Raport AP2

Lăcătuș Stefania

17 Ianuarie 2025

Introducere

Problema dată cere să se dezvolte o soluție folosind algoritmii din cadrul paradigmei de învățare supervizată pentru a prezice o ierarhie a celor 6 categorii de activități din coloana "Category" a setului de date pentru una dintre țările disponibile.

Pentru a rezolva această problemă, am testat performanța de a prezice această ierarhie și valoarea de "Revenue" pentru o țară dată a trei algoritmi învățați în cadrul materiei Învățare Automată: k-NN, ID3 și AdaBoost, dar și alți trei algoritmi care nu au fost prezentați la curs: Regresie Liniară, Gradient Boosting și Random Forest.

Testarea performanței

Am realizat această testare a performanței cu scopul de a identifica algoritmul cu acuratețea cea mai mare, în scopul de a-l selecta și a-l implementa. În cadrul acestei etape am folosit implementările algoritmilor din librăria scikit-learn.

k-NN

Algoritmul K-Nearest Neighbors oferă flexibilitate, dar poate suferi în prezența unui set de date mare și necesită optimizarea numărului de vecini pentru a obține rezultate optime.

În cadrul acestei testări a obținut rezultatele:

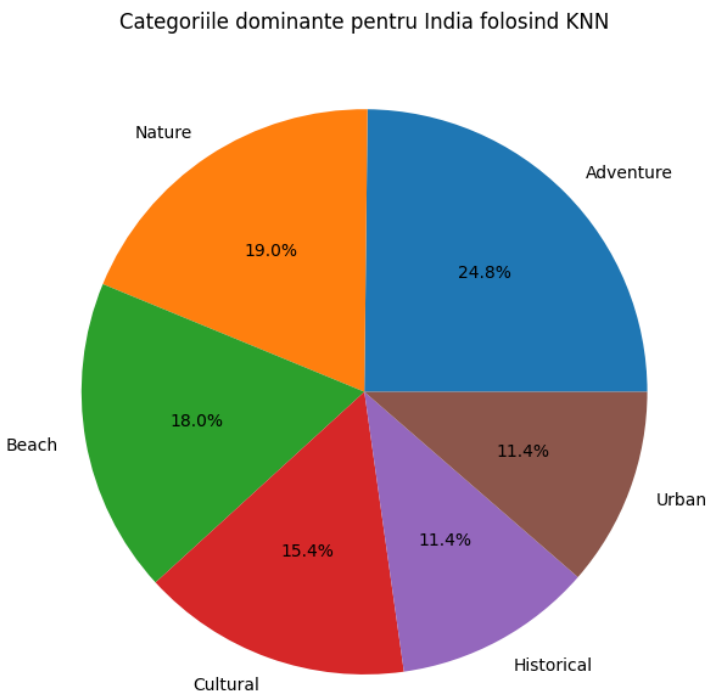

Figure 1: Pie Chart predicții - k-NN

```
Top Categories (KNN) for India:
1. Adventure - Revenue/Profit: 704622.66
2. Nature - Revenue/Profit: 537908.02
3. Beach - Revenue/Profit: 511164.09
4. Cultural - Revenue/Profit: 438535.63
5. Historical - Revenue/Profit: 323099.13
6. Urban - Revenue/Profit: 323099.13
KNN Accuracy: 78.87%
```

Figure 2: Rezultate și acuratețe - k-NN

Algoritmul a avut o acuratețe de `78.87%`, dar a produs rezultate mai puțin robuste, sugerând că nu este ideal pentru această problemă.

ID3

Algoritmul ID3 este simplu și eficient în a împărți datele, însă este predispus la overfitting.

Rezultatele acestuia în cadrul testării:

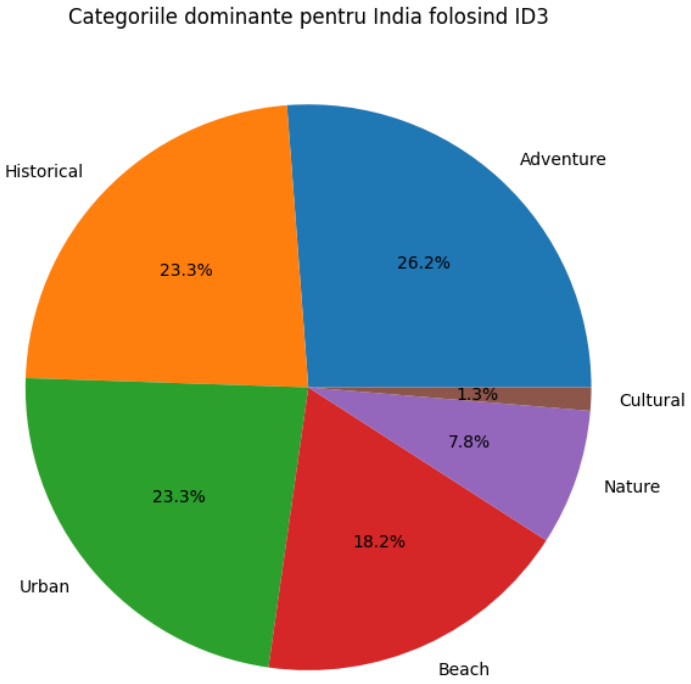

Figure 3: Pie chart predicții - ID3

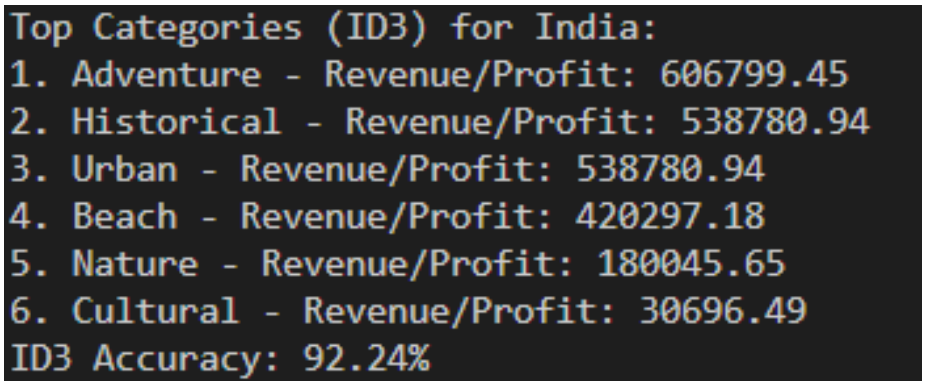

Figure 4: Rezultate și acuratețe - ID3

ID3 au oferit o performanță de `92.24%`, însă distribuția categoriilor în ierarhie a variat considerabil față de celelalte metode.

AdaBoost

AdaBoost combină clasificatori simpli pentru a obține un model puternic, fiind ideală pentru problemele în care erorile mici de predicție sunt penalizate sever.

Rezultatele acestuia în cadrul testării:

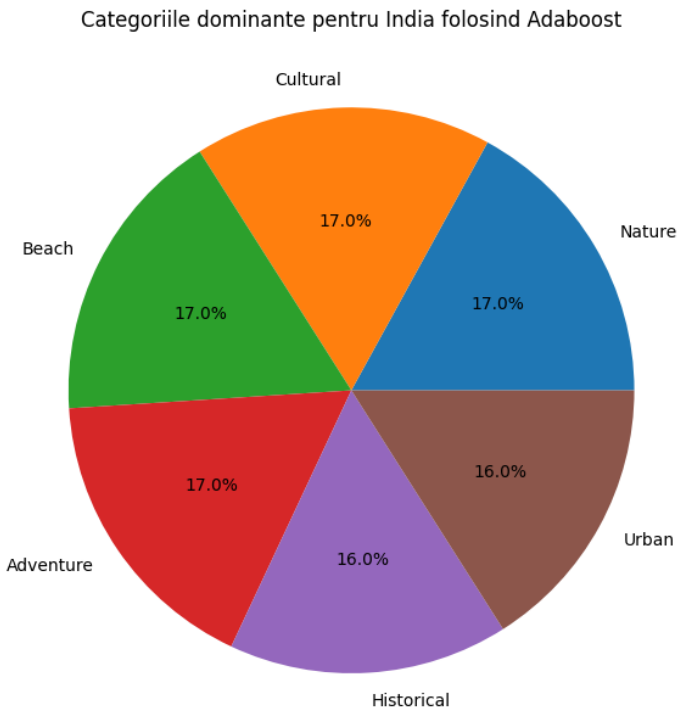

Figure 5: Pie chart predicții - AdaBoost

```
Top Categories (Adaboost) for India:
1. Nature - Revenue/Profit: 516011.42
2. Cultural - Revenue/Profit: 516011.42
3. Beach - Revenue/Profit: 516011.42
4. Adventure - Revenue/Profit: 516011.42
5. Historical - Revenue/Profit: 485194.08
6. Urban - Revenue/Profit: 485194.08
Adaboost Accuracy: 99.16%
```

Figure 6: Rezultate și acurateță - AdaBoost

Metoda a obținut o acuratețe ridicată (`99.16%`), combinând avantajele arborilor de decizie cu robustețea metodelor ensemble. Ierarhia a fost dominată de categoriile *Nature*, *Cultural*, *Beach* și *Adventure*.

Regresie Liniară, Gradient Boosting, Random Forest

Regresia Liniară a oferit o predicție aproape perfectă (`Acuratețe: 99.71%`) datorită naturii sale simple și a corelației puternice între caracteristici. Categoriile de top pentru `India` sunt: *Nature*, *Adventure*, și *Cultural*.

Random Forest a obținut un `Acuratețe: 80.80%`. Deși nu este printre cele mai bune rezultate, metoda a arătat o variabilitate mai mare în predicții, ceea ce indică sensibilitate la distribuția caracteristicilor. Categoria de top identificată a fost *Adventure*, urmată de *Urban* și *Beach*.

Gradient Boosting a avut o acuratețe de `98.74%` și s-a apropiat de performanța regresiei liniare, însă cu o ierarhie diferită: *Adventure*, *Cultural*, și *Beach*.

Concluzie

Dintre algoritmii analizați, **Regresia Liniară** și **AdaBoost** s-au remarcat ca având acuratețea cea mai mare. AdaBoost a demonstrat o adaptabilitate mai mare la complexitatea relațiilor din date și am ales să îl implementez, datorită echilibrului între precizie și robustețe. Alegerea se justifică prin capacitatea sa de a combina clasificatori simpli pentru a obține o performanță optimă în maximizarea profitului pentru ierarhizarea categoriilor de activități.

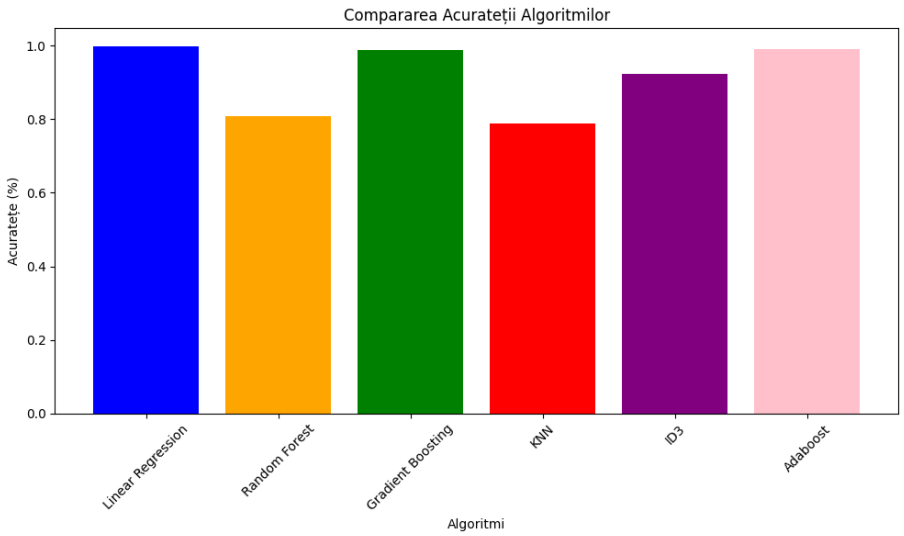

Figure 7: Grafic comparativ

Implementarea algoritmului AdaBoost

Am implementat acest algoritm utilizând un *base learner*, `DecisionTreeRegressor`, și un proces iterativ care ajustează greutățile eșantioanelor pentru a se concentra pe exemplele greu de prezis.

Pentru antrenare și testare, am împărțit setul de date astfel: `80%` din date pentru antrenare și `20%` din date pentru testare.

Performanța sa a fost evaluată folosind Mean Absolute Error (MAE) și acuratețea predicției.

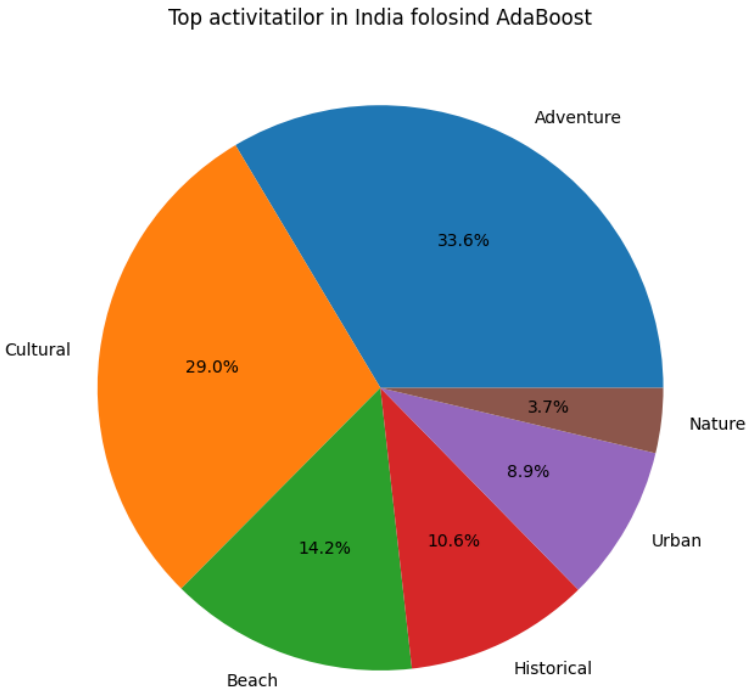

Figure 8: Pie chart predicții - AdaBoost Implementat

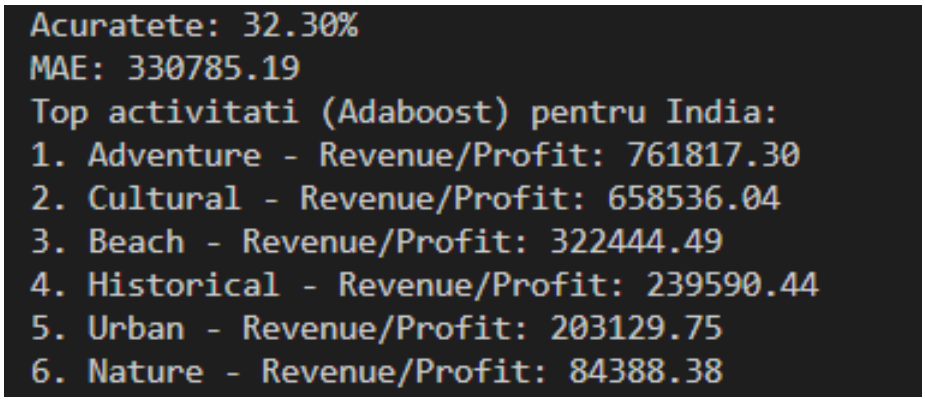

Figure 9: Rezultate și acurateț̦e + MAE - AdaBoost Implementat

Concluzie

Implementarea algoritmului AdaBoost a avut o acurateț̦e mai scăzută (32.30%) comparativ cu implementarea din `scikit-learn` (99.16%).

Această discrepanț̦ă evidenț̦iază posibile probleme în implementare, cum ar fi ajustarea incorectă a greutăț̦ilor sau gestionarea insuficientă a erorilor de predicț̦ie. Pe lângă asta, clasificările profitului pentru categorii au variat semnificativ între cele două implementări, sugerând diferenț̦e în modul de interpretare a setului de date.