Practical Assignment - Machine Learning 2025 Fall

Restaurant Sales Analysis: Sauce Prediction and Product Ranking

Elisa Mercas & Denis Munteanu

January 2025

Rezumat

Acest raport prezintă implementarea și analiza unor algoritmi de Machine Learning aplicați pe un set de date cu tranzacții dintr-un restaurant. Scopul principal este de a prezice dacă un client va cumpăra un sos și de a crea un sistem de ranking pentru produse cu potențial de upselling. Am implementat Logistic Regression from scratch folosind Gradient Descent, precum și Naive Bayes și k-NN pentru ranking. Rezultatele arată că modelul LR #1 obține un F1-score de 0.76 pentru predicția Crazy Sauce, iar pentru ranking, baseline-ul de popularitate obține cele mai bune rezultate cu Hit@5 de 0.66.

Cuprins

1 Introducere

1.1 Descrierea Problemei

Acest proiect abordează analiza datelor de vânzări dintr-un restaurant pentru a:

1. **LR #1**: Prezice dacă un client care comandă Crazy Schnitzel va cumpăra și Crazy Sauce
2. **LR #2**: Crea un sistem de recomandare pentru sosuri multiple
3. **Ranking**: Construi un sistem de ranking pentru produse cu potențial de upselling

1.2 Descrierea Dataset-ului

Dataset-ul conține tranzacții de la un restaurant, cu următoarele caracteristici:

Statistică	**Valoare**
Perioada	5 Septembrie - 3 Decembrie 2025
Total bonuri	7,869
Total linii	28,039
Produse unice	59
Coș mediu	3.56 produse per bon
Valoare medie coș	67.87 RON

Tabela 1: Statistici dataset

1.2.1 Distribuția pe Categorii

Categorie	**Vânzări**
Schnitzel	6,978
Sauce	6,117
Drinks	5,962
Mac & Cheese	3,546
Sides	2,918
Other	2,312
Salad	206

Tabela 2: Distribuția vânzărilor pe categorii

1.2.2 Coloane Utilizate

- `id_bon` – Identificator unic pentru fiecare bon/tranzacție
- `data_bon` – Data și ora tranzacției
- `retail_product_name` – Numele produsului
- `SalePriceWithVAT` – Prețul cu TVA

1.2.3 Sosuri Standalone

Sosurile analizate sunt prezentate în tabelul 3.

Sos	Vânzări	Procent
Crazy Sauce	1,662	20.3%
Cheddar Sauce	1,100	13.1%
Garlic Sauce	778	9.4%
Blueberry Sauce	743	9.0%
Spicy Sauce	386	4.9%
Tomato Sauce	212	2.7%
Pink Sauce	147	1.9%
Extra Cheddar Sauce	24	0.3%

Tabela 3: Vânzări sosuri (Total: 5,052 = 18% din toate vânzările)

2 Preprocesarea Datelor

2.1 Feature Engineering

Pentru a transforma datele brute în features utilizabile de algoritmi, am aplicat:

2.1.1 Vectorul de Produse

Pentru fiecare produs p dintr-un bon, am creat:

- `has_p` – variabilă binară (1 dacă produsul este în coș, 0 altfel)
- `count_p` – numărul de apariții ale produsului în coș

2.1.2 Agregări la Nivel de Coș

- `cart_size` – numărul total de produse din coș
- `distinct_products` – numărul de produse unice
- `total_value` – $\sum$ SalePriceWithVAT

2.1.3 Features Temporale

- `day_of_week` – ziua săptămânii (1-7)
- `hour` – ora tranzacției
- `is_weekend` – 1 dacă weekend, 0 altfel

2.2 Împărțirea Datelor

Am împărțit datele la nivel de **bon** (nu pe rânduri individuale) pentru a evita data leakage:

- Training set: 80% din bonuri
- Test set: 20% din bonuri
- Stratified split pentru a menține proporțiile claselor

3 Logistic Regression #1: Crazy Sauce Prediction

3.1 Formularea Problemei

Obiectiv: Pentru bonurile care conțin Crazy Schnitzel, prezice dacă bonul conține și Crazy Sauce.

- **Input (X)**: 56 features ce descriu conținutul coșului (excluzând toate sosurile)
- **Output (y)**: 1 dacă Crazy Sauce este în coș, 0 altfel
- **Dataset**: 1,783 bonuri cu Crazy Schnitzel
- **Rată conversie**: 53.2% (948 din 1,783 bonuri conțin și Crazy Sauce)

3.2 Implementare Logistic Regression from Scratch

3.2.1 Funcția Sigmoid

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{1}$$

3.2.2 Cross-Entropy Loss

$$\mathcal{L}(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right] \tag{2}$$

3.2.3 Gradient Descent Update

$$\theta := \theta - \alpha \cdot \frac{1}{m} X^T (\sigma(X\theta) - y) \tag{3}$$

unde $\alpha = 0.1$ este learning rate-ul.

3.2.4 Regularizare L2

Am adăugat regularizare L2 ($\lambda = 0.01$) pentru a preveni overfitting-ul:

$$\mathcal{L}_{reg}(\theta) = \mathcal{L}(\theta) + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2 \tag{4}$$

3.2.5 Antrenare

Modelul a fost antrenat pe 1,000 de iterații, cu loss-ul scăzând de la 0.6931 (inițial) la 0.5575 (final).

3.3 Rezultate

Metrică	Valoare
Accuracy	0.6975
Precision	0.6565
Recall	0.9053
F1 Score	0.7611
Specificity	0.4611

Tabela 4: Rezultate LR #1 pe setul de test (357 samples)

3.3.1 Matricea de Confuzie

	Predicted Neg	**Predicted Pos**
Actual Neg	77	90
Actual Pos	18	172

Tabela 5: Matricea de confuzie LR #1

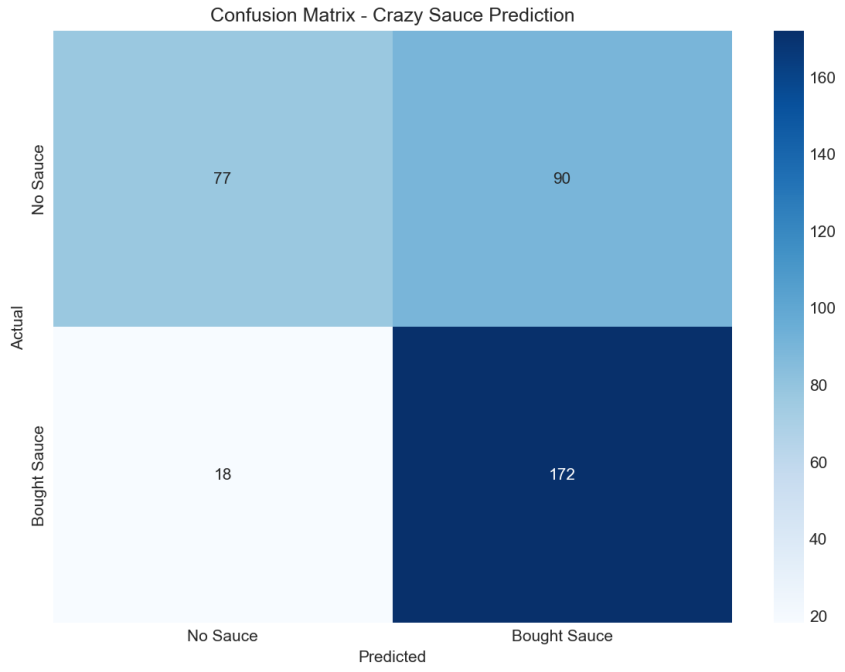

Figura 1: Matricea de confuzie pentru LR #1

3.3.2 ROC Curve și AUC

Pentru a evalua capacitatea modelului de a discrimina între clase, am calculat curba ROC și scorul AUC.

Metrică	Valoare
ROC-AUC	0.7142

Tabela 6: ROC-AUC pentru LR #1

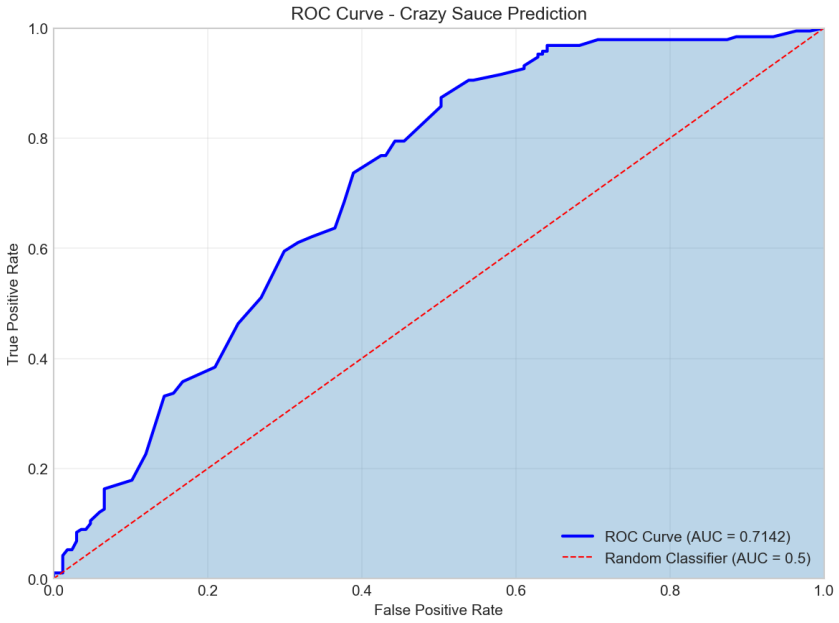

Figura 2: Curba ROC pentru LR #1. AUC de 0.71 indică o capacitate bună de discriminare, semnificativ peste baseline-ul aleator (AUC=0.5).

Figura 3: Evoluția loss-ului în timpul antrenării

3.4 Interpretarea Coeficienților

Coeficienții cu valori pozitive mari indică produse care cresc probabilitatea de a cumpăra Crazy Sauce, în timp ce coeficienții negativi indică produse care scad această probabilitate.

Feature	Coeficient	Efect
distinct_products	+0.7846	POZITIV
has_Mac & cheease	+0.3411	POZITIV
has_Pepsi Cola 0.25L Doze	+0.2795	POZITIV
cart_size	+0.2237	POZITIV
has_Aqua Carpatica Minerala	+0.1547	POZITIV
has_Breaded Chicken Schnitzel	-0.4179	NEGATIV
has_Crazy Fries with Parmesan	-0.3659	NEGATIV
has_Prigat Still Orange	-0.3309	NEGATIV
has_Breaded Pork Schnitzel	-0.2659	NEGATIV
has_Mac & Cheese with Bacon	-0.2593	NEGATIV

Tabela 7: Top features pentru predicția Crazy Sauce

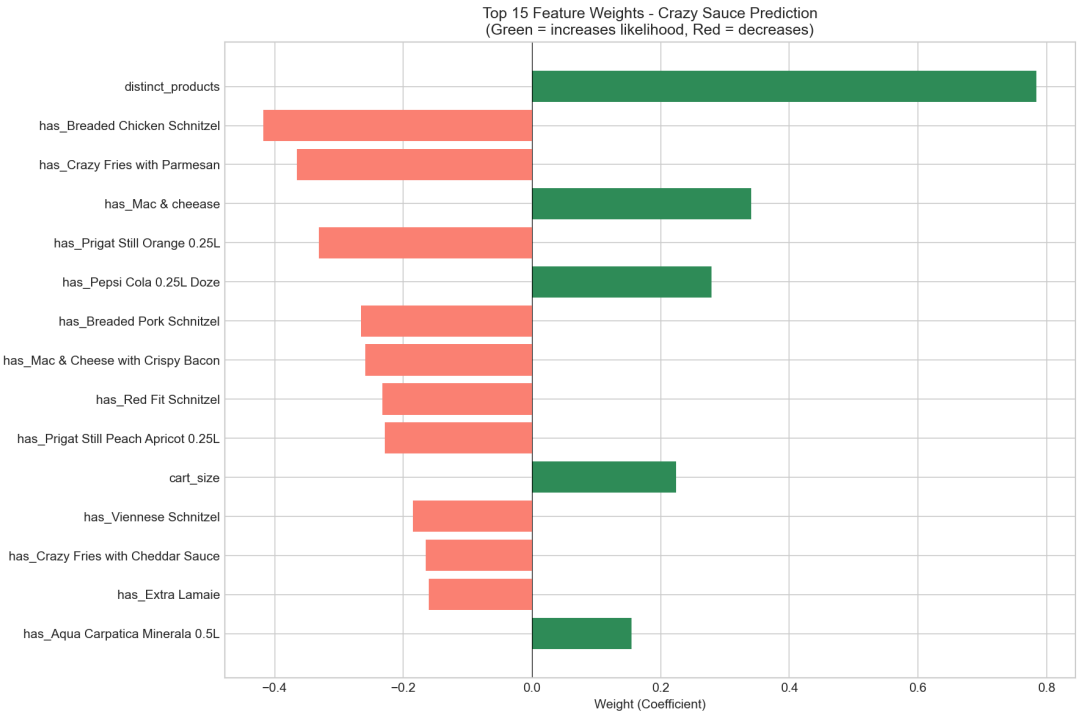

Figura 4: Importanța features pentru LR #1

Interpretare: Clienții cu coșuri mai diverse (distinct_products mare) și care comandă Mac & Cheese au o probabilitate mai mare să cumpere Crazy Sauce. Pe de altă parte, cei care comandă alt tip de schnitzel (Breaded Chicken sau Pork) nu mai cumpără Crazy Sauce.

3.5 Comparație cu Baseline

Baseline-ul (majority class) ar prezice întotdeauna clasa majoritară (pozitivă = 53.2%), obținând o acuratețe de 53.2%. Modelul nostru cu acuratețe de 69.75% depășește semnificativ baseline-ul.

4 Logistic Regression #2: Multi-Sauce Recommendation

4.1 Formularea Problemei

Pentru fiecare sos s din lista de sosuri, am antrenat un model separat:

- **Input (X)**: Features ale coșului (excluzând toate sosurile)
- **Output (y_s)**: 1 dacă sosul s este în coș, 0 altfel

4.2 Pseudo-Recomandare

Pentru un coș dat (fără sos), calculăm $P(s|\text{coș})$ pentru fiecare sos și recomandăm Top-K sosuri cu probabilitatea cea mai mare.

Exemplu de recomandare pentru un coș cu Crazy Schnitzel, French Fries, la ora 13:00:

- Crazy Sauce: 21.7% probabilitate
- Cheddar Sauce: 3.9% probabilitate
- Garlic Sauce: 3.9% probabilitate

4.3 Evaluare Hit@K pentru Recomandare

Am evaluat sistemul de recomandare folosind metrica Hit@K: pentru fiecare bon de test care conține un sos, verificăm dacă sosul real apare în Top-K recomandări.

Metodă	**Hit@1**	**Hit@3**	**Hit@5**
LR Recommendation	0.4320	0.7720	0.9400
Popularity Baseline	0.3780	0.7440	0.9320

Tabela 8: Comparație Hit@K: LR Recommendation vs Popularity Baseline

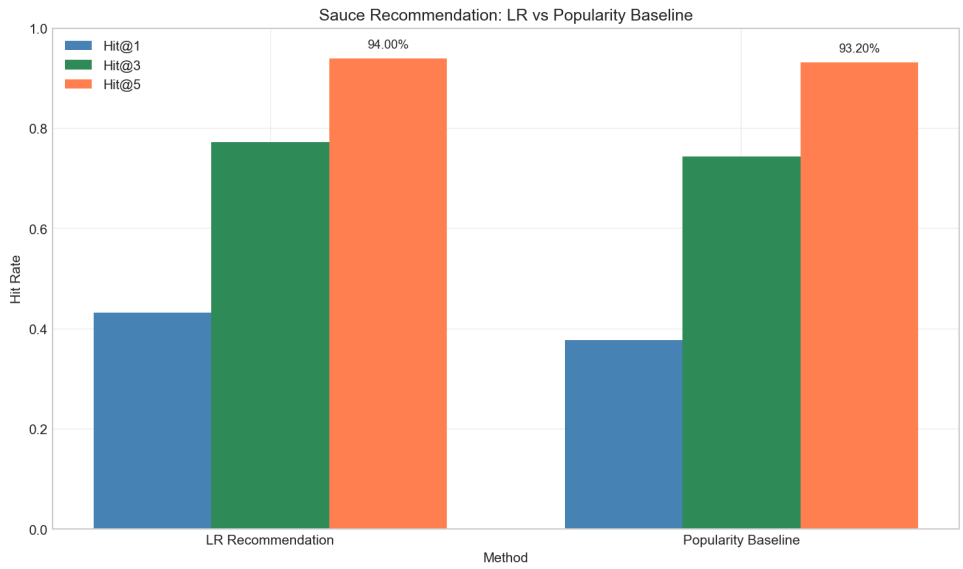

Figura 5: Comparație Hit@K între sistemul LR și baseline-ul de popularitate

Observație: Sistemul LR depășește baseline-ul de popularitate la toate metricile, în special la Hit@1 (43.2% vs 37.8%), demonstrând că folosirea feature-urilor contextuale (conținutul coșului, tipul de schnitzel) aduce valoare adăugată față de recomandarea simplă bazată pe popularitate globală. La Hit@5, ambele sisteme ating performanțe foarte bune ($> 93\%$).

4.4 Rezultate Comparative

Sos	**Base Rate**	**Accuracy**	**Precision**	**Recall**	**F1**
Crazy Sauce	20.3%	0.8424	0.6291	0.5423	0.5825
Cheddar Sauce	13.1%	0.8698	0.5094	0.1311	0.2085
Blueberry Sauce	9.0%	0.9111	0.5294	0.1268	0.2045
Garlic Sauce	9.4%	0.9041	0.4091	0.0612	0.1065
Spicy Sauce	4.9%	0.9524	0.6667	0.0263	0.0506
Extra Cheddar Sauce	0.3%	0.9968	0.0000	0.0000	0.0000
Tomato Sauce	2.7%	0.9733	0.0000	0.0000	0.0000
Pink Sauce	1.9%	0.9816	0.0000	0.0000	0.0000

Tabela 9: Performanța modelelor per sos

Observație: Sosurile rare (Extra Cheddar, Tomato, Pink) au F1=0 deoarece modelul nu poate învăța suficiente pattern-uri din puținele exemple pozitive.

4.5 Top Features pentru Fiecare Sos

Sos	Top 3 Features (pozitive)
Crazy Sauce	distinct_products (+0.59), has_Crazy Schnitzel (+0.56), has_Mac (+0.31)
Cheddar Sauce	distinct_products (+0.45), cart_size (+0.30), has_Breaded Chicken (+0.28)
Blueberry Sauce	distinct_products (+0.43), has_Viennese Schnitzel (+0.39), cart_size (+0.18)
Garlic Sauce	distinct_products (+0.50), cart_size (+0.37), has_Breaded Chicken (+0.21)
Spicy Sauce	distinct_products (+0.30), has_Viennese Schnitzel (+0.29), has_Pepsi (+0.13)

Tabela 10: Top features per sos

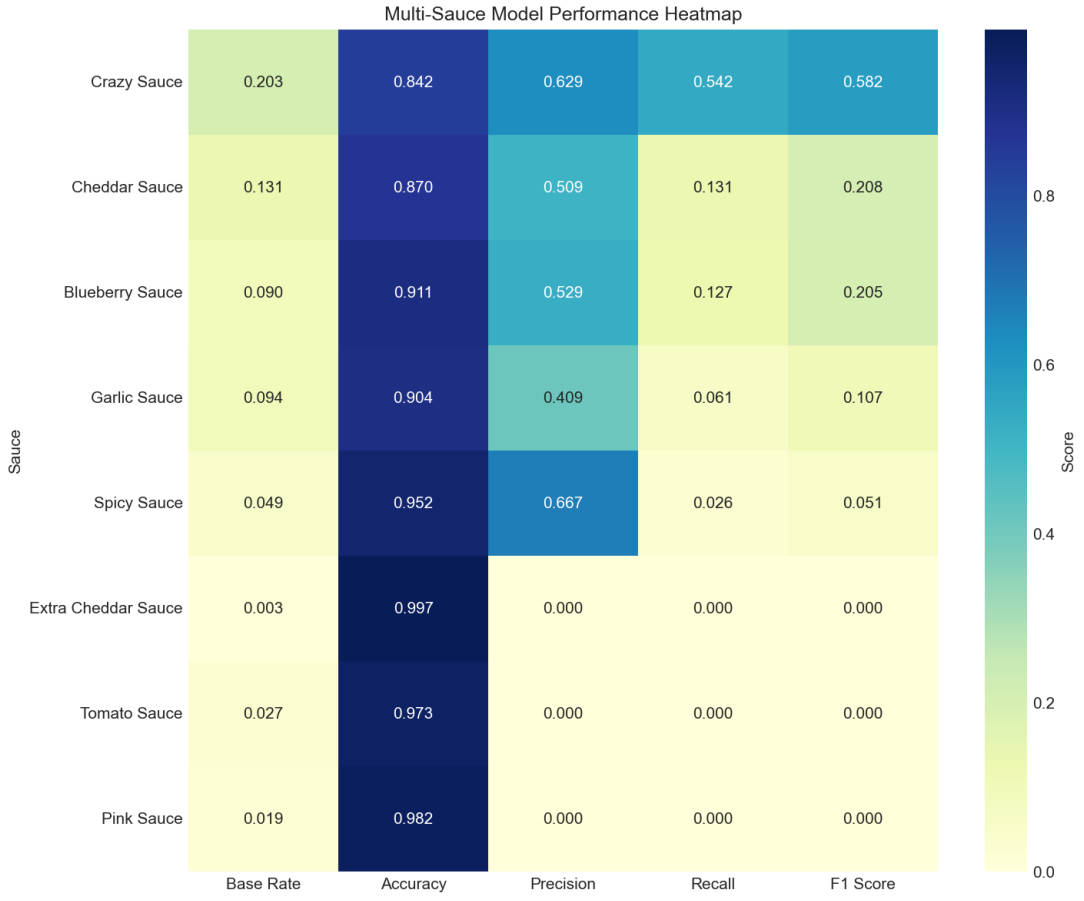

Figura 6: Heatmap performanță multi-sauce

5 Ranking pentru Upselling

5.1 Formularea Problemei

Scopul este de a construi o metodă care produce o ierarhie de produse candidate pentru upselling.

5.1.1 Produse Candidate

Am selectat 30 de produse pentru ranking: 8 sosuri, 11 băuturi și 11 garnituri.

5.1.2 Scor de Ranking

$$\text{Score}(p|\text{coș}) = P(p|\text{coș}) \times \text{price}(p) \tag{5}$$

Acest scor maximizează valoarea așteptată a vânzărilor (Expected Value).

5.2 Top Produse după Expected Value

Produs	Preț	P(cumparare)	Expected Value
Mac & cheease	32.9	28.4%	9.34 RON
Breaded Chicken Schnitzel	27.9	24.7%	6.90 RON
Crazy Schnitzel	28.9	22.7%	6.55 RON
Viennese Schnitzel	48.9	8.6%	4.19 RON
Mac & Cheese with Bacon	25.9	14.6%	3.77 RON

Tabela 11: Top 5 produse după Expected Value general

5.3 Upsell pentru Crazy Schnitzel

Pentru clienții care comandă Crazy Schnitzel, cele mai bune recomandări sunt:

Produs	P(condițională)	Expected Value
Mac & cheease	49.4%	16.24 RON
Baked potatoes	36.8%	4.75 RON
Pepsi Cola 0.25L	36.2%	4.35 RON
Crazy Sauce	53.2%	3.67 RON
Aqua Carpatica 0.5L	25.9%	2.46 RON

Tabela 12: Top upsell pentru Crazy Schnitzel

5.4 Algoritmi Implementați

5.4.1 Naive Bayes (from scratch)

Am implementat Gaussian Naive Bayes pentru estimarea $P(p|\text{coș})$:

$$P(C_k|x) = \frac{P(C_k)\prod_{i=1}^{n} P(x_i|C_k)}{P(x)} \tag{6}$$

unde presupunem că feature-urile sunt distribuite Gaussian:

$$P(x_i|C_k) = \frac{1}{\sqrt{2\pi\sigma_{ki}^2}} \exp\left(-\frac{(x_i - \mu_{ki})^2}{2\sigma_{ki}^2}\right) \tag{7}$$

5.4.2 k-Nearest Neighbors (from scratch)

Am implementat k-NN cu voting ponderat după distanță:

$$P(C|x) = \frac{\sum_{i \in N_k(x)} w_i \cdot \mathbb{1}[y_i = C]}{\sum_{i \in N_k(x)} w_i} \tag{8}$$

unde $w_i = \frac{1}{d(x,x_i)}$ pentru voting ponderat.

5.5 Experimental Setup

Pentru evaluare, pentru fiecare bon din test:

1. Construim un "coș parțial" eliminând 1 produs din cele 30 candidate
2. Folosim algoritmul de ranking pentru a genera Top-K recomandări
3. Verificăm dacă produsul eliminat apare în Top-K

Dataset: Training set: 6,295 bonuri; Test set: 1,574 bonuri (200 evaluate).

5.6 Rezultate

Algoritm	**Hit@1**	**Hit@3**	**Hit@5**	**MRR**
Naive Bayes (scratch)	0.0060	0.0302	0.0967	0.0307
k-NN (scratch)	0.0000	0.4125	0.5750	0.2104
Popularity Baseline	**0.2205**	**0.4441**	**0.6586**	**0.3635**
Revenue Baseline	0.2024	0.3263	0.6163	0.3177

Tabela 13: Performanța algoritmilor de ranking

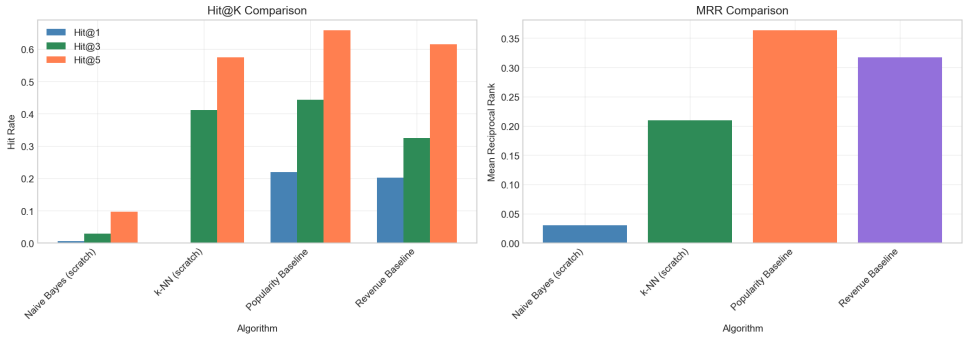

Figura 7: Comparație algoritmi de ranking

5.7 Exemple de Recomandări

Produse în coș	**Top 3 Naive Bayes**	**Hit?**
Breaded Chicken Schnitzel, Crazy Schnitzel, Extra parmezan, Aqua, Blueberry Sauce	Crazy Fries Cheddar bacon (25.9), Crazy Fries Parmesan (24.9), Baked potatoes (12.9)	✓
Viennese Schnitzel, Crazy Fries Parmesan, Extra bacon, Pepsi Zero, Blueberry Sauce	Crazy Fries Cheddar bacon (25.9), Crazy Fries Parmesan (24.9), Baked potatoes (12.9)	✓

Tabela 14: Exemple de recomandări cu Naive Bayes

5.8 Discuție Rezultate

De ce Popularity Baseline bate algoritmii from scratch?

- **Curse of dimensionality**: Cu 35 features și 6,000 exemple, algoritmii estimează prost probabilitățile
- **Class imbalance**: Produsele rare au puține exemple pozitive
- **Simplicitate**: Popularitatea globală e un semnal puternic în retail

6 Concluzii

6.1 Rezultate Principale

- **LR #1**: Modelul de Logistic Regression from scratch atinge un F1-score de 0.76 pentru predicția Crazy Sauce, depășind baseline-ul de majority class cu 16.5 puncte procentuale (acuratețe 69.75% vs 53.2%).
- **LR #2**: Sistemul de recomandare multi-sos funcționează bine pentru sosurile populare (Crazy Sauce F1=0.58), dar are dificultăți cu sosurile rare (Extra Cheddar, Tomato, Pink cu F1=0).
- **Ranking**: Popularity Baseline obține cele mai bune rezultate (Hit@5=0.66, MRR=0.36), depășind algoritmii from scratch. Acest rezultat e consistent cu literatura de specialitate în sisteme de recomandare.

6.2 Variante Încercate și Lecții Învățate

Ce nu a funcționat bine:

- **Learning rate prea mare (> 0.5)**: convergență instabilă, oscilații în loss
- **Fără regularizare L2**: overfitting pe setul de antrenare
- **Features de interacțiune**: am încercat produse carteziene (ex. Schnitzel $\times$ Drink), dar au crescut dimensionalitatea fără beneficiu semnificativ
- **One-hot encoding pentru ore**: mai puțin eficient decât ora ca feature numeric

Ce am învățat:

1. Feature-ul `distinct_products` este cel mai predictiv pentru toate sosurile
2. Există corelații puternice între tipul de schnitzel și sosul ales
3. Baseline-urile simple sunt competitivi în setări cu date limitate
4. Normalizarea feature-urilor (z-score) este esențială pentru convergența gradient descent

6.3 Direcții de Îmbunătățire

1. **Features suplimentare**: sezonalitate, time-series, cross-sell patterns
2. **Ensemble methods**: combinarea Naive Bayes + k-NN + Popularity
3. **Matrix Factorization**: pentru capturarea pattern-urilor latente
4. **Deep Learning**: rețele neuronale pentru embedding produse
5. **Cross-validation**: pentru optimizarea hiperparametrilor

7 Contribuții

- **Elisa Mercas**: Implementare Logistic Regression from scratch, preprocesare date, EDA, notebook-uri 01-03
- **Denis Munteanu**: Implementare Ranking (Naive Bayes, k-NN from scratch), evaluare Hit@K, notebook-uri 04-05, raport LaTeX

8 Anexe

8.1 Instrucțiuni de Rulare

```
# Install dependencies
pip install -r requirements.txt

# Run notebooks in order
jupyter notebook notebooks/

# Notebooks order: 01_eda -> 02_lr_crazy_sauce ->
#                  03_lr_multi_sauce -> 04_ranking_upsell -> 05
   _ranking_ml
```

8.2 Structura Repository

```
data/raw/           # Dataset (ap_dataset.csv)
data/processed/     # Preprocessed features
src/                # Source code
   data_loader.py
   preprocessing.py
   models/
       logistic_regression.py  # LR from scratch
       evaluation.py           # Metrics
       ranking.py              # NB + k-NN from scratch
notebooks/          # Jupyter notebooks (5 files)
results/            # Generated figures (24 PNG files)
report/             # LaTeX report
```

8.3 Figuri Suplimentare

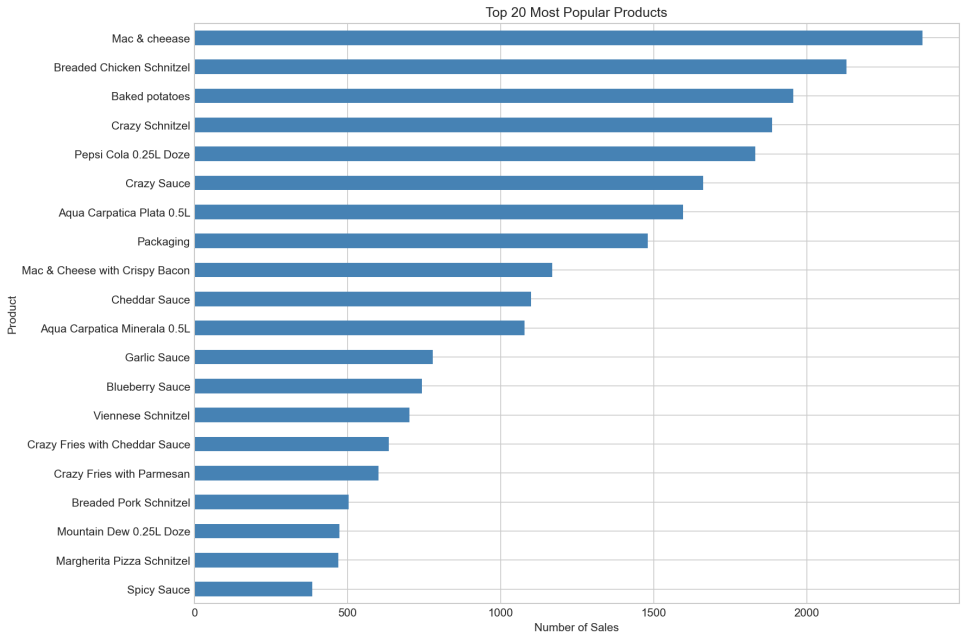

Figura 8: Top produse după vânzări

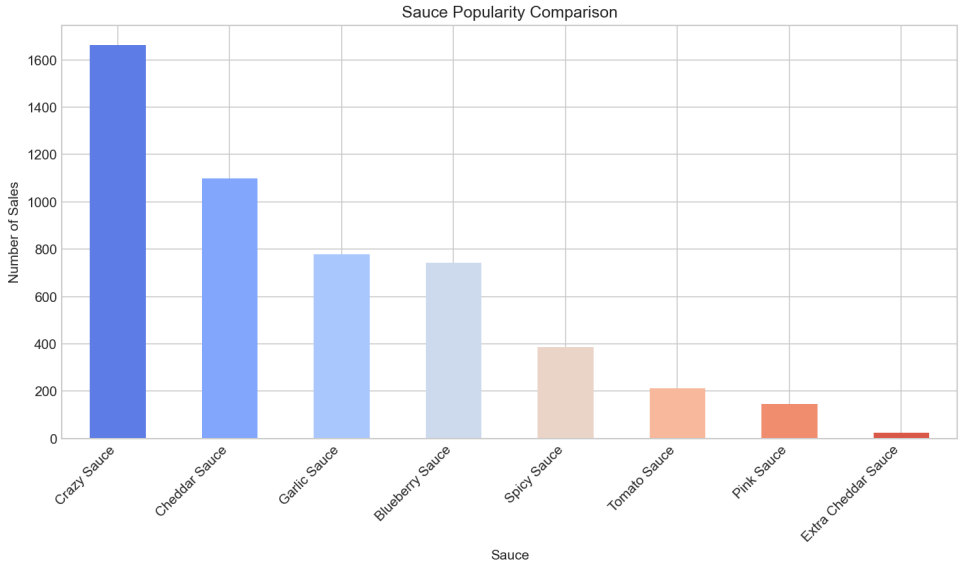

Figura 9: Popularitatea sosurilor

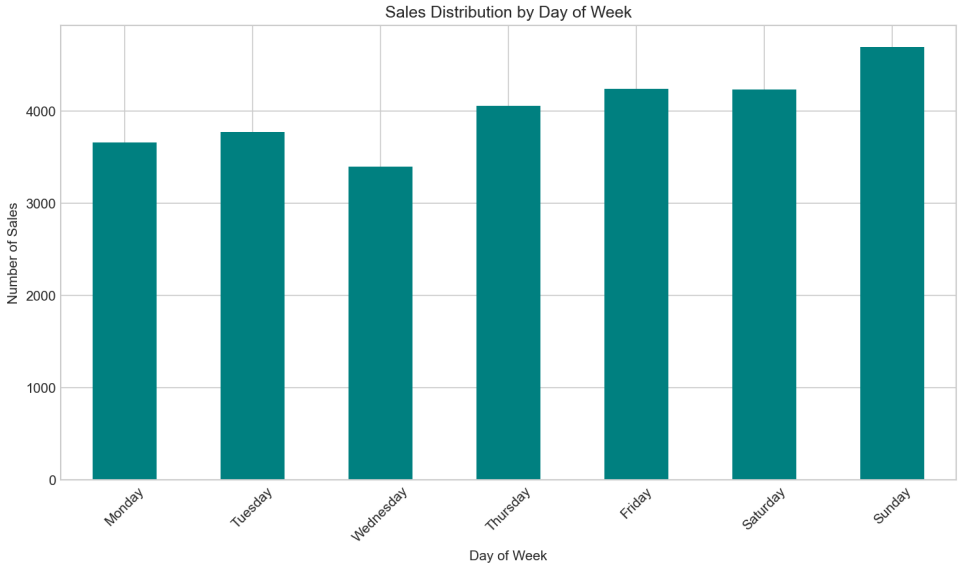

Figura 10: Vânzări pe zile