

# Raport z projektu: Analiza i przewidywanie niewypłacalności kart kredytowych

Data: 29 maja 2025 r.

**Cel projektu:** Opracowanie modeli uczenia maszynowego do klasyfikacji (przewidywanie niewypłacalności, zmienna `default.payment.next.month`) i regresji (przewidywanie limitu kredytowego, `LIMIT_BAL`) na podstawie danych UCI Credit Card Dataset.

## Przetwarzanie danych:

- Dane załadowano z `UCI_Credit_Card.csv`. Brak brakujących wartości.
- Cechy numeryczne: `SimpleImputer` (średnia), `StandardScaler`.
- Cechy katagoryczne: `OneHotEncoder`.
- Podział na zbiory: treningowy, walidacyjny, testowy.

## Modele klasyfikacji:

1. Logistic Regression (scikit-learn)
2. Decision Tree (scikit-learn)
3. SVM (scikit-learn)
4. Custom Logistic Regression (NumPy, gradient descent)
5. PyTorch Logistic Regression (CPU)

## Wyniki klasyfikacji:

| Model                       | Train Accuracy | Train F1 | Val Accuracy | Val F1 | Test Accuracy | Test F1 |
|-----------------------------|----------------|----------|--------------|--------|---------------|---------|
| Logistic Regression         | 0.822          | 0.477    | 0.815        | 0.431  | 0.826         | 0.479   |
| Decision Tree               | 0.996          | 0.991    | 0.721        | 0.384  | 0.729         | 0.404   |
| SVM                         | 0.823          | 0.456    | 0.816        | 0.406  | 0.824         | 0.449   |
| Custom Logistic Regression  | 0.786          | 0.542    | 0.768        | 0.497  | 0.791         | 0.558   |
| PyTorch Logistic Regression | 0.821          | 0.475    | 0.816        | 0.439  | 0.825         | 0.478   |

## Modele regresji:

1. Custom Linear Regression (NumPy, closed-form)
2. SKLearn Linear Regression

## Wyniki regresji:

| Model                     | MSE Train | MSE Val   | MSE Test  |
|---------------------------|-----------|-----------|-----------|
| Custom Linear Regression  | 9.986e+09 | 1.034e+10 | 1.021e+10 |
| SKLearn Linear Regression | 9.986e+09 | 1.034e+10 | 1.021e+10 |

#### Analiza:

- Klasyfikacja: Logistic Regression i SVM najlepsze (Test Accuracy ~0.824–0.826). Decision Tree przeuczony (Test Accuracy: 0.729). Custom Logistic Regression lepsza w F1 (0.558). PyTorch porównywalny z scikit-learn.
- Regresja: Custom i SKLearn Linear Regression identyczne, ale wysokie MSE sugerują potrzebę bardziej złożonych modeli.

#### Wnioski:

- Wykonano przetwarzanie danych, implementacje modeli w NumPy i PyTorch oraz ocenę (Accuracy, F1, MSE).
- Wyniki zapisano w results/classification\_results.csv i results/regression\_results.csv.
- Zalecenia: rozważyć Random Forest lub sieci neuronowe dla lepszych wyników.