

# Analiza coșurilor de cumpărături și recomandare de produse

## Regresie Logistică și metode de ranking

Spilevoi Bogdan, Anton Cosmin

13 ianuarie 2026

## 1 Introducere

Scopul acestei teme este analiza coșurilor de cumpărături dintr-un restaurant, cu accent pe:

- modelarea probabilității de a include un sos în coș (clasificare binară și multi-label);
- construirea unui mecanism de recomandare Top-K pentru sosuri;
- dezvoltarea unei metode de ranking pentru *upsell*, care ordonează produse candidate în funcție de utilitate (probabilitate + venit).

## 2 Dataset

Dataset-ul (`restaurant sells`) conține cumpărături asociate mai multor bonuri fiscale. Fiecare rând reprezintă un produs de pe un bon, iar coșul se reconstruiește prin grupare după `id_bon`. Coloanele utilizate sunt:

- `id_bon` – ID-ul bonului (identifică un coș / o tranzacție);
- `data_bon` – data și ora bonului (permite derivări: zi din săptămână, weekend etc.);
- `retail_product_name` – numele produsului;
- `SalePriceWithVAT` – prețul per linie (cu TVA).

Setul de date este folosit exclusiv în scop didactic și nu conține date personale despre clienți.

## 3 Preprocesare și trăsături

Preprocesarea este realizată la nivel de bon (`id_bon`), nu la nivel de rând. Principalele etape:

- conversie tipuri: `data_bon` la dată/oră, `SalePriceWithVAT` numeric;
- curățare nume produse (strip) și eliminare intrări nerelevante, când e cazul;
- construcție vector de produse per coș: count-uri (sau binar, opțional);
- agregări pe coș: `cart_size`, `distinct_products`, `total_value`;
- trăsături temporale: `day_of_week` și `is_weekend`.

Pentru a preveni scurgerile de date, sosul țintă este exclus din trăsături (și, în cerința 2.2, se poate exclude fie doar sosul curent, fie toate sosurile).

## 4 Metode

### 4.1 2.1: Regresie Logistică — “Will the client include the sauce?”

Problema este o clasificare binară:

$y = 1 \iff$  bonul conține Crazy Sauce, condiționat de faptul că bonul conține Crazy Schnitzel.

Am implementat o variantă proprie de Regresie Logistică cu *Gradient Descent*, iar pentru comparație am rulat și un model `scikit-learn LogisticRegression`. Ca baseline am folosit *majority class*.

### 4.2 2.2: Câte un model per sos + recomandare Top-K

Pentru fiecare sos  $s$  din lista de sosuri “standalone”, antrenăm un model binar:

$$y_s = 1 \iff s \in \text{bon}.$$

Pentru un coș dat, calculăm  $P(s \mid \text{coș})$  pentru fiecare sos și recomandăm Top-K sosuri cu probabilitatea cea mai mare (care nu sunt deja în coș). Evaluarea este de tip multi-label, cu metrici Hit@K și Precision@K. Ca baseline am folosit Top-K după popularitatea sosurilor în train.

### 4.3 3: Ranking pentru upsell

Am implementat un model de tip co-ocurență inspirat din Naive Bayes (count-uri pe coș, smoothing), care produce un scor pentru fiecare produs candidat, combinând:

- un termen probabilistic (bazat pe co-ocurența produselor în coșuri);
- un termen de venit (ex.  $\log$  din preț mediu / venit asociat).

Evaluarea se face prin *leave-one-out*: dintr-un coș din test se elimină un produs, iar algoritmul trebuie să îl rankeze cât mai sus.

## 5 Cadru experimental și metrici

Împărțirea datelor este realizată la nivel de bon (coș). Pentru clasificare raportăm:

- Accuracy, Precision, Recall, F1;
- ROC-AUC;
- matrice de confuzie.

Pentru recomandare / ranking raportăm:

- Hit@K: proporția cazurilor în care elementul corect apare în Top-K;
- Precision@K: proporția elementelor corecte în Top-K (mediată).

## 6 Rezultate

### 6.1 2.1: Crazy Sauce condiționat de Crazy Schnitzel

**Regresie Logistică (implementare proprie, Gradient Descent).**

- Accuracy: 0.9608
- Precision: 0.9490
- Recall: 0.9789
- F1: 0.9637
- ROC-AUC: 0.9844

Matricea de confuzie:

$$\begin{bmatrix} 157 & 10 \\ 4 & 186 \end{bmatrix}$$

**Baseline (majority class).**

- Accuracy: 0.4678
- Precision: 0.0000
- Recall: 0.0000
- F1: 0.0000
- ROC-AUC: 0.5000

Matricea de confuzie:

$$\begin{bmatrix} 167 & 0 \\ 190 & 0 \end{bmatrix}$$

**Scikit-learn LogisticRegression.**

- Accuracy: 0.9804
- Precision: 0.9791
- Recall: 0.9842
- F1: 0.9816
- ROC-AUC: 0.9917

Matricea de confuzie:

$$\begin{bmatrix} 163 & 4 \\ 3 & 187 \end{bmatrix}$$

**Interpretarea coeficienților (modelul GD).** Tabelul 1 listează trăsături cu impact pozitiv și negativ asupra probabilității de a include **Crazy Sauce**. Se observă că trăsăturile de dimensiune a coșului (**distinct\_products**, **cart\_size**) au coeficienți pozitivi mari, sugerând că sosul este mai probabil în coșuri mai „bogate”. În schimb, prezența altor sosuri are coeficienți negativi, ceea ce este intuitiv: dacă un alt sos este în coș, probabilitatea de a adăuga și **Crazy Sauce** scade.

Tabela 1: Top coeficienți pozitivi și negativi (Regresie Logistică GD) pentru 2.1.

Pozitivi (Top)	Negativi (Top)
distinct_products (1.9620)	Cheddar Sauce (-2.5154)
cart_size (1.4984)	Garlic Sauce (-2.1089)
Pepsi Cola 0.25L Doze (0.6018)	Blueberry Sauce (-1.8091)
Baked potatoes (0.3465)	Pink Sauce (-0.9987)
Mac & cheese (0.3417)	Tomato Sauce (-0.9861)
total_value (0.3018)	Spicy Sauce (-0.9441)

## 6.2 2.2: Recomandare sosuri (Top-K)

Evaluare pentru recomandare de sosuri (coș fără sosuri, Top-3):

- **Model-based recommender:** Hit@3 = 0.8032, Precision@3 = 0.2918
- **Popularity baseline:** Hit@3 = 1.0000, Precision@3 = 0.1443

Observație: baseline-ul de popularitate atinge Hit@3 foarte mare, deoarece lista conține sosuri foarte frecvente, deci „nimerește” des cel puțin un sos real. Totuși, Precision@3 este considerabil mai mică față de model, ceea ce indică faptul că modelul produce recomandări mai „precise” (mai relevante), chiar dacă baseline-ul are o acoperire (hit) mai mare.

**Exemplu de recomandare (coș manual).** Pentru coșul: [Baked potatoes, Pepsi Cola 0.25L Doze], Top-K sosuri recomandate de model (ordonate după probabilitate) sunt:

- Crazy Sauce:  $P = 0.281$
- Cheddar Sauce:  $P = 0.151$
- Garlic Sauce:  $P = 0.115$
- Blueberry Sauce:  $P = 0.093$
- Spicy Sauce:  $P = 0.075$
- Tomato Sauce:  $P = 0.034$
- Extra Cheddar Sauce:  $P = 0.018$
- Pink Sauce:  $P = 0.017$

## 6.3 3: Ranking pentru upsell (leave-one-out)

Model (co-ocurență + venit).

- Hit@1 = 0.2855
- Hit@3 = 0.5509
- Hit@5 = 0.6503

### Baseline-uri.

- Popularity baseline: Hit@1 = 0.1135, Hit@3 = 0.3656, Hit@5 = 0.4716
- Revenue baseline: Hit@1 = 0.1110, Hit@3 = 0.2404, Hit@5 = 0.3748

Modelul depășește clar baseline-urile la toate valorile lui  $K$ , ceea ce sugerează că folosirea co-ocurenței (context din coș) aduce informație predictivă reală față de simple ordonări globale (popularitate/venit).

**Exemplu de ranking.** Pentru coșul parțial [Crazy Schnitzel, Crazy Sauce], primele recomandări sunt:

[Mac & cheese, Pepsi Cola 0.25L Doze, Baked potatoes, ...]

ceea ce este plauzibil pentru un scenariu de upsell (garnituri/băuturi asociate frecvent).

## 7 Concluzii și direcții de îmbunătățire

Rezultatele indică faptul că:

- Regresia Logistică (inclusiv implementarea proprie) oferă performanțe foarte bune în problema 2.1, mult peste baseline.
- Pentru recomandarea de sosuri, modelul produce recomandări mai precise decât baseline-ul de popularitate, deși baseline-ul are hit foarte ridicat.
- Pentru ranking, metoda bazată pe co-ocurență + venit depășește semnificativ ordonările globale.

Direcții viitoare:

- split temporal (train pe luni mai vechi → test pe luni mai noi) pentru evaluare mai realistă;
- regularizare (L2) și calibrare probabilități pentru LR;
- includerea unor interacțiuni între produse (pair features) pentru modelele de sos;
- rafinarea scorului de ranking (ex. combinarea  $P(p|\text{coș})$  cu venit total per produs sau marjă).

## 8 Contribuții

Spilevoi Bogdan - Ex 2.1, Ex 2.2 Anton Cosmin - Ex 3, Doc