Analiza predictivă a riscului de credit

Descrierea proiectului

Proiectul implementează o analiză completă de prelucrare avansată a datelor și compararea modelelor de machine learning pentru predicția riscului de credit. Obiectivul principal este să identifice persoanele cu risc ridicat de întârziere la plăți folosind date prelucate de colegul de echipa.

Dataset

Structura datelor

Dataset-ul conține **16,714 înregistrări** cu **9 caracteristici** pentru fiecare persoană:

Nume	Tip	Descriere
rev_util	float	Utilizarea limitei cardului (procentuală)
age	int	Vârsta persoanei
debt_ratio	float	Venit dedicat datoriilor (procentual)
monthly_inc	int	Salariul lunar
open_credit	int	Numărul de credite active
late	string	Valoare categorială (none, low, medium, severe)
inc_per_dep	float	Venit per membrii familiei
relationship_status	string	Starea relației (single, couple, family, etc.)
dlq_2yrs	int	**Variabila țintă**: 0/1 - întârziere gravă în ultimii 2 ani

Observații inițiale

- Nu există valori lipsă în dataset
- Distribuții asimetrice pentru majoritatea variabilelor numerice
- Dezechilibru în datele categorice (majoritatea persoanelor nu au întârzieri)
- Prezența outlier-ilor extremi în variabilele financiare

Prelucrarea datelor

1. Gestionarea outlier-ilor

- **Metoda IQR**: Eliminarea valorilor extreme folosind iqr
- **Percentile clipping**: Aplicarea **99%** pentru variabilele cu distribuții foarte asimetrice

2. Normalizare și standardizare

- **Normalizare [0,1]**: Pentru variabilele procentuale (rev_util, debt_ratio, inc_per_dep)
- **Standardizare**: Pentru monthly_inc (media ≈ 0, deviația standard = 1)

3. Encodarea variabilelor categorice

- late: Mapare ordinală (none=0, low=1, medium=2, severe=3)
- relationship_status: Mapare numerică (single=1, couple=2, family=3, etc.)

Analiza exploratorie (EDA)

Distribuții

- **Variabile numerice**: Histograme cu densidade și boxplot-uri pentru identificarea asimetriei
- **Variabile categorice**: Grafice de frecvență pentru înțelegerea dezechilibrului claselor

Corelații

- **Corelația cea mai puternică** cu variabila țintă: rev_util (utilizarea cardului)
- **Alte corelații semnificative**: late (0.5), inc_per_dep, monthly_inc
- Heatmap pentru vizualizarea relațiilor între toate variabilele

Analiză comparative (Violin plots)

- Persoanele cu întârzieri grave au utilizare mai mare a limitei de credit
- Venitul mai mic corelează cu probabilitatea mai mare de întârziere
- Vârsta de ~40 ani prezintă risc crescut

Modele implementate

Având în vedere natura binarului problemei de clasificare, au fost testati 4 algoritmi:

1. Logistic Regression

- **Parametri**: max_iter=3000
- **Acuratețe**: 76.38%

2. Decision Tree Classifier

- **Parametri**: max_depth=5, min_samples_split=2
- **Acuratețe**: 75.72%

3. Random Forest Classifier

- **Parametri**: n_estimators=300, max_depth=7, min_samples_split=2
- **Acuratețe**: 76.74% ➜ **Cel mai bun**

4. Support Vector Machine (SVM)

- **Parametri**: kernel='rbf', C=1.0, gamma=0.1
- **Acuratețe**: 76.09%
- **Preprocessare**: StandardScaler pentru scalarea caracteristicilor

Rezultate și evaluare

Tabel comparativ performanță

Model	Acuratețe
Random Forest	**76.74%**
Logistic Regression	76.38%
SVM	76.09%
Decision Tree	75.72%

Concluzii

1 Modelul optim: Random Forest obține cea mai bună performanță (76.74%)

2 Diferențe mici: Toate modelele au performanțe similare, sugerând limitări în separabilitatea claselor

3 Factori predictivi cheie:

* Utilizarea limitei de credit (rev_util)
* Istoricul întârzierilor (late)
* Venitul per dependent (inc_per_dep)

4 Provocări identificate: Dezechilibrul claselor și limitările inherente în dataset

Tehnologii utilizate

- **Python 3.13**
- **Pandas** - manipularea datelor
- **NumPy** - operații numerice
- **Matplotlib & Seaborn** - vizualizare
- **Scikit-learn** - modele de machine learning

- **Jupyter Notebook** - mediul de dezvoltare

Structura fișierelor

```
### ├── train.csv              # Dataset de antrenare
### ├── test.csv               # Dataset de testare
### ├── tema.ipynb             # Notebook principal cu analiza
### └── README.md              # Documentația proiectului
```