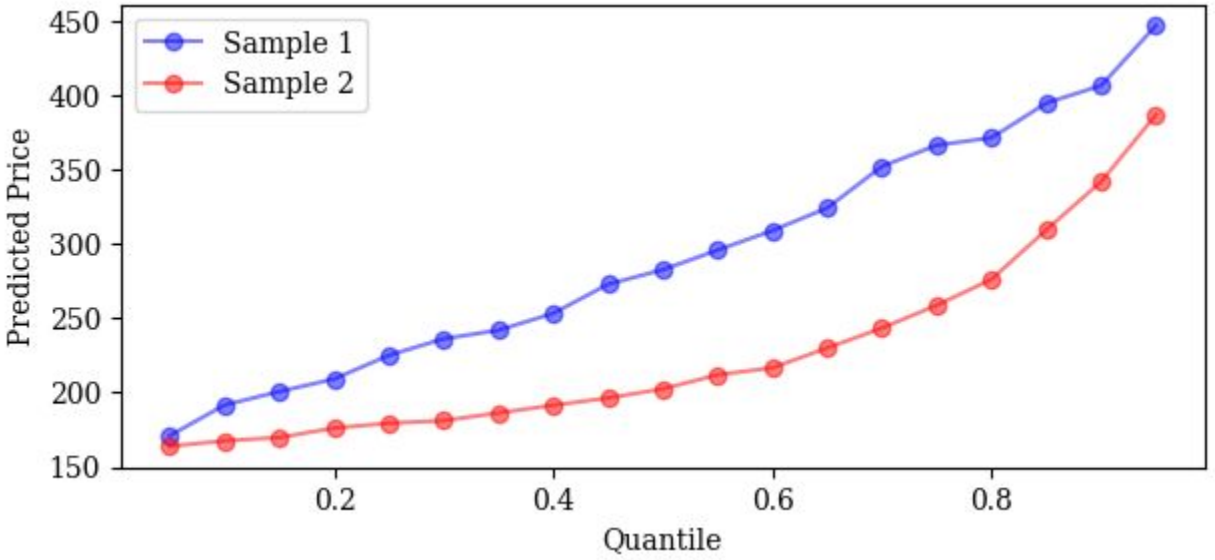
- Base models for competitor price estimation.
- Optimised pricing strategy for our particular objective.



Base models

Employ gradient-boosted trees (from XGBoost):

- Flexible, little hyperparameter tuning, fast to train.
- Perform basic feature engineering: convert datetimes, convert categoricals, fill NAs…
- Construct ensemble of quantile models for prediction uncertainty/risk estimation.



Pricing strategy

"Only" need 30% market share.

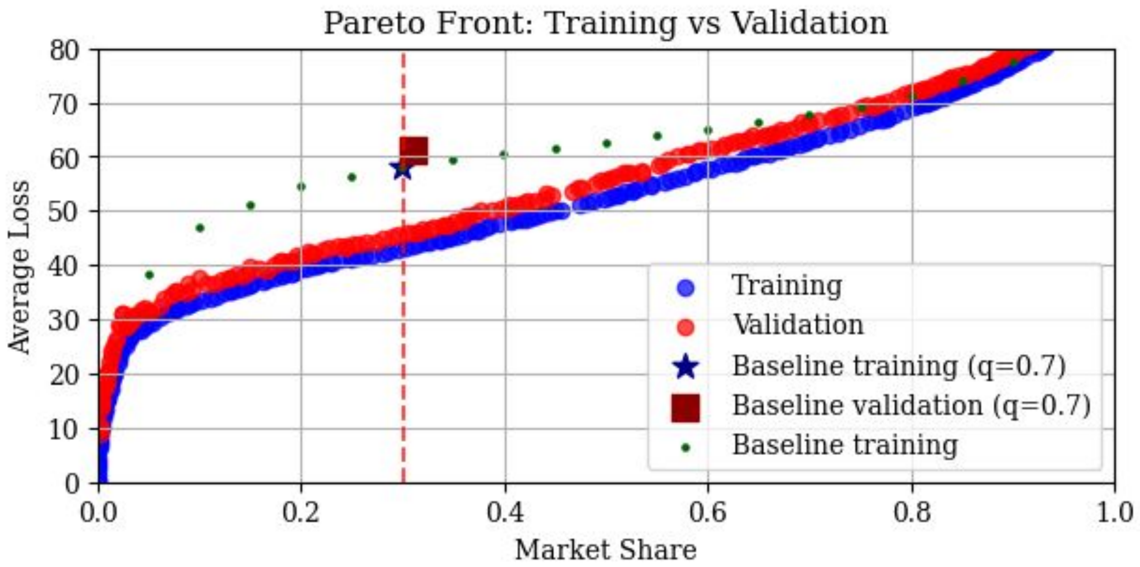
- Avoid providing price in high-uncertainty predictions.
- Parametrize strategy and use optimizer for tuning:

```
P(Q): pricing model at quantile Q.

q: quantile used for pricing estimate.
q_high: upper quantile for uncertainty estimate.
q_low: lower quantile for uncertainty estimate.
s_lim: uncertainty cutoff. Above s_lim, no price is provided.

Strategy:

    s = P(q_high) - P(q_low)
    s = (s - s.mean())/s.std()

    if s < s_lim
        Offer P(q)

    else:
        No price offered
```

Optimization procedure

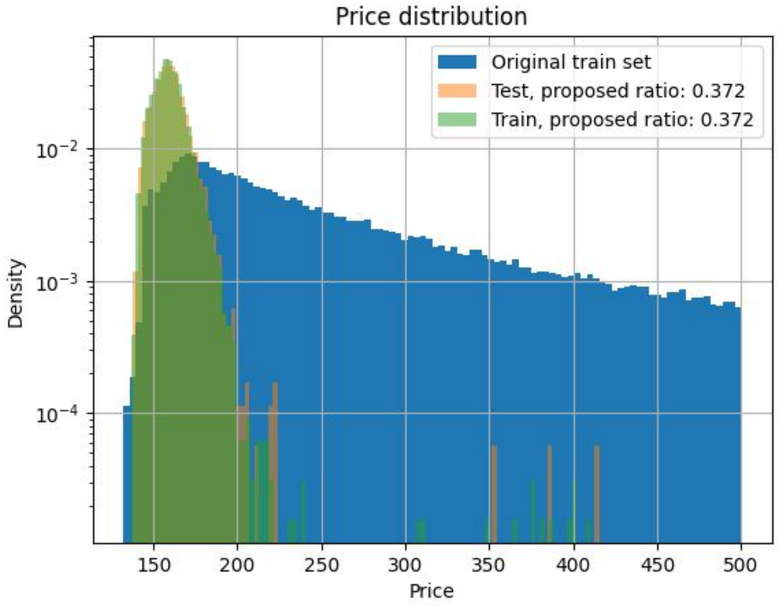
Employ multi-objective genetic algorithm

- Get Pareto front of solutions (avg_loss, market_share).
- Provides a sense of robustness of strategy.
- In the end, select only solution with lowest avg_loss, while market_share > 0.3.



Results

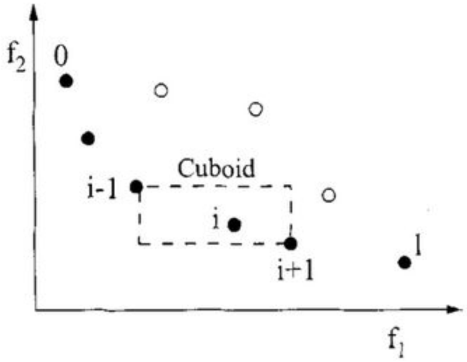
- Base model (q=0.7) showed avg loss of ~60, with ~30% market share.
- Model + strategy bring avg loss down to ~44, with ~32% market share.
- Strategy clearly shows pruning of high-price predictions (inherently more uncertain in absolute error).



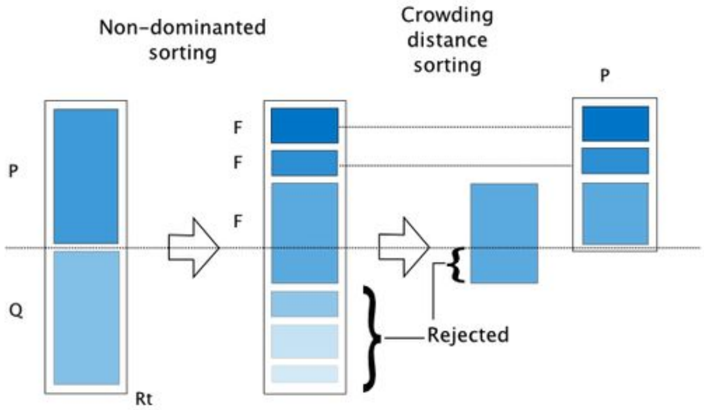
Conclusion

- Created quantile pricing models using gradient-boosted trees.
- Developed pricing strategy to avoid high-risk offers, tuning parameters with multi-objective genetic algorithm.
- Full solution outperforms pricing-only solution (on train+validation sets).
- Further improvements:
    - More sophisticated pricing strategy (at the cost of extra parameters to optimize).
    - Move to other pricing model architectures (e.g. NNs).

EXTRA SLIDES



The crowding distance is the Manhatten Distance in the objective space. However, the extreme points are desired to be kept every generation and, therefore, get assigned a crowding distance of infinity.



Furthermore, to increase some selection pressure, NSGA-II uses a binary tournament mating selection. Each individual is first compared by rank and then crowding distance. There is also a variant in the original C code where instead of using the rank, the domination criterium between two solutions is used.