

Predictia Sosurilor în Restaurant

Machine Learning - Practical Assignment

Mario Emanuel Tescovschi

Ianuarie 2025

1 Descrierea Problemei și a Dataset-ului

1.1 Dataset

Datele provin de la un restaurant și conțin 7,869 bonuri fiscale cu 28,039 linii de produse.

- **Interval temporal:** 05.09.2025 - 03.12.2025
- **Produse unice:** 59
- **Coș mediu:** 3.56 produse/bon

1.2 Sosuri disponibile

Crazy Sauce, Cheddar Sauce, Extra Cheddar Sauce, Garlic Sauce, Tomato Sauce, Blueberry Sauce, Spicy Sauce, Pink Sauce.

2 LR #1: Crazy Sauce condiționat de Crazy Schnitzel

2.1 Problemă

Clasificare binară: prezice dacă un client care comandă Crazy Schnitzel va comanda și Crazy Sauce.

2.2 Preprocesare

1. Filtrare bonuri care conțin Crazy Schnitzel (1,783 din 7,869)
2. Extragere features:
 - **Temporal:** day_of_week (1-7), is_weekend, hour
 - **Agregări:** cart_size, distinct_products, total_value
 - **Produse:** contor pentru fiecare produs în coș
3. Excludere Crazy Sauce din features (evitare data leakage)

2.3 Implementare

Regresie Logistică de la zero folosind Gradient Descent:

- Learning rate: 0.1
- Iterații: 1000
- Regularizare L2: $\lambda = 0.01$

2.4 Rezultate

Metrică	Model	Baseline
Accuracy	95.52%	53.22%
Precision	96.28%	-
Recall	95.26%	-
F1	95.77%	-
ROC-AUC	99.64%	50%

Tabela 1: Metrici de evaluare LR #1

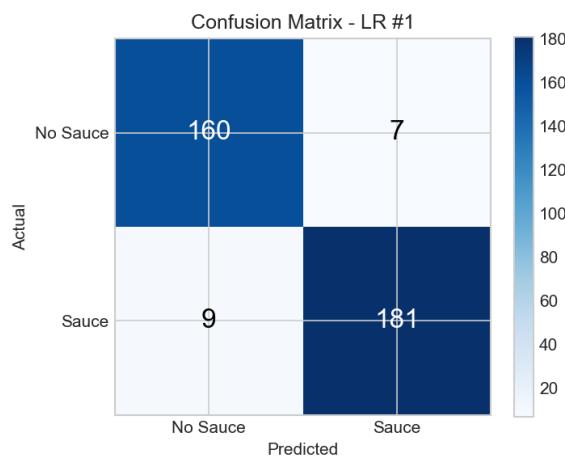


Figura 1: Matricea de confuzie - LR #1

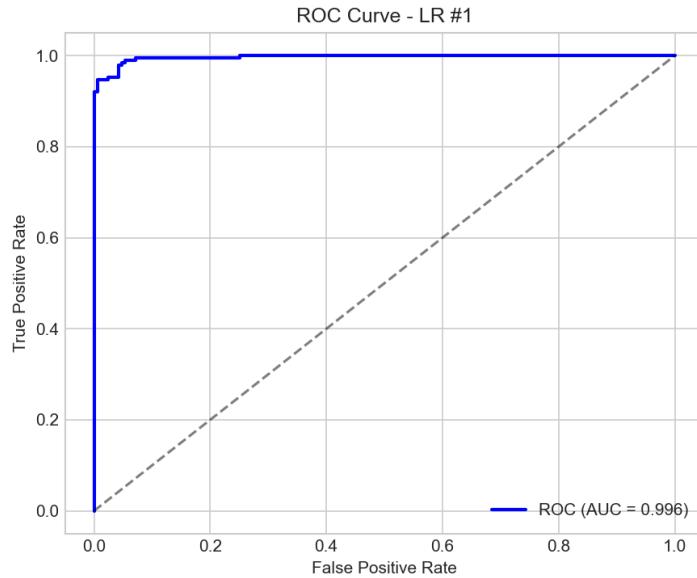


Figura 2: Curba ROC - LR #1

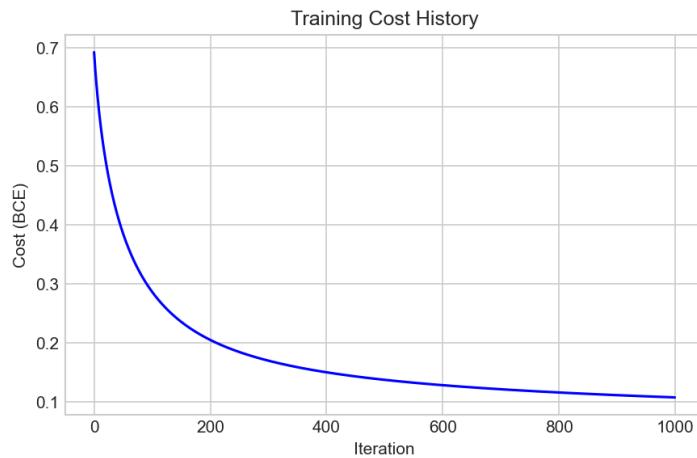


Figura 3: Evoluția costului în timpul antrenării

2.5 Interpretare Coeficienti

Feature	Weight
distinct_products	1.84
cart_size	1.65
prod_pepsi_col	0.61
prod_baked_potatoe	0.46
prod_cheddar_sauce	-2.53
prod_garlic_sauce	-2.10
prod_blueberry_sauce	-1.70

Tabela 2: Top features pozitive și negative

Concluzie: Sosurile se exclud reciproc. Coșurile mari și diverse cresc probabilitatea de Crazy Sauce.

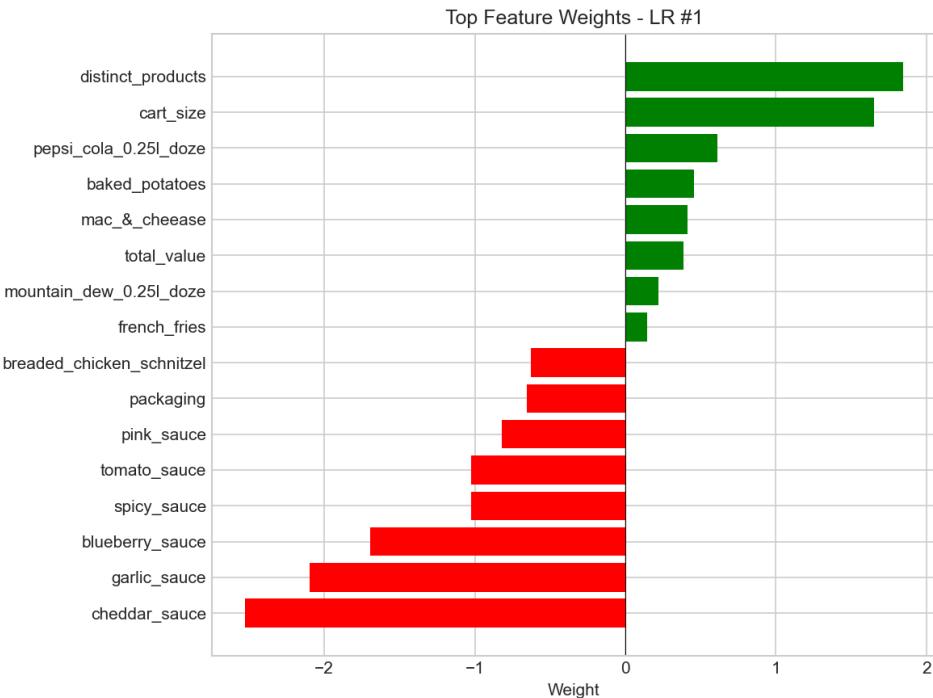


Figura 4: Ponderile features în model

3 LR #2: Multi-Sauce + Recomandări

3.1 Problemă

Antrenare câte un model LR pentru fiecare sos. Recomandare Top-K sosuri pentru un cos.

3.2 Rezultate

Sos	Accuracy	AUC
Crazy Sauce	0.853	0.887
Cheddar Sauce	0.864	0.823
Extra Cheddar Sauce	0.996	0.946
Garlic Sauce	0.900	0.788
Tomato Sauce	0.970	0.765
Blueberry Sauce	0.899	0.834
Spicy Sauce	0.955	0.830
Pink Sauce	0.986	0.800

Tabela 3: Performanța modelelor per sos

K	Model	Baseline
1	44.7%	35.9%
3	81.3%	73.6%
5	93.7%	94.1%

Tabela 4: Hit@K - Model vs Baseline popularitate

Modelul bate baseline-ul la K mic, converge la K=5.

4 Ranking: Naive Bayes pentru Upsell

4.1 Problemă

Pentru un cosă parțial, ierarhizează produsele candidate pentru upsell.

$$\text{Score}(p|\cosă) = P(p|\cosă) \times \text{pret}(p)$$

4.2 Implementare

Naive Bayes de la zero bazat pe co-ocurențe produs-produs.

4.3 Candidați

14 produse (sosuri + băuturi + garnituri).

4.4 Evaluare

Eliminare 1 produs din cosă, încercare de recuperare prin ranking.

K	Naive Bayes	Popularity
1	45.9%	22.8%
3	79.0%	62.3%
5	90.6%	81.2%

Tabela 5: Hit@K - Ranking

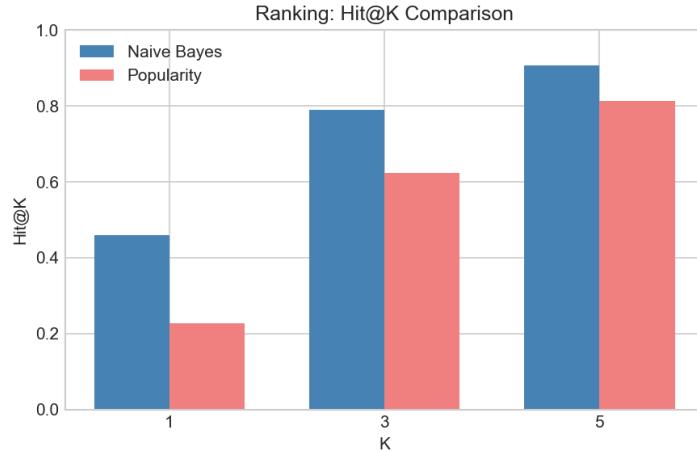


Figura 5: Comparație Hit@K: Naive Bayes vs Popularity

Naive Bayes e de 2x mai bun la K=1 (45.9% vs 22.8%).

5 Concluzii

5.1 Ce a mers bine

- LR #1 are performanță excelentă (AUC 0.996) datorită corelațiilor puternice între produse
- Sosurile se exclud reciproc - insight valoros pentru business
- Naive Bayes pentru ranking bate baseline-ul semnificativ

5.2 Limitări

- Dataset relativ mic (7,869 bonuri)
- Unele sosuri au distribuție dezechilibrată (Extra Cheddar: 0.3%)
- Nu am implementat interacțiuni între features

5.3 Directii de îmbunătățire

- Adăugare feature-uri de interacțiune (ex: Crazy Schnitzel × Baked Potatoes)
- Testare cu alte modele
- Împărțire temporală a datelor (train pe luni vechi, test pe luni noi)
- Optimizare hiperparametri (learning rate, regularizare)

Anexă: Output-uri Terminal

```
(.venv) mariotescovschi@MacBook-Pro Practical Assignment % python train_lr1.py

Dataset: 1783 receipts with Crazy Schnitzel
Label distribution: {1: 948, 0: 835}
Train: 1426, Test: 357

--- Training ---
Iteration 100/1000, Cost: 0.287812
Iteration 200/1000, Cost: 0.205480
Iteration 300/1000, Cost: 0.170062
Iteration 400/1000, Cost: 0.150255
Iteration 500/1000, Cost: 0.137429
Iteration 600/1000, Cost: 0.128310
Iteration 700/1000, Cost: 0.121397
Iteration 800/1000, Cost: 0.115904
Iteration 900/1000, Cost: 0.111383
Iteration 1000/1000, Cost: 0.107555

--- Test Results ---
Accuracy: 0.9552
Confusion Matrix: TN=160, FP=7, FN=9, TP=181
Precision: 0.9628, Recall: 0.9526, F1: 0.9577
ROC-AUC: 0.9964

Baseline (majority=1): 0.5322

--- Feature Weights ---

Positive (increase Crazy Sauce probability):
    feature      weight
    distinct_products 1.844238
                cart_size 1.653343
prod_pepsi_col_0.25l_doze 0.610833
    prod_baked_potatoes 0.458528
    prod_mac_&_cheease 0.414869
                total_value 0.386096
prod_mountain_dew_0.25l_doze 0.222700
    prod_french_fries 0.145289
prod_aqua_carpatica_minerala_0.5l 0.111613
    prod_pepsi_zero_can_0.33l 0.111001

Negative (decrease probability):
    feature      weight
prod_mac_&_cheese_with_crispy_bacon -0.539986
prod_crazy_fries_with_cheddar_sauce -0.586134
    prod_breaded_chicken_schnitzel -0.627960
                prod_packaging -0.654830
                prod_pink_sauce -0.822590
                prod_tomato_sauce -1.024651
                prod_spicy_sauce -1.025538
prod_blueberry_sauce -1.696636
    prod_garlic_sauce -2.097242
    prod_cheddar_sauce -2.525384
```

Figura 6: Output train_lr1.py

```
(.venv) mariotescovschi@MacBook-Pro Practical Assignment % python train_lr2.py
Dataset: 7869 receipts

==== Training Models ====
Crazy Sauce: Accuracy=0.853, AUC=0.887
Cheddar Sauce: Accuracy=0.864, AUC=0.823
Extra Cheddar Sauce: Accuracy=0.996, AUC=0.946
Garlic Sauce: Accuracy=0.900, AUC=0.788
Tomato Sauce: Accuracy=0.970, AUC=0.765
Blueberry Sauce: Accuracy=0.899, AUC=0.834
Spicy Sauce: Accuracy=0.955, AUC=0.830
Pink Sauce: Accuracy=0.986, AUC=0.800

==== Recommendation Evaluation ====
Popularity baseline: ['Crazy Sauce', 'Cheddar Sauce', 'Garlic Sauce']

Hit@K Results:
K      Model          Baseline
-----  

1      0.447          0.359
3      0.813          0.736
5      0.937          0.941

==== Example Recommendation ====
Actual sauces in cart: ['Garlic Sauce']
Model recommends: [('Cheddar Sauce', '0.26'), ('Crazy Sauce', '0.21'), ('Blueberry Sauce', '0.12')]
```

Figura 7: Output train_lr2.py

```
(.venv) mariotescovschi@MacBook-Pro Practical Assignment % python train_ranking.py
Train: 6295, Test: 1574

==== Ranking Evaluation ====
Candidates: 14 products

Total test cases: 1715

Hit@K:
K      Naive Bayes      Popularity
-----  

1      0.459            0.228
3      0.790            0.623
5      0.906            0.812

==== Example ====
Cart: ['Crazy Schnitzel', 'French Fries']
Top 5:
    Pepsi Cola 0.25L Doze: score=0.0008, price=12.00
    Crazy Sauce: score=0.0007, price=6.90
    Aqua Carpatica Minerala 0.5L: score=0.0006, price=12.00
    Mountain Dew 0.25L Doze: score=0.0006, price=9.00
    Blueberry Sauce: score=0.0003, price=7.90
```

Figura 8: Output train_ranking.py