

**Metody sztucznej inteligencji**

laboratorium

**Opis aplikacji**

Aplikacja Internetowa do klasyfikacji opinii

Adrian Matys

Sara Fijołek

Vladyslav Kutsyn  
Mykyta Mykulskyi

Prowadzący mgr inż. Zdzisław Pawelec

[**Typy danych, zapis i działanie systemu 3**](#_hfwq7dsj7tm7)

[1. Typy danych używane w systemie 3](#_4vro0t5jrlab)

[● Tekst (string): 3](#_620i20sqh1n3)

[● Liczby zmiennoprzecinkowe (float): 3](#_rdvk0rs0rnwd)

[● Lista (list): 3](#_gowdjmhy9sc0)

[● Label (string): 3](#_6hc69ay94lex)

[● JSON / Dict (słownik Python): 3](#_git9pqibwkt8)

[● Pliki modelu (np. .h5, .pt): 3](#_9ixrb2ieygwl)

[2. Zapis danych 4](#_adi0ii5nk5ko)

[● Dane wejściowe: 4](#_vcmp3p2pxz2w)

[● Dane do trenowania: 4](#_dfhy5rj1zywc)

[3. Opis działania systemu 5](#_ap7q8gvdyct8)

[Krok 1: Wprowadzenie danych przez użytkownika 5](#_qrvlpdfeebll)

[Krok 2: Przesłanie danych do backendu 5](#_h9axi5d7gcah)

[Krok 3: Przetwarzanie w backendzie 5](#_fa5v2bpvuzge)

[Krok 4: Klasyfikacja wyniku 5](#_vq8vr6hexdhl)

[Krok 5: Wyświetlenie wyniku 5](#_kbkpmhpdmrid)

[4. Technologie i formaty danych 6](#_f59mu2w88f9)

[● Frontend: 6](#_nfxrki46kqv4)

[● Backend: 6](#_mhuoooddht1p)

[● Biblioteki ML: 6](#_5qyl29hppek)

[● Format modelu: 6](#_3sb93ttpwdr)

[● Format danych treningowych: 6](#_c0ho4yymb8kw)

# Typy danych, zapis i działanie systemu

## 1. Typy danych używane w systemie

W aplikacji będą wykorzystywane następujące typy danych:

### **Tekst (string):**

Główne dane wejściowe pochodzące od użytkownika – opinia wprowadzona w pole tekstowe. Przykład: "Ten produkt jest świetny, polecam każdemu!".

### **Liczby zmiennoprzecinkowe (float):**

Wartości reprezentujące ocenę sentymentalną poszczególnych słów (np. +0.8 dla "świetny", -0.9 dla "okropny").

### **Lista (list):**

Tekst użytkownika zostanie rozbity na listę tokenów (słów lub fraz), które będą analizowane indywidualnie.

### **Label (string):**

Wynik klasyfikacji: "pozytywna", "negatywna" lub "neutralna".

### **JSON / Dict (słownik Python):**

Reprezentacja pośrednich danych w czasie przetwarzania, np. lista słów z przypisanymi ocenami.

## 2. Zapis danych

### **Dane wejściowe:**

Tekst wprowadzony przez użytkownika nie będzie przechowywany na serwerze w celu ochrony prywatności (chyba że użytkownik wyrazi zgodę). Można jednak tymczasowo zapisać dane sesji w pamięci RAM.

### **Dane do trenowania:**

Opinie z etykietami ("pozytywna", "negatywna", "neutralna") zapisane będą w pliku CSV lub bazie danych SQLite/PostgreSQL w formacie:

| **Opinia** | **Klasa** |
| --- | --- |
| Produkt spełnił moje oczekiwania | pozytywna |
| Nie polecam – bardzo słaba jakość | negatywna |
| Wszystko było w porządku, nic specjalnego | neutralna |

### 

## 3. Opis działania systemu

### Krok 1: Wprowadzenie danych przez użytkownika

Użytkownik na stronie wpisuje opinię w pole tekstowe i klika przycisk „Analizuj opinię”.

### Krok 2: Przesłanie danych do backendu

Tekst zostaje przesłany przez frontend (Vue.js) do backendu Flask przy użyciu zapytania HTTP POST.

### Krok 3: Przetwarzanie w backendzie

1. Tekst jest tokenizowany – rozbijany na pojedyncze słowa.
2. Każde słowo trafia do wytrenowanej sieci neuronowej.
3. Sieć przypisuje każdemu słowu wartość sentymentalną (zwykle w zakresie od -1 do 1).
4. Obliczana jest średnia wartość sentymentalna całego tekstu.

### Krok 4: Klasyfikacja wyniku

* Jeżeli średnia wartość > 0.2 → **opinia pozytywna**
* Jeżeli wartość < -0.2 → **opinia negatywna**
* W przeciwnym wypadku → **opinia neutralna**

### Krok 5: Wyświetlenie wyniku

Wynik klasyfikacji jest przesyłany z backendu do frontend i wyświetlany użytkownikowi w ustalonej formie.

### 

## 4. Technologie i formaty danych

### Frontend:

**Vue 3 + Vite (dane w formacie JSON)**

### Backend:

**Flask (Python), komunikacja REST API**

### Biblioteki ML:

**TensorFlow/Keras lub PyTorch**

### Format modelu:

**.h5 (Keras) lub .pt (PyTorch)**

### Format danych treningowych:

**.csv lub baza SQL**