# Analiza możliwości lokalnego uruchamiania modeli językowych wśród graczy Steam na podstawie danych sprzętowych

Autor: [Twoje imię i nazwisko]

Data: [Data złożenia]

Przedmiot / prowadzący: [opcjonalnie]

# Spis treści

1. Wstęp
2. Abstrakt i hipotezy badawcze
3. Metodologia i środowisko analizy
4. Dane i proces ich przygotowania
5. Klasyfikacja sprzętu wg zdolności ML/LLM
6. Analiza statystyczna i wizualna
7. Wnioski i rekomendacje
8. Literatura i przypisy

# 1. Wstęp

W ostatnich latach obserwujemy dynamiczny wzrost popularności dużych modeli językowych (LLM – Large Language Models), wykorzystywanych zarówno w zastosowaniach naukowych, jak i codziennej produktywności. Mimo to, ich użytkowanie wciąż opiera się w dużej mierze na rozwiązaniach chmurowych, co wiąże się z kosztami, ograniczeniami prywatności oraz wymogiem stałego połączenia z Internetem.

Równolegle rośnie dostępność modeli open-source oraz lokalnych środowisk uruchomieniowych takich jak LM Studio, Ollama, czy WebUI, umożliwiających uruchomienie modeli bezpośrednio na komputerze użytkownika. Kluczowym ograniczeniem pozostaje jednak kwestia mocy obliczeniowej – w szczególności posiadanej przez przeciętnego użytkownika domowego.

Platforma Steam, gromadząca miesięcznie ponad 120 milionów aktywnych użytkowników, publikuje regularnie dane o konfiguracjach sprzętowych swoich graczy. Analiza tych danych pozwala nie tylko oszacować obecny stan „komputerów domowych”, ale także odpowiedzieć na pytanie: ilu użytkowników jest już dziś w stanie lokalnie uruchomić model językowy?

W niniejszym opracowaniu przeanalizowano dane sprzętowe graczy Steam z kwietnia 2025 roku pod kątem zgodności ich GPU z wymaganiami typowymi dla uruchamiania LLM, ze szczególnym uwzględnieniem obsługi CUDA, VRAM, typu pamięci i klasy sprzętu. Badanie uzupełniono o dane z TechPowerUp oraz benchmarki inference, w celu przypisania klas zdolności (np. None, ML only, LLM Ready) do najpopularniejszych modeli GPU.

# 2. Abstrakt i hipotezy badawcze

Rozwój dużych modeli językowych (LLM) i ich zastosowań w codziennej pracy sprawia, że coraz więcej użytkowników interesuje się możliwością ich lokalnego uruchamiania. Rozwiązania takie jak LM Studio, Ollama czy WebUI umożliwiają prywatne przetwarzanie danych, eliminując zależność od infrastruktury chmurowej i poprawiając suwerenność cyfrową. Kluczową barierą pozostaje jednak sprzęt — nie każdy komputer osobisty jest zdolny do efektywnego działania z modelem klasy LLM.

W niniejszym badaniu przeanalizowano dane z kwietniowego Steam Hardware Survey 2025, obejmujące konfiguracje sprzętowe milionów graczy. Najpopularniejsze jednostki GPU zestawiono z danymi technicznymi pozyskanymi z bazy TechPowerUp, obejmującymi ilość pamięci VRAM, typ zastosowanej pamięci, wsparcie dla CUDA oraz innych technologii akceleracyjnych.

Na tej podstawie dokonano klasyfikacji kart graficznych względem zdolności do uruchamiania lokalnych modeli — od „braku możliwości” po „pełną gotowość do inference dużych LLM”. Analiza statystyczna udziałów poszczególnych klas wskazuje, że większość graczy dysponuje sprzętem zdolnym do podstawowego uczenia maszynowego, zaś znaczący odsetek użytkowników może już dziś uruchamiać modele językowe lokalnie, pod warunkiem ich optymalizacji.

Wyniki te sugerują, że dalszy rozwój lekkich modeli językowych, zoptymalizowanych pod kątem powszechnie dostępnego sprzętu, może odegrać kluczową rolę w dekomercjalizacji dostępu do AI.

# 3. Metodologia i środowisko analizy

Analiza została przeprowadzona na podstawie danych z kwietniowego Steam Hardware Survey 2025[1](#user-content-fn-1), udostępnianych cyklicznie przez firmę Valve w postaci zestawień udziałów procentowych konfiguracji sprzętowych wśród aktywnych użytkowników platformy Steam. Dane te są **anonimowe**, **agregowane** oraz **niezawierające informacji o liczebności próby**. Nie są też publicznie udostępniane w formacie umożliwiającym bezpośredni import, co wymusiło ręczne przekształcenie (scraping + oczyszczenie) danych do postaci tabelarycznej.

Dla uzupełnienia brakujących informacji o parametrach kart graficznych (VRAM, typ pamięci, rok wydania, wsparcie CUDA/ROCm), wykorzystano dane z bazy TechPowerUp GPU Database pozyskiwane automatycznie za pomocą własnego skryptu Python-Scrapera (BeautifulSoup + pandas). Dane te są nieoficjalne, ale szeroko stosowane w środowisku technicznym i edukacyjnym jako wiarygodne źródło parametrów kart.

Ostateczna klasyfikacja kart na potrzeby tego badania została oparta na następujących kryteriach:

* VRAM: ≥ 12 GB – *LLM Pro Ready*
* VRAM: 8–11 GB – *LLM Basic Ready*
* VRAM: 4–7 GB – *ML only*
* VRAM < 4 GB lub brak wsparcia CUDA – *None*

Modele typu Mistral 7B, LLaMA 2 7B i Phi-2 posłużyły jako reprezentatywne przykłady LLM testowanych lokalnie, zaś informacje o ich wymaganiach pozyskano m.in. z dokumentacji [LM Studio](https://lmstudio.ai), [llama.cpp](https://github.com/ggerganov/llama.cpp) oraz benchmarków użytkowników społeczności HuggingFace[2](#user-content-fn-2).

Należy podkreślić, że:

* Steam nie udostępnia liczbowej wielkości próby (liczby respondentów),
* udział zintegrowanych układów graficznych może być zaniżony przez samoograniczenie próby (gracze z mocniejszym sprzętem częściej uruchamiają gry, co zwiększa szansę ankiety),
* parametry techniczne mogą się różnić w obrębie tego samego modelu (np. VRAM 4/6 GB),
* nie uwzględniono niestandardowych buildów LLM z offloadingiem RAM/CPU.

# 4. Dane i proces ich przygotowania

Dane ze Steam Hardware Survey nie są udostępniane w formie umożliwiającej bezpośrednie pobranie i analizę (np. JSON, CSV, API). W związku z tym dokonano **manualnego scrapowania tabeli z poziomu przeglądarki**, co wymagało:

* zidentyfikowania głównej tabeli zawierającej modele GPU,
* przekształcenia danych do formatu tabelarycznego .xlsx,
* usunięcia pól nieistotnych (np. wzrostów % m/m),
* oznaczenia układów zintegrowanych (iGPU) oraz konsumenckich GPU dyskretnych.

Dla danych technicznych GPU wykorzystano TechPowerUp GPU Database, scrapowaną automatycznie przez skrypt w Pythonie oparty o:

* requests,
* BeautifulSoup,
* pandas.

Kod scrapera dostępny jest w repozytorium (lub załączniku):

<https://github.com/JorgenDeToussaint/LLM-statistic-project>

Każdy model GPU z listy top 50 (według udziału) został dopasowany do strony TechPowerUp, a następnie wyodrębniono kluczowe pola:

* VRAM (w GB),
* typ pamięci (GDDR5/6/6X, HBM, DDR),
* wsparcie CUDA,
* rok premiery,
* TDP (opcjonalnie dla uzupełnienia danych energetycznych).

# 5. Klasyfikacja sprzętu wg zdolności ML/LLM

W tej sekcji przedstawiono sposób podziału najpopularniejszych kart graficznych na klasy odpowiadające ich zdolnościom do uruchamiania lokalnych zastosowań uczenia maszynowego (ML) oraz dużych modeli językowych (LLM). Celem klasyfikacji jest wyodrębnienie czterech grup sprzętu, które w prosty sposób przekładają się na możliwości użytkownika:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Klasa** | **Kryteria minimalne** | **Przykładowe GPU** | | **None** | VRAM < 4 GB **lub** brak wsparcia CUDA/ROCm/oneAPI | Intel Iris Xe, zintegrowane Vega, UHD Graphics | | **ML only** | VRAM ≥ 4 GB, brak Tensor Cores, wsparcie CUDA/ROCm | GTX 1050 Ti, GTX 1650, AMD RX 6600 | | **LLM Basic** | VRAM 6–11 GB, wsparcie CUDA/ROCm, bez (lub słabych) Tensor Cores | RTX 2060, RTX 3050, Intel Arc A770 | | **LLM Pro** | VRAM ≥ 12 GB, wsparcie CUDA/ROCm + Tensor Cores | RTX 3060 (12 GB), RTX 3070, RTX 3090, RX 7900 XT |   **5.1. Kryteria podziału**   1. **Pojemność VRAM**    * < 4 GB: zbyt mało pamięci do wygodnej pracy z większymi modelami ML i LLM → klasyfikacja None.    * 4–5 GB: umożliwia proste modele ML (np. drzewa decyzyjne, regresję, lekkie sieci) → ML only.    * 6–11 GB: wystarczająco do inference „małych” LLM (~7 mld parametrów, quantized) → LLM Basic.    * ≥ 12 GB: pozwala na inference i częściowe fine-tuning „średnich”/„dużych” modeli (~13–30 mld) → LLM Pro. 2. **Wsparcie akceleracji GPU**    * CUDA (NVIDIA) lub ROCm/oneAPI (AMD/Intel) – niezbędne do przyspieszenia operacji tensorowych.    * Obecność Tensor Cores (w architekturze Ampere i nowszych) przyspiesza inferencję LLM, ale nie jest obowiązkowym warunkiem klasy „Basic”. 3. **Architektura GPU**    * Modele nowszych generacji (Ampere, RDNA 2/3, Ada Lovelace) oferują lepsze wsparcie FP16/INT8.    * Starsze architektury (Pascal, Turing) mogą mieć ograniczenia wydajności, ale przy odpowiedniej optymalizacji nadal trafiają do klas „ML only” lub „LLM Basic”. |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **GPU Model** | **VRAM (GB)** | **CUDA/ROCm** | **Architektura** | **Klasa** |
| Intel Iris Xe Graphics | 0 (shared) | – | Xe-LP | None |
| NVIDIA GeForce GTX 1650 | 4 | CUDA | Turing | ML only |
| NVIDIA GeForce RTX 2060 | 6 | CUDA | Turing | LLM Basic |
| Intel Arc A770 | 16 | oneAPI | Xe-HPG | LLM Basic |
| NVIDIA GeForce RTX 3060 (laptop) | 12 | CUDA | Ampere | LLM Pro |
| AMD Radeon RX 6700 XT | 12 | ROCm | RDNA 2 | LLM Pro |

# 6. Analiza statystyczna i wizualna

[Tutaj wpisz treść sekcji]

# 7. Wnioski i rekomendacje

[Tutaj wpisz treść sekcji]

# 8. Literatura i przypisy

Valve Corporation. *Steam Hardware & Software Survey: April 2025*. Dostępne online: <https://store.steampowered.com/hwsurvey/>

TechPowerUp. *GPU Database*. Dostępne online: <https://www.techpowerup.com/gpu-specs/>

LM Studio. *Documentation*. <https://lmstudio.ai/docs>

G. Gerganov. *llama.cpp* (GitHub). <https://github.com/ggerganov/llama.cpp>

HuggingFace. *LLM Inference Benchmarks*. <https://huggingface.co/blog/llm-benchmarks>

Pew Research Center. *The Challenges of Polling When Fewer People Are Available to Be Polled* (2017). <https://www.pewresearch.org>