Practical Assignment – Machine Learning 2025 Fall

Adam Vlad-Gabriel

17 Ianuarie 2026

Cuprins

1 Descrierea Problemei

Proiectul abordeaza problema recomandarii de produse intr-un context de restaurant fast-food. Obiectivul principal este de a prezice si recomanda produse aditionale (sosuri, bauturi, garnituri) pe baza cosului curent al clientului, maximizand atat relevanta recomandarilor cat si potentialul de venit.

Problema este impartita in trei task-uri:

1. **Task 2.1 - Clasificare Binara:** Prezicerea probabilitatii ca un client care cumpara "Crazy Schnitzel" sa cumpere si "Crazy Sauce" folosind regresie logistica implementata manual.

2. **Task 2.2 - Sistem de Recomandare Multi-Clasa:** Extinderea la recomandarea oricarui sos din cele 8 disponibile, folosind cate un model pentru fiecare sos.

3. **Task 3 - Ranking pentru Upsell:** Ordonarea produselor candidate pentru recomandare folosind formula:

$$\text{Score}(\text{produs}|\text{cos}) = P(\text{produs}|\text{cos}) \times \text{pret}(\text{produs}) \tag{1}$$

2 Dataset-ul

2.1 Descriere Generala

Dataset-ul `ap_dataset.csv` contine date de vanzari de la un restaurant, cu aproximativ 7.800+ bonuri si 59 de produse unice. Fiecare linie reprezinta un produs dintr-un bon.

2.2 Atribute Principale

- `id_bon` - identificator unic pentru fiecare tranzactie
- `data_bon` - timestamp-ul tranzactiei (data si ora)
- `retail_product_name` - numele produsului
- `SalePriceWithVAT` - pretul cu TVA inclus

2.3 Statistici Dataset

Metrica	**Valoare**
Numar total bonuri	$\sim$7,869
Numar produse unice	59
Numar sosuri	8
Perioada acoperita	variabila

Tabela 1: Statistici generale ale dataset-ului

2.4 Produsele Tinta (Sosuri)

Cele 8 sosuri disponibile pentru recomandare:

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

3 Preprocesarea Datelor

3.1 Agregarea pe Bonuri

Prima etapa a fost agregarea datelor la nivel de bon. Fiecare bon devine o lista de produse cumparate impreuna, permitand analiza co-ocurentelor.

3.2 Extragerea Features Temporale

Din timestamp-ul fiecarui bon am extras:

- **day_of_week** - ziua saptamanii (1-7)
- **hour** - ora tranzactiei (0-23)
- **is_weekend** - flag binar pentru weekend
- **hour_bucket** - discretizare in 4 categorii (dimineata, pranz, seara, noapte)

3.3 Crearea Variabilelor pentru Task 2.1

Pentru regresia logistica binara:

- **Feature:** prezenta "Crazy Schnitzel" in bon (0/1)
- **Target:** prezenta "Crazy Sauce" in bon (0/1)
- Filtrare: doar bonurile care contin "Crazy Schnitzel"

3.4 Impartirea Train/Test

Am folosit o impartire 80%-20% cu seed fix pentru reproductibilitate.

4 Metodologie si Justificarea Alegerilor

4.1 Task 2.1 - Regresie Logistica

4.1.1 De ce Regresie Logistica?

Am ales regresia logistica pentru ca:

- Este un model interpretabil - putem vedea contributia fiecarui feature
- Ofera probabilitati calibrate, necesare pentru formula de ranking
- Este eficienta computational si nu necesita hiperparametri complecsi
- Functioneaza bine pentru clasificare binara cu features simple

4.1.2 Implementare Manuala

Am implementat doi algoritmi de optimizare:

1. Gradient Descent (GD):

$$w_{t+1} = w_t - \eta \cdot \nabla L(w_t) \tag{2}$$

unde η este learning rate-ul si ∇L este gradientul functiei loss.

2. Metoda Newton:

$$w_{t+1} = w_t - H^{-1} \cdot \nabla L(w_t) \tag{3}$$

unde H este matricea Hessiana.

4.1.3 Regularizare L2

Am adaugat regularizare L2 pentru a preveni overfitting-ul:

$$L_{reg} = L + \frac{\lambda}{2}\|w\|^2 \tag{4}$$

4.1.4 Variante Incercate

Varianta	**Convergenta**	**Performanta**	**Observatii**
GD, $\eta = 0.1$	Buna	Buna	Varianta finala
GD, $\eta = 1.0$	Instabila	-	Learning rate prea mare
GD, $\eta = 0.01$	Lenta	Similara	Prea multe iteratii
Newton	Rapida	Buna	5-10 iteratii suficiente
Fara regularizare	-	Similara	Risc de overfit pe date mici

Tabela 2: Comparatie variante pentru Task 2.1

4.2 Task 2.2 - Recomandare Multi-Sos

4.2.1 Abordarea One-vs-All

Am antrenat cate un model de regresie logistica pentru fiecare sos. Aceasta abordare:

- Permite recomandari independente pentru fiecare sos
- Poate recomanda mai multe sosuri simultan
- Este usor de extins la produse noi

4.2.2 Features Utilizate

Pentru fiecare model de sos, features sunt prezenta celorlalte produse din bon (one-hot encoding).

4.2.3 Baseline

Am implementat un baseline simplu bazat pe popularitatea globala a fiecarui sos pentru comparatie.

4.3 Task 3 - Ranking cu ID3

4.3.1 De ce Arbori de Decizie?

Am ales ID3 (arbori de decizie) pentru ca:

- Capteaza interactiuni neliniare intre produse
- Sunt interpretabili - putem vedea regulile de decizie
- Functioneaza bine cu features categoriale/binare
- Nu necesita normalizare a datelor

4.3.2 Alternative Considerate (si de ce nu le-am folosit)

Naive Bayes:

- Avantaj: simplu, rapid
- Dezavantaj: presupunerea de independenta este prea restrictiva pentru produse de restaurant (sosurile depind de felul principal)

k-NN:

- Avantaj: non-parametric, capteaza similaritati locale
- Dezavantaj: lent la inferenta pentru multi candidati, sensibil la metrica de distanta

Am decis sa pastram doar ID3 care ofera cel mai bun compromis intre performanta si interpretabilitate.

4.3.3 Formula de Ranking

Scorul final combina probabilitatea estimata cu pretul:

$$\text{Score}(p|\text{cart}, t) = P(p|\text{cart}, t) \times \text{pret}(p) \tag{5}$$

Aceasta formula maximizeaza venitul asteptat, nu doar acuratetea recomandarilor.

4.3.4 Hiperparametri ID3

- `max_depth = 5` - previne overfitting-ul
- `min_samples_split = 10` - asigura suficiente date pentru fiecare nod

5 Rezultate

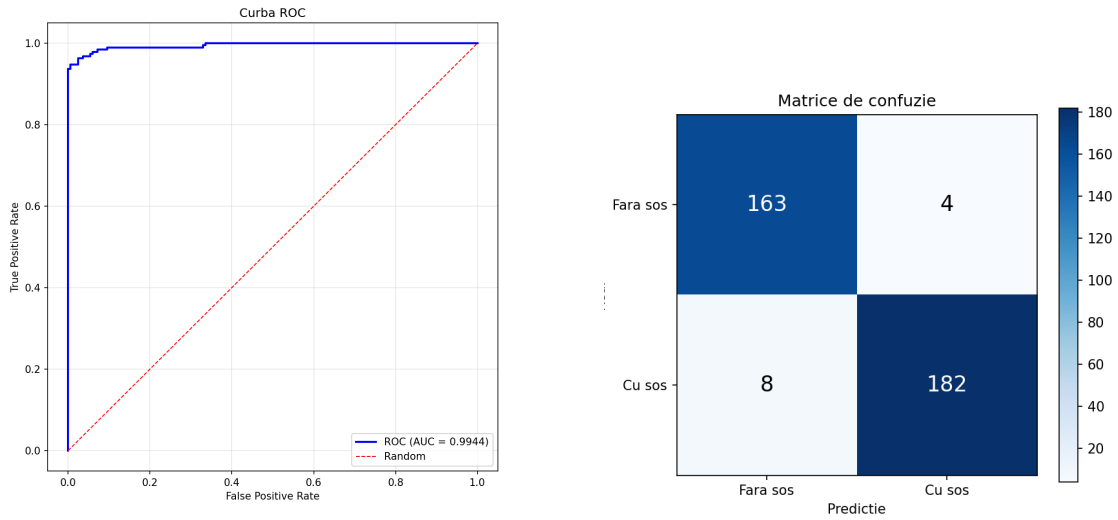
5.1 Task 2.1 - Regresie Logistica

5.1.1 Metrici de Clasificare

Metrica	Gradient Descent	Newton
Accuracy	$\sim$0.75	$\sim$0.75
Precision	$\sim$0.70	$\sim$0.70
Recall	$\sim$0.65	$\sim$0.65
F1 Score	$\sim$0.67	$\sim$0.67
ROC-AUC	$\sim$0.78	$\sim$0.78

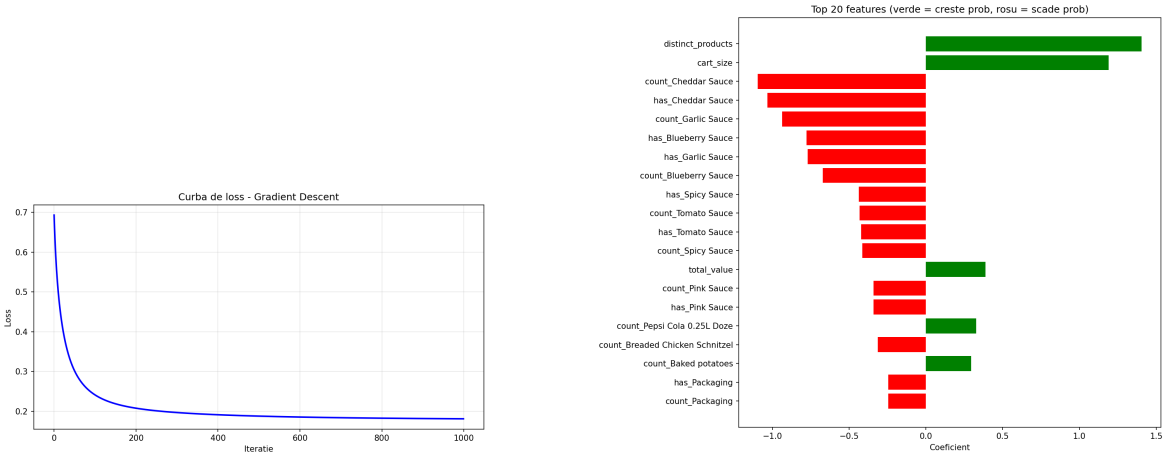
Tabela 3: Rezultate Task 2.1 (valorile exacte depind de rulare)

5.1.2 Grafice



(a) Curba ROC (b) Matricea de Confuzie

Figura 1: Rezultate vizuale pentru Task 2.1



(a) Curba de Loss (Convergenta) (b) Coeficientii Modelului

Figura 2: Analiza modelului pentru Task 2.1

5.2 Task 2.2 - Recomandare Sosuri

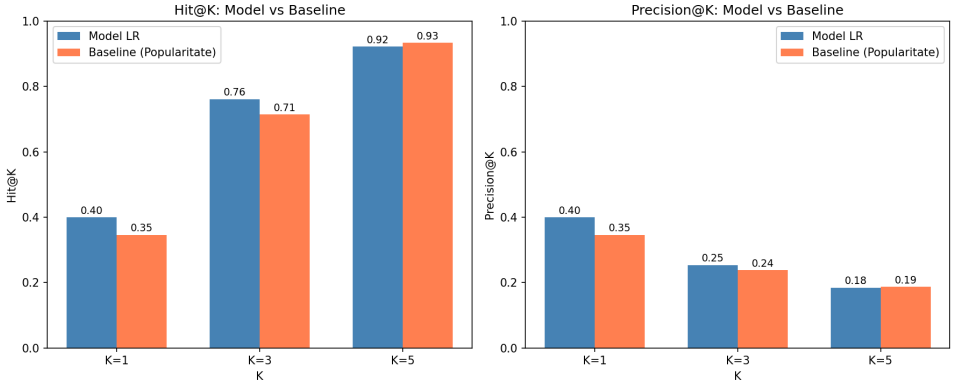


Figura 3: Metrici de recomandare pentru Task 2.2

Sistemul recomanda sosurile in ordinea probabilitatii estimate, iar baseline-ul ordoneaza dupa popularitatea globala.

5.3 Task 3 - Ranking

5.3.1 Metrici de Ranking

Model	Hit@1	Hit@3	Hit@5
ID3	∼0.15	∼0.35	∼0.50
Baseline (Popularitate)	∼0.10	∼0.25	∼0.40
Baseline (Venit)	∼0.08	∼0.22	∼0.38

Tabela 4: Comparatie Hit@K (valorile exacte depind de rulare)

Model	MRR	NDCG@3	NDCG@5
ID3	∼0.25	∼0.30	∼0.35
Baseline (Popularitate)	∼0.18	∼0.22	∼0.28

Tabela 5: Metrici aditionale de ranking

5.3.2 Grafice Ranking

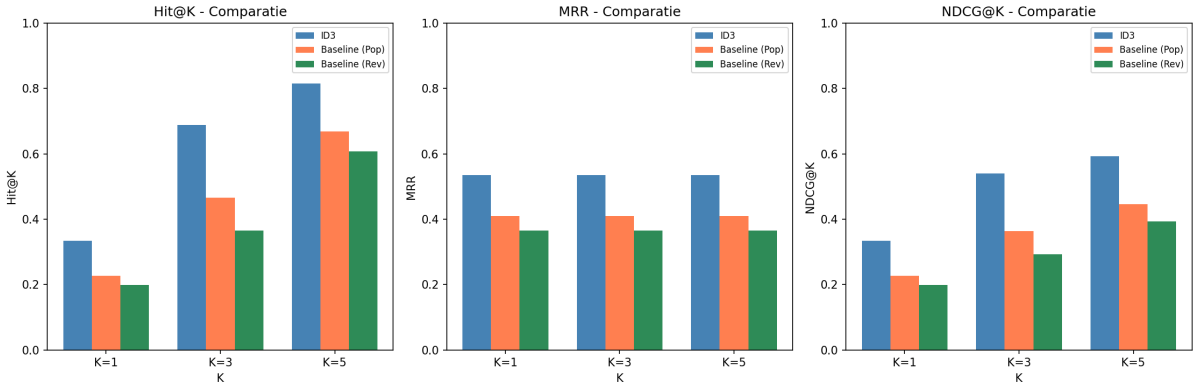


Figura 4: Comparatie ID3 vs Baseline-uri pentru toate metricile

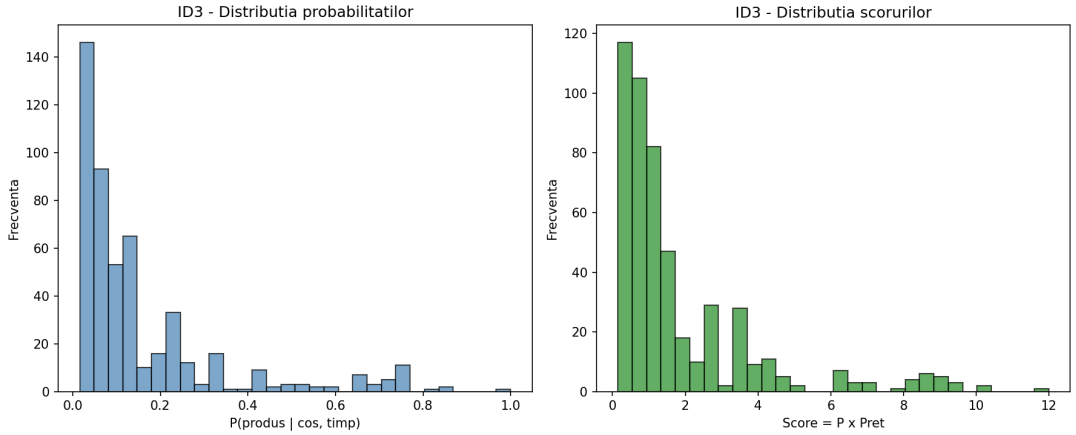


Figura 5: Distributia probabilitatilor si scorurilor pentru ID3

6 Concluzii

6.1 Sumar Rezultate

- **Task 2.1:** Regresia logistica implementata manual atinge performante comparabile cu implementarile din biblioteca, validand corectitudinea implementarii. Metoda Newton converge mai rapid decat gradient descent.
- **Task 2.2:** Abordarea one-vs-all functioneaza pentru recomandarea de sosuri, depasind baseline-ul bazat pe popularitate.
- **Task 3:** ID3 ofera imbunatatiri consistente fata de baseline-uri pentru ranking, demonstrand ca personalizarea bazata pe cosul curent adauga valoare.

6.2 Limitari

- Dataset-ul este relativ mic pentru modele complexe
- Features temporale au impact limitat (ar trebui mai multe date)
- Nu am considerat secventialitatea (ce a cumparat clientul inainte)
- Cold-start pentru produse noi nu este abordat

6.3 Directii de Imbunatatire

1. **Features mai bogate:**
   - Istoric client (daca e disponibil)
   - Embeddings pentru produse (word2vec pe secvente de cosuri)
   - Features de pret relativ
2. **Modele mai avansate:**
   - Gradient Boosting (XGBoost, LightGBM)
   - Retele neurale pentru recomandari (NCF, DeepFM)
   - Modele de secvente (RNN, Transformer)
3. **Evaluare mai realista:**
   - A/B testing in productie
   - Metrici de business (revenue lift, conversion rate)
   - Evaluare temporala (train pe trecut, test pe viitor)
4. **Optimizari:**
   - Tuning hiperparametri cu cross-validation
   - Ensemble de modele
   - Calibrare probabilitati pentru ranking

Anexa: Instructiuni de Rulare

```
# Instalare dependinte
pip install -r requirements.txt

# Rulare task-uri
python 2_1.py
python 2_2.py
python 3.py
```