



Оптимизация цен

Артём Копань

Постановка задачи

Даны сведения о сезонных ценах на P разных продуктов за W недель. Также даны сведения о среднем количестве продаж в S магазинах.

Требуется выставить цены r_{pw} на каждый продукт в каждую неделю так, чтобы выручка $F(R)$, $R = \{r_{pw}\}_{p,w=1}^{P,W}$ была максимальной.

$$F(R) = \sum_{s=1}^S \sum_{p=1}^P \sum_{w=1}^W r_{pw} \left(M_{sp} \frac{-\left(\frac{r_{pw}}{r_{pw}^{seasonal}}\right)^2 - 0.075 \frac{r_{pw}}{r_{pw}^{seasonal}} + 2.85}{1.775} + L_{spw} \right)$$

$$L_{spw} = \sum_{o=1}^p \left(\frac{r_{ow}^{seasonal}}{r_{ow}} \right)^2 C_{po} M_{so}$$

Постановка задачи



При этом должны выполняться ограничения:

$$\forall p, w \quad 0.75r_{pw}^{seasonal} \leq 1.25r_{pw}^{seasonal}$$

Используемые технологии



Для решения задачи была выбрана библиотека Puomo и солвер ipopt, который позволяет решать задачи нелинейного программирования и работать с вещественными переменными.

Немного кода



```
model = ConcreteModel()

model.r = Var(range(1, P + 1), range(1, W + 1), domain=PositiveIntegers)

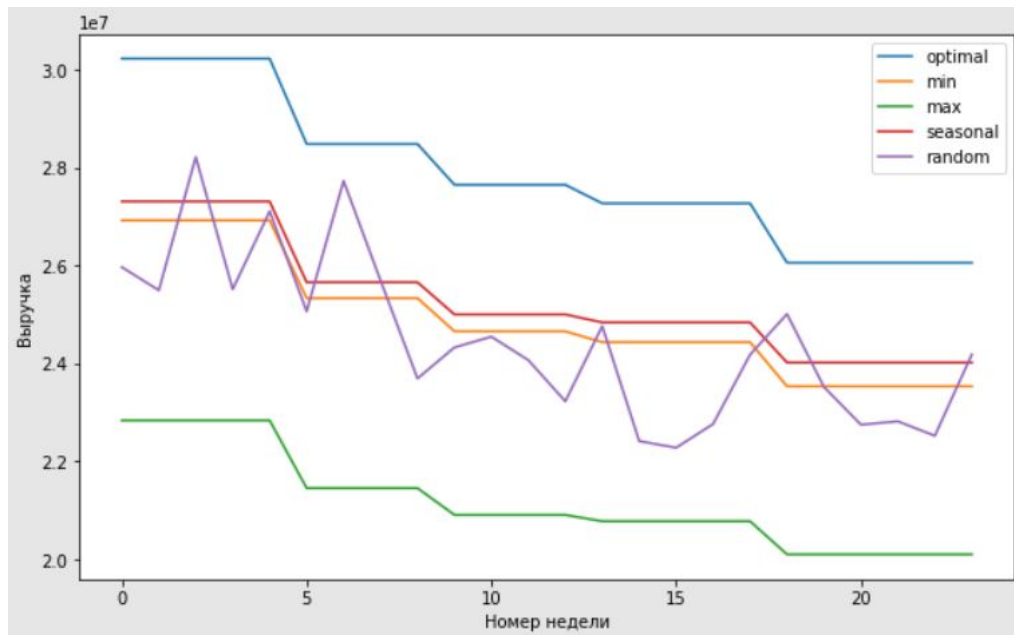
model.objective = Objective(
    expr=quicksum(quicksum(quicksum(f(s, p, w, var_matrix_to_list(model.r, P, W))
                                   for w in range(W)) for p in range(P)) for s in range(S)),
    sense=maximize)

model.constraints = ConstraintList()
for p in range(P):
    for w in range(W):
        r_pw_seasonal = prices.loc[w][f'price_{products[p]}']
        model.constraints.add(0.75 * r_pw_seasonal <= model.r[p + 1, w + 1])
        model.constraints.add(model.r[p + 1, w + 1] <= 1.25 * r_pw_seasonal)

solver = SolverFactory('ipopt')
result = solver.solve(model, tee=True)

return [[model.r[p, w].value for w in range(1, W + 1)] for p in range(1, P + 1)]
```

Результаты



Выводы



- Значения цен, полученные при оптимизации, действительно являются оптимальными
- Самая сложная часть задачи — грамотно написать выражение для оптимизации