



Uniwersytet
Bielsko-Bialski

Prognozowanie parametrów zdrowotnych pacjentów

Sprawozdanie z Ćwiczeń
Nauka o Danych II

Data wykonania:
28.06.2025

Autor:
Bartosz Bieniek 058085

1. Cel ćwiczenia

12.1 Prognozowanie parametrów zdrowotnych pacjentów

Dane treningowe:

- wiek, BMI, aktywność fizyczna, spożycie kalorii, liczba godzin snu

Zmienne wyjściowe:

- poziom cukru we krwi, ciśnienie skurczowe, ciśnienie rozkurczowe

Zadanie:

Stwórz model przewidujący parametry zdrowotne pacjentów.

```
import numpy as np
```

```
np.random.seed(42)
```

```
X = np.random.rand(1000, 5) * [80, 40, 2, 3000, 10] # wiek, BMI, aktywność, kalorie, sen
```

```
y = np.random.rand(1000, 3) * [200, 180, 120] # cukier, ciśnienie skurczowe, ciśnienie rozkurczowe
```

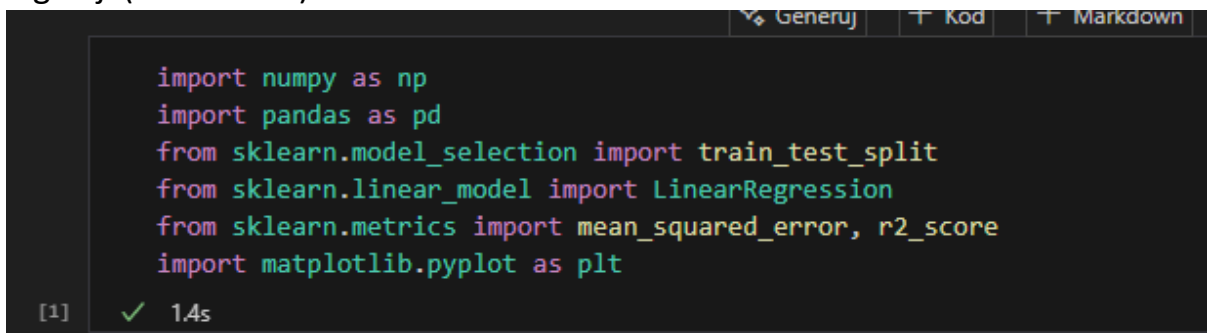
Zadania umieścić na [Github](#).

2. Przebieg ćwiczenia

Sprawozdanie z regresji wielowymiarowej: Prognozowanie parametrów zdrowotnych pacjentów

1. Import bibliotek

Zaimportowano niezbędne biblioteki, które umożliwiły manipulację danymi (NumPy, pandas), wizualizację (matplotlib) oraz budowę i ocenę modelu regresji (scikit-learn).



```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
```

[1] ✓ 1.4s

Rys. 1. Import bibliotek

2. Generacja danych

W celu zapewnienia powtarzalności wyników ustawiono ziarno generatora liczb losowych. Następnie wygenerowano 1000 próbek danych wejściowych o pięciu wymiarach: wiek, wskaźnik BMI, poziom aktywności fizycznej, dzienne spożycie kalorii oraz liczba godzin snu. Równocześnie wygenerowano odpowiadające dane wyjściowe, czyli trzy zmienne zdrowotne: poziom cukru we krwi, ciśnienie skurczowe i ciśnienie rozkurczowe.

```
np.random.seed(42)

# Wiek, BMI, aktywność fizyczna, spożycie kalorii, liczba godzin snu
X = np.random.rand(1000, 5) * [80, 40, 2, 3000, 10]

# Poziom cukru, ciśnienie skurczowe, ciśnienie rozkurczowe
y = np.random.rand(1000, 3) * [200, 180, 120]
```

[2] ✓ 0.0s

Rys. 2. Generacja danych

3. Podział zbioru danych

Zbiór danych podzielono na część treningową (80%) i testową (20%) z wykorzystaniem funkcji `train_test_split`. Użyto tego samego ziarna losowości, co wcześniej, aby zapewnić spójność w replikowaniu wyników.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

[3] ✓ 0.0s

Rys. 3. Podział zbioru danych

4. Trenowanie modelu

Utworzono model regresji liniowej i dokonano jego treningu na danych treningowych. Model został dopasowany do danych wejściowych w celu odwzorowania zależności pomiędzy nimi a trzema zmiennymi wyjściowymi.

```
model = LinearRegression()
model.fit(X_train, y_train)
```

[4] ✓ 0.0s

Rys. 4. Trenowanie modelu

5. Predykcja

Po zakończeniu procesu uczenia, wykorzystano model do przewidzenia wyników (cukru, ciśnienia skurczowego i rozkurczowego) dla danych testowych.

```
[5] y_pred = model.predict(X_test) ✓ 0.0s
```

Rys. 5. Predykcja

6. Obliczenie błędu MSE

W celu oceny dokładności modelu obliczono średni błąd kwadratowy (MSE) dla każdej zmiennej wyjściowej. Otrzymano wartości: 3352,41 dla poziomu cukru, 2482,28 dla ciśnienia skurczowego oraz 1236,10 dla ciśnienia rozkurczowego.

```
[6] mse = mean_squared_error(y_test, y_pred, multioutput='raw_values')
    r2 = r2_score(y_test, y_pred, multioutput='raw_values')

    print("Mean Squared Error (cukier, skurczowe, rozkurczowe):", mse)
    print("R^2 Score (cukier, skurczowe, rozkurczowe):", r2) ✓ 0.0s

... Mean Squared Error (cukier, skurczowe, rozkurczowe): [3352.40938387 2482.28127583 1236.10357925]
    R^2 Score (cukier, skurczowe, rozkurczowe): [-0.00895378 -0.00739375 -0.02209181]
```

Rys. 6. Obliczenie błędu MSE

7. Obliczenie współczynnika R^2

Dodatkowo obliczono współczynniki determinacji R^2 . Otrzymane wartości: -0,009, -0,007 oraz -0,022 wskazują, że model nie wyjaśnia zmienności danych lepiej niż przewidywanie średniej — skuteczność była bardzo niska.

8. Porównanie wyników

Porównano wartości rzeczywiste i przewidywane dla pierwszych dziesięciu obserwacji z danych testowych. Zauważono wyraźne rozbieżności między prognozowanymi a faktycznymi wartościami.

```
print("Rzeczywiste wartości:\n", y_test[:10])
print("Przewidywane wartości:\n", y_pred[:10])
```

[7] ✓ 0.0s Python

... Rzeczywiste wartości:

[115.38722467	27.87145104	42.74327322]
[34.153652	48.09778119	61.09360622]
[189.21512361	44.04880746	15.7878368]
[38.82575409	96.58935167	112.17864485]
[34.89969728	68.166604	101.1773838]
[89.14596418	150.5013508	31.79604739]
[153.75283614	47.8057677	92.0680073]
[121.39462325	33.9299735	96.59602007]
[142.87751241	164.99575848	84.65959823]
[64.19940913	94.69692239	82.53230476]]

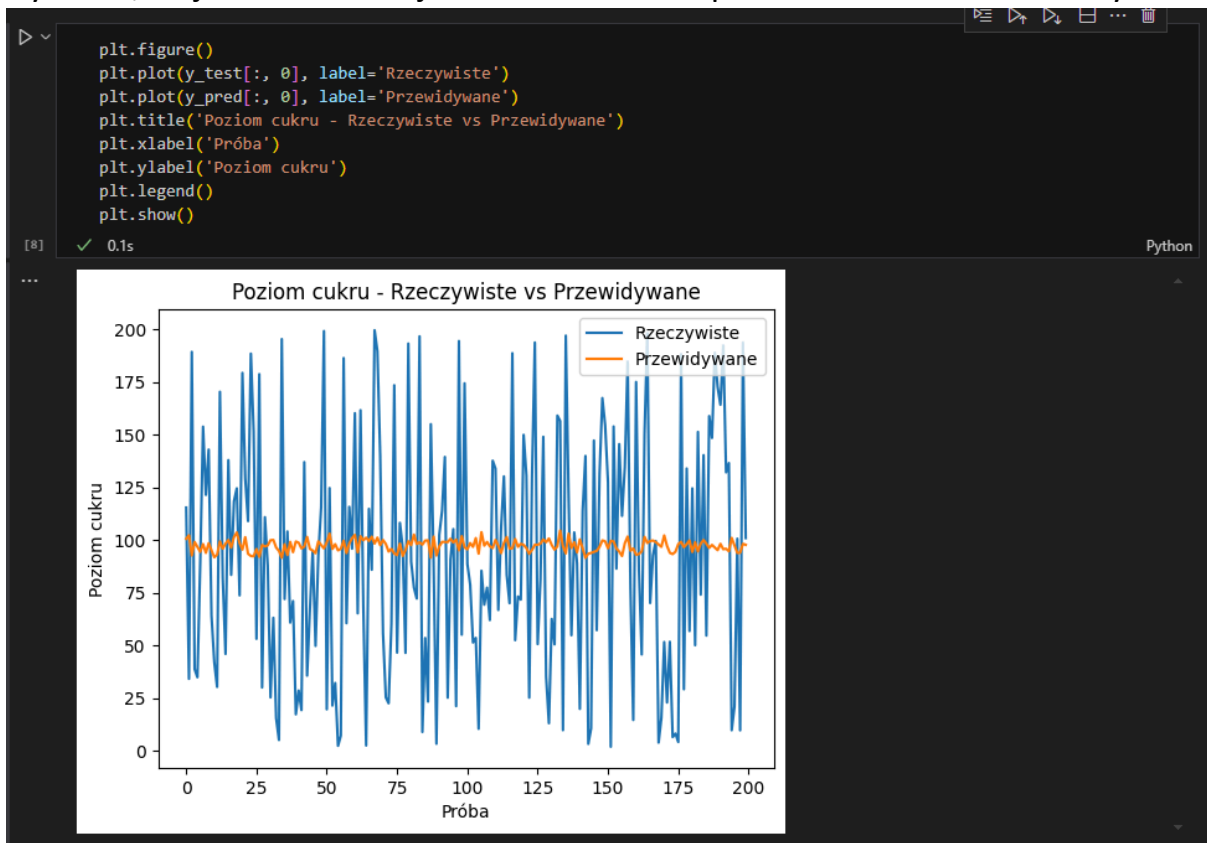
Przewidywane wartości:

[100.57359063	89.22104702	62.65150398]
[102.263723	87.49461266	60.3270513]
[92.6587479	91.75116989	53.22685562]
[99.08376285	86.62942943	60.47844493]
[96.74246001	91.36821944	56.57134357]
[94.40975983	88.65106883	57.24190241]
[98.09898255	89.86724033	58.58506873]
[93.90693479	90.32692639	57.40890739]
[98.49166898	89.65707746	59.7225709]
[95.0701386	94.98254489	51.82526185]]

Rys. 8. Porównanie wyników

9. Wizualizacja

Przedstawiono wykres porównujący rzeczywiste i przewidywane wartości poziomu cukru we krwi. Obserwacje wizualne potwierdziły niską zgodność wyników, co jeszcze bardziej uwidocznili niedopasowanie modelu do danych.



Rys. 9. Wizualizacja

3. Wnioski

Zastosowany model regresji liniowej nie poradził sobie z przewidywaniem parametrów zdrowotnych pacjentów, na co wskazują bardzo niskie, ujemne wartości współczynnika determinacji R^2 oraz wysokie błędy średniokwadratowe dla wszystkich zmiennych wyjściowych. Brak skuteczności modelu wynikał stuprocentowo z faktu, że dane zostały wygenerowane losowo i nie zawierały rzeczywistych zależności między zmiennymi wejściowymi a wyjściowymi.