# Predictia venitului unei persoane

Barbu Andrei-Catalin, Caragea Matei-Ioan

25 noiembrie 2025

## 1 Descrierea implementarii

### 1.1 Preprocesarea setului de date

Implementarea clasificarii veniturilor a fost realizata cu ajutorul bibliotecilor Pandas si NumPy pentru manipularea datelor. Am inceput cu proprocesarea datelor din setul de date UCI Adult Income, unde valorile lipsa au fost inlocuite cu valorile cele mai intalnite in atributul respectiv.

In continuare am aplicat o transformare pe coloanele capital-gain si capital-loss, deoarece am observat faptul ca majoritatea valorilor din aceste coloane erau 0, iar restul puteau avea valori foarte mari. Pentru a normaliza distributia si a reduce impactul valorilor extreme, am aplicat o transformare logaritmica (log1p).

Urmatorul pas a fost simplificarea feature-urilor. Am observat ca in coloana native-country valoarea 'United-States' era predomintanta, asa ca am modificat feature-ul astfel incat sa aiba valoarea 1 pentru United-States si 0 pentru orice alta tara. Atributele income si sex au fost si ele modificate astfel incat sa contina valorile 0 si 1 pentru '$<=$ 50K' si '$>$ 50K', respectiv 'female' si 'male'.

In continuare am eliminat coloana education, care continea titlul celui mai inalt nivel de educatie pe care il are fiecare persoana. Aceasta coloana era strans legata de education-num, care reprezinta numarul de ani de educatie, asa ca importanta ei a devenit nesemnificativa.

Pentru atributele categoriale am folosit One-Hot Encoding. Acesta transforma categoriile in vectori binari pentru fiecare categorie. Metoda a fost aleasa din cauza situatiei in care modelul ar fi putut intelege gresit valorile categoriile in cazul in care le codificam numeric (Wife=1, Own-Child=2, Husband=3 etc.).

Urmatoarea problema rezolvata este cea a scalarii. Fara scalare, modelul ar putea intelege ca diferenta de 1000 de unitati la capital-gain este mult mai importanta decat o diferenta de 10 ani la age, doar din cauza numarului mai mare. Scalarea aduce toate variabilele la acelasi nivel. In proiectul nostru am folosit StandardScaler, o tehnica de normalizare care transforma datele astfel incat sa aiba o medie egala cu 0 si o deviatie standard egala cu 1.

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

Datele au fost in final impartite in 80% date de antrenare, respectiv 20% date de testare.

## 1.2 Modelul ales

Pentru predictia venitului unei persoane am ales modelul Random Forest. Acesta ne ajuta la reducerea overfitting-ului, gestionarea datelor non-liniare si este putin influentat de valorile extreme comparativ cu modelele liniare.

Hiperparametrii importanti sunt:

1. n_estimators=100: echilibru intre stabilitate si cost
2. max_depth=15: limitarea adancimii arborilor pentru a nu invata fiecare detaliu nesemnificativ din date
3. class_weight='balanced': deoarece avem putin oameni cu venit mare, am ales ca modelul sa fie mai atent la erorile pe aceasta clasa, penalizandu-le mai drastic

## 1.3 Rezultate

Pentru a evalua nu doar performanta modelului, ci si comportamentul sau etic, am utilizat biblioteca Fairlearn[BDE+20]. Ne-am bazat pe utilizarea clasei MetricFrame, care ne-a ajutat sa calculam urmatorii indicatori pentru fiecare grup in parte:

1. acuratete: masoara procentul total de predictii corecte
2. rata de selectie: indica proportia instantelor pentru care modelul a prezis clasa pozitiva (venit $>$ 50K). O diferenta mare intre grupuri indica o sub-reprezentare a unui grup in categoria veniturilor mari
3. rata fals-negativa: reprezinta proportia persoanelor care au in realitate venituri mari, dar sunt clasificate gresit de model ca avand venituri mici
4. rata fals-pozitiva: reprezinta proportia persoanelor cu venituri mici care sunt clasificate gresit ca avand venituri mari

Rezultatele obtinute sunt:

Tabela 1: Analiza performantei modelului pe atribute sensibile)

| **grup demografic** | **acuratete** | **rata de selectie** | **rata fals-negativa** | **rata fals-pozitiva** |
|---|---|---|---|---|
| **Analiza pe sex** | | | | |
| Femei (0) | 0.912 | 0.131 | 0.300 | 0.061 |
| Barbati (1) | 0.761 | 0.487 | 0.089 | 0.303 |
| **Analiza pe rasa** | | | | |
| Non-Black (False) | 0.803 | 0.389 | 0.117 | 0.223 |
| Black (True) | 0.884 | 0.188 | 0.213 | 0.101 |

Analizand datele obtinute, observam bias-uri semnificative preluate de model. Exista o discrepanta in tratamentul femeilor comparativ cu barbatii. Modelul prezice venituri mari pentru barbati de aproape 4 ori mai des decat pentru femei. Dintre femeile care au in realitate venituri mari, modelul nu le identifica in 30% din cazuri, in timp ce la barbati rata fals-negativa este de aproximativ 9%. Desi acuratetea este mai mare la femei, aceasta se datoreaza faptului ca modelul prezice corect majoritatea veniturilor mici, dar esueaza pe veniturile mari.

In ceea ce priveste rasa, modelul prezinta tendinte similare. Rata fals-negativa este aproape dubla pentru persoanele de culoare, comparativ cu restul populatiei. Modelul tinde sa subestimeze veniturile persoanelor de culoare, clasificandu-le eronat in categoria cu venituri mici chiar si atunci cand nu este asa.

# 2 Probleme intalnite

Printre problemele pe care le-am intalnit in setul de date sunt urmatoarele:

1. date lipsa: desi la prima vedere setul parea complet, acesta avea date lipsa marcate cu ?. Daca le-am fi ignorat, modelul le-ar fi tratat ca pe o categorie reala, iar daca le-am fi sters, am fi ramas fara aproximativ 3000 de inregistrari in setul de date. Solutia aleasa a fost sa le inlocuim cu modul coloanei.

2. dezechilibrul claselor: distributia venitului este de 76% pentru persoane cu venit mic, respectiv 24% pentru venit mare. Din cauza asta modelul ar putea avea tendinta sa prezica mereu clasa majoritara pentru a obtine o acuratete mai mare. Totusi, folosirea parametrului class_weight='balanced' reduce aceasta posibilitate.

3. distributia extrema: variabilele financiare capital-gain si capital-loss au o distributie ciudata. Marea majoritate a populatiei are valoarea 0, iar un procent mic are valori foarte mari. Pentru aceasta problema am folosit o transformare logaritmica astfel incat sa putem normaliza corect distributia

# 3 Activitati urmatoare

Analiza a demonstrat ca modelul actual, desi performant din punct de vedere al acuratetii, incalca principiile de echitate. In continuare vom utiliza algoritmul Exponentiated Gradient Reduction, disponibil in biblioteca Fairlearn. Acest algoritm antreneaza o secventa de predictori ponderati, incercand sa minimizeze eroarea de clasificare, facand mai importanta echitatea in detrimentul acuratetei.

Pentru a aborda problema dezechilibrului de clase, vom utiliza tehnica SMOTE (Syntethic Minority Over-sampling Technique). Aceasta presupune modificarea setului de antrenament prin generarea de date artificiale, inainte ca acestea sa ajunga la model. Algoritmul selecteaza o instanta din clasa minoritara, gaseste cei mai apropiati vecini, traseaza o linie imaginara intre instanta selectata si unui dintre vecini si genereaza un nou punct pe acea linie. In acest mod obtinem date artificiale care nu exista in realitate, dar care au caracteristici matematice coerente cu profilul persoanelor deja existente. [CBHK02]

# Bibliografie

[BDE+20] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. Fairlearn: A toolkit for assessing and improving fairness in ai. 2020.

[CBHK02] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.