Detectarea fraudelor cu carduri de credit

Kaggle Credit Card Fraud Detection Dataset[1]

The dataset contains transactions made by credit cards in September 2013 by European cardholders.

This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, ... V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

[1] https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/

1. Încărcați setul de date din fișierul `creditcard.csv`.

Afișați numărul de instanțe și numărul de atribute ale setului de date.

```
(284807, 31)
```

2. Afișați 10 instanțe din setul de date, selectate în mod aleatoriu.

```
            Time         V1        V2  ...       V28  Amount  Class
64358    51120.0 -0.995260  0.517073  ... -0.171833    1.00      0
238614  149743.0 -3.366991 -2.516860  ... -0.358974  359.22      0
253273  156172.0  1.976573  0.176315  ... -0.047503   46.00      0
67663    52646.0  1.220535 -0.541122  ...  0.054277   99.33      0
10270    16114.0  0.589056 -0.951085  ...  0.048831  226.78      0
231294  146692.0  0.086674  0.864377  ...  0.091495    6.99      0
164385  116677.0 -1.808220  0.699659  ...  0.027104   29.95      0
249775  154572.0  2.334682 -0.411422  ... -0.097804   20.00      0
70516    53902.0  1.180631 -1.109888  ...  0.040819   81.35      0
283563  171693.0  1.509038 -0.661789  ... -0.084257   38.96      0

[10 rows x 31 columns]
```

3. Verificați dacă în setul de date există atribute cu valori lipsă.

```
Missing values: False
```

4. Separați din setul de date atributele predictive (`X`) și atributul țintă (`y`).

```
X shape: (284807, 29)
y shape: (284807,)
```

5. Calculați numărul de instanțe pozitive din setul de date.

Comentați rezultatele obținute.

```
Number of positive samples in dataset: 492 (0.17% of total)
```

6. Împărțiți datele în set de antrenare și set de validare.

7. Standardizați valorile atributelor predictive.

8. Creați următorul model de rețea neuronală:

```
model = models.Sequential([
    layers.Dense(128, activation='relu',
                 input_shape=(X_train.shape[-1],)),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(1, activation='sigmoid'),
])

model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=[Precision(name='precision'),
                       Recall(name='recall')])
```

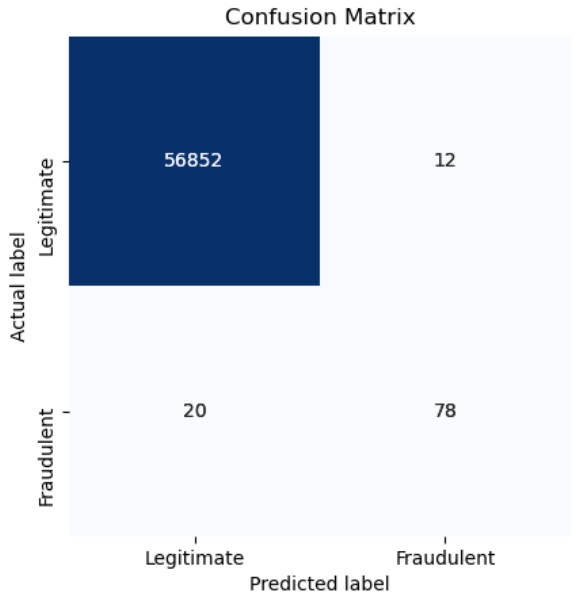
Antrenați modelul timp de 30 de epoci folosind o dimensiune a lotului de 128 de instanțe.

Evaluați modelul pe setul de validare.

Reprezentați grafic matricea de confuzie.

Comentați rezultatele obținute.

```
Val Loss: 0.0155, Precision: 0.87, Recall: 0.80
```



9. Antrenați un nou model folosind ponderi pentru clase, invers proporționale cu numărul de instanțe din fiecare clasă.

Comentați rezultatele obținute.

```
Val Loss: 0.0335, Precision: 0.12, Recall: 0.90
```