

# Evopy Model Comparison Report

## Raport porównawczy modeli LLM dla Evopy

Data wygenerowania: 2025-05-09 15:07:08

### Podsumowanie wyników

#### Wyniki ogólne

Model	Testy zapytań	Testy poprawności	Testy wydajności	Średni czas (s)	Całkowity wynik
claude	3/3 (100.0%)	0/0 (0.0%)	0/0 (0.0%)	0.00	3/3 (100.0%)
gemini	3/3 (100.0%)	0/0 (0.0%)	0/0 (0.0%)	0.00	3/3 (100.0%)
gpt-4	3/3 (100.0%)	0/0 (0.0%)	0/0 (0.0%)	0.00	3/3 (100.0%)

#### Dokładność konwersji tekst-na-kod

Model	Poprawność kodu	Błędy składniowe	Błędy semantyczne	Zgodność z intencją
claude	60.0%	0.0%	0.0%	54.0%
gemini	60.0%	0.0%	0.0%	54.0%
gpt-4	60.0%	0.0%	0.0%	54.0%

## Wydajność kodu

Model | Złożoność czasowa | Ocena | Złożoność pamięciowa | Ocena | Efektywność rozmiaru | Wykorzystanie zasobów | Ogólna ocena |

	---		-----		----		-----		----		-----		-----			claude		O(n)		0.0%		O(1)		0.0%		0.0%		0.0%		0.0%			gemma
	O(n)		0.0%		O(1)		0.0%		0.0%		0.0%		0.0%			gpt-4		O(n)		0.0%		O(1)		0.0%		0.0%		0.0%		0.0%			llama3

**Wyjaśnienie złożoności obliczeniowej:** - **O(1)**: Złożoność stała - czas wykonania nie zależy od rozmiaru danych wejściowych - **O(log n)**: Złożoność logarytmiczna - czas wykonania rośnie logarytmicznie z rozmiarem danych - **O(n)**: Złożoność liniowa - czas wykonania rośnie liniowo z rozmiarem danych - **O(n log n)**: Złożoność linearytmiczna - typowa dla efektywnych algorytmów sortowania - **O(n²)**: Złożoność kwadratowa - czas wykonania rośnie z kwadratem rozmiaru danych - **O(2^n)**: Złożoność wykładnicza - czas wykonania rośnie wykładniczo z rozmiarem danych

## Jakość wyjaśnień i kodu

Model | Jakość dokumentacji | Klarowność wyjaśnień | Czytelność kodu | Indeks utrzymywalności |

-----	-----	-----	-----	-----	claude   0.0%   0.0%   0.0%   0.0%	gemini   0.0%   0.0%   0.0%   0.0%	gpt-4
0.0%	0.0%	0.0%	0.0%				

## Zgodność z intencjami użytkownika

Model	Spełnienie wymagań	Obsługa przypadków brzegowych	Ocena użytkownika	Ogólna zgodność
claude	100.0%	80.0%	90.0%	92.0%
gemini	100.0%	80.0%	90.0%	92.0%
gpt-4	100.0%	80.0%	90.0%	92.0%

## Wizualizacja wyników

## Wykresy porównawcze

```
<canvas id="radar-chart" class="evopy-chart" data-chart='{
  "type": "radar",
```

```
"data": {
  "labels": [
    "Poprawność kodu",
    "Jakość wyjaśnień",
    "Wydajność kodu",
    "Zgodność z intencjami",
    "Testy podstawowe"
  ],
  "datasets": [
    {
      "label": "claude",
      "data": [
        60.0,
        0,
        0,
        92.0,
        100.0
      ],
      "fill": true,
      "backgroundColor": "rgba(54, 162, 235, 0.2)",
      "borderColor": "rgba(54, 162, 235, 1)",
      "pointBackgroundColor": "rgba(54, 162, 235, 1)",
      "pointBorderColor": "#fff",
      "pointHoverBackgroundColor": "#fff",
      "pointHoverBorderColor": "rgba(54, 162, 235, 1)"
    },
    {
      "label": "gemini",
      "data": [
        60.0,
        0,
        0,
        92.0,
        100.0
      ],
      "fill": true,
      "backgroundColor": "rgba(255, 99, 132, 0.2)",
      "borderColor": "rgba(255, 99, 132, 1)",
      "pointBackgroundColor": "rgba(255, 99, 132, 1)",
      "pointBorderColor": "#fff",
      "pointHoverBackgroundColor": "#fff",
      "pointHoverBorderColor": "rgba(255, 99, 132, 1)"
    }
  ]
}
```



```

        "borderWidth": 1
      }
    ]
  },
  "options": {
    "scales": {
      "y": {
        "beginAtZero": true,
        "max": 100,
        "title": {
          "display": true,
          "text": "Procent sukcesu (%)"
        }
      }
    },
    "plugins": {
      "title": {
        "display": true,
        "text": "Porównanie wyników testów"
      }
    }
  }
}
}'></canvas>

```

```

<canvas id="performance-chart" class="evopy-chart" data-chart='{
  "type": "line",
  "data": {
    "labels": ['claude', 'gemini', 'gpt-4'],
    "datasets": [
      {
        "label": "Średni czas wykonania (s)",
        "data": [0, 0, 0],
        "backgroundColor": "rgba(255, 99, 132, 0.2)",
        "borderColor": "rgba(255, 99, 132, 1)",
        "borderWidth": 2,
        "tension": 0.1
      }
    ]
  },
  "options": {
    "scales": {
      "y": {

```

```
        "beginAtZero": true,  
        "title": {  
            "display": true,  
            "text": "Czas (sekundy)"  
        }  
    },  
    "plugins": {  
        "title": {  
            "display": true,  
            "text": "Porównanie czasu wykonania"  
        }  
    }  
}  
'></canvas>
```

## Analiza trendów

### Postępy w czasie

#### Model: claude

- Brak wystarczających danych historycznych do analizy trendów

#### Model: gemini

- Brak wystarczających danych historycznych do analizy trendów

#### Model: gpt-4

- Brak wystarczających danych historycznych do analizy trendów

## Szczegółowe wyniki testów

### Model: claude

#### Wyniki testów zapytań

- Zaliczone testy: 3/3 (100.0%)
- Ilość wygenerowanego kodu: 0 linii
- Średnia ilość linii na zapytanie: 0.0

#### **Wyniki testów poprawności**

- Zaliczone testy: 0/0 (0.0%)
- Skuteczność kompilacji: 0.0%
- Skuteczność wykonania: 0.0%

#### **Wyniki testów wydajności**

- Brak wyników testów wydajności

### **Model: gemini**

#### **Wyniki testów zapytań**

- Zaliczone testy: 3/3 (100.0%)
- Ilość wygenerowanego kodu: 0 linii
- Średnia ilość linii na zapytanie: 0.0

#### **Wyniki testów poprawności**

- Zaliczone testy: 0/0 (0.0%)
- Skuteczność kompilacji: 0.0%
- Skuteczność wykonania: 0.0%

#### **Wyniki testów wydajności**

- Brak wyników testów wydajności

### **Model: gpt-4**

#### **Wyniki testów zapytań**

- Zaliczone testy: 3/3 (100.0%)

- Ilość wygenerowanego kodu: 0 linii
- Średnia ilość linii na zapytanie: 0.0

#### Wyniki testów poprawności

- Zaliczone testy: 0/0 (0.0%)
- Skuteczność kompilacji: 0.0%
- Skuteczność wykonania: 0.0%

#### Wyniki testów wydajności

- Brak wyników testów wydajności

## Wnioski

Na podstawie przeprowadzonych testów można wyciągnąć następujące wnioski:

1. **Najlepszy model pod względem poprawności:** (0.0%)
2. **Najszybszy model:** (średni czas: infs)
3. **Najlepszy model ogólnie:** claude (ogólny wynik: 100.0%)

## Metodologia testów

Testy zostały przeprowadzone w trzech kategoriach:

1. **Testy zapytań:** Sprawdzają zdolność modelu do generowania poprawnego kodu na podstawie zapytań w języku naturalnym
2. **Testy poprawności:** Weryfikują poprawność wygenerowanego kodu i opisów
3. **Testy wydajności:** Mierzą czas wykonania różnych operacji przez model

## Zalecenia

Na podstawie wyników testów zalecamy:

1. **Do zadań wymagających wysokiej dokładności:**
2. **Do zadań wymagających szybkiego działania:**



### 3. Do ogólnego użytku: claude

---

Wygenerowano przez Evopy Report Generator

© 2025 Evopy