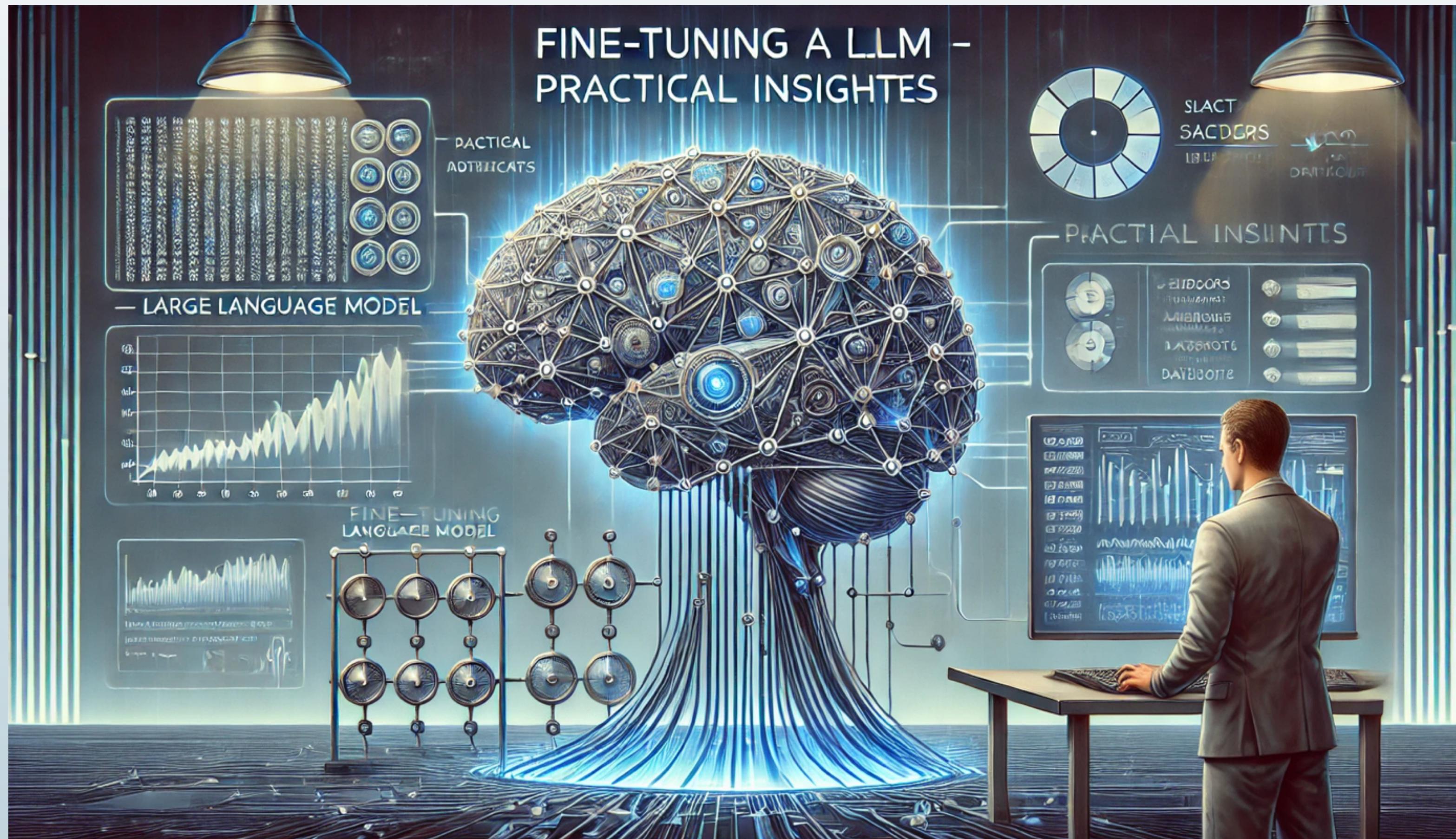


Fintuning a LLM - Practical Insights



Von Marcel Franz und Linnea Feddeck
Metric Space AI



Worum gehts heute?

- Training eines LLMs
- Erklärung an unserem Modell - GDPR
- Complexity, Sensitivity und Entity detection
- Herausforderungen



Trainingsmethode für LLM

Datenaufbereitung & -beschaffung

- Schritt 1: Trainingsdaten erstellen → Ohne Trainingsdaten kann das Modell nichts lernen.

Modelltraining & Feintuning

- Schritt 2: Vortrainiertes Modell als Basis nutzen (z. B. Qwen 2.5 oder ähnliche Modelle von Huggingface)
- Schritt 3: Supervised Fine-Tuning (SFT) → Modell mit gelabelten Daten trainieren.



GDPR Input Detection & Anonymization Model

Ohne:

- Keine automatische Erkennung sensibler Informationen
- Direkte Verarbeitung ohne Schutzmaßnahmen
- Datenschutzrisiken & Compliance-Probleme

The screenshot shows a user interface for a GDPR input detection model. At the top, there is a message from a user asking about composing an email with sensitive information. Below the message, a red shield icon indicates that sensitive content has been detected. A red warning message states: "Sensitive content is detected. Chat temporarily blocked for safety." Below this, there are four buttons: "No Sensitive Data" (with a circular icon), "Remove Data" (with a trash bin icon), "Replace with anonymized data" (with a star icon), and "Stop inspection (30 min)" (with a triangle icon). In the top right corner, there is a "Regenerate" button.

Kannst du mit den Daten von meinem Arbeitskollegen eine Email verfassen?

Mustafa Berlino, mustafa@gmail.com, Tel: 0156 7853452, Bank: DE76 678896876. Schreibe ihm das ich ihn warnen muss weil er zu oft zu spät gekommen ist. Schreibe sie professionell.

Sensitive content is detected. Chat temporarily blocked for safety.

No Sensitive Data Remove Data Replace with anonymized data

Stop inspection (30 min)

Regenerate

Funktion mit:

- 1) **Komplexitäterkennung** für Model Routing
- 2) **Sensitivitäterkennung** als Upload-Filter
- 3) **Anonymisierung**

—> Schutz sensibler Daten vor Cloud-Verarbeitung

Complexity Score

- Einfache Aufgaben → kleinere, schnellere Modelle
- Komplexe Aufgaben → mehr Rechenleistung nötig → leistungsstärkere Modelle
- Effizientere Nutzung von Ressourcen → Reduziert Kosten & Wartezeit

1-3 – Einfache Aufgaben: Faktenabfragen, Paraphrasierung

Beispiel: „Was ist die Hauptstadt von Frankreich?“

4-6 – Mittlere Komplexität: Zusammenfassungen, einfache Analysen

Beispiel: „Extrahiere Herausforderungen und Lösungen als Bullet Points.“

7-10 – Hohe Komplexität: Mehrstufige Analysen, Expertenwissen

Beispiel: „Analysiere den Einfluss von KI auf Immobilieninvestitionen bis 2030 basierend auf zehn Berichten.“



Datenerstellung um das Modell zu trainieren

- Auswahl eines Open-Source-Instruction-Datensatzes
- Automatisches Labeling mit GPT-4o zur Bewertung der Textkomplexität
- Ziel: Trainingsdaten für die Modellbewertung generieren



Beispiel für Komplexitätsbewertung

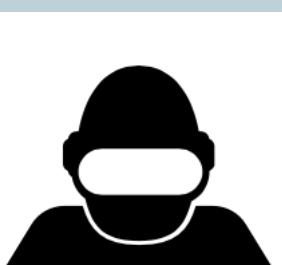
```
▶ ▾ text = "We have a community picnic at Greenfield Park, it is on thursday at 11 AM. Write me an e-mail annoucment!"  
      # Generate complexity score  
      complexity_score = model_inference(text, mode="complexity")  
      print(f"Complexity: {complexity_score}")  
  
[2] ✓ 0.7s  
... Complexity: 3
```



Datenstruktur RAW ohne Parsing

```
<|im_start|>system  
Complexity<|im_end|>  
<|im_start|>user  
{Beispieltext}<|im_end|>  
<|im_start|>assistant  
{Komplexitäts-Score}<|im_end|>
```

- `<|im_start|>system` definiert eine Systemnachricht (Aufgabe) zur Steuerung des Modells.
- `<|im_start|>user` enthält den zu bewertenden Text.
- `<|im_end|>` markiert das Ende jedes Abschnitts



Sensitivity Scoring

0 – Öffentlich

- Keine vertraulichen Inhalte, frei teilbar
- Beispiel: „Community Picnic am Samstag um 11 Uhr im Greenfield Park.“

1 – Intern

- Leicht sensible Daten, für internen Gebrauch
- Beispiel: „Neue Bürozeiten ab nächstem Monat: Mo-Do 9-18 Uhr, Fr bis 13 Uhr.“

2 – Vertraulich

- Informationen, die geschäftliche oder finanzielle Auswirkungen haben können
- Beispiel: „Q3-Budget: Marketing 150.000 €, R&D 200.000 €.“

3 – Hochsensibel

- Kritische oder persönliche Daten, streng geschützt
- Beispiel: „Fusion: Acme übernimmt 75 % von BetaTech für 2,5 Milliarden €.“

Synthetische Datenerstellung um das Modell zu trainieren

-Sensitivity-

- Texte aus dem **C4-Datensatz** als Basis
- Umformulierung mit GPT4omini nach Sensitivitätsregeln 0-3
- **Thema aus vordefinierter Liste** passend zum Level
- Struktur und Kontext von C4 beibehalten
- Ziel: Realistische Datensätze für die Modellbewertung erstellen.

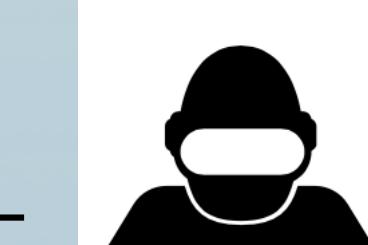


Beispiel für die Sensitivitätsbewertung

```
▶ ▾
text = "We have a community picnic at Greenfield Park, it is on thursday at 11 AM. Write me an e-mail annoucment!"

# Generate sensitivity score
sensitivity_score = model_inference(text, mode="sensitivity")
print(f"Sensitivity Score: {sensitivity_score}")

[3] ✓ 0.1s
...
Sensitivity Score: 0
```



Datenstruktur RAW ohne Parsing

- <|im_start|>system
Sensitivity<|im_end|>
- <|im_start|>user
{Beispieltext}<|im_end|>
- <|im_start|>assistant
{Sensitivitäts-Score}<|im_end|>
- `<|im_start|>system` definiert eine Systemnachricht (Aufgabe) zur Steuerung des Modells.
 - `<|im_start|>user` enthält den zu bewertenden Text.
 - `<|im_end|>` markiert das Ende jedes Abschnitts

Anonymisierung von Texten

Warum ist Anonymisierung wichtig?

- Schutz personenbezogener Daten, ohne die Nützlichkeit zu verlieren
- Platzhalter oder Schwärzen verzerrt oft den Kontext
- Beispiel: Ein Chatbot mit Platzhaltern erkennt keine Namen und kann Nutzer nicht ansprechen

• Kategorien sensibler Daten:

- PERSON, DEM, CODE, LOC, ORG, DATETIME, QUANTITY, MISC

• Methoden der Anonymisierung:

- Randomisierung: Ersetzen durch zufällige Werte derselben Klasse (z. B. „John Doe“ → „Max Mustermann“)
- Generalisierung: Verallgemeinerung sensibler Daten (z. B. „Berlin“ → „Stadt in Deutschland“)

• Austauschen sensibler Entities:

- Entscheidung pro Klasse: Jede erkannte Entität wird entweder randomisiert oder generalisiert.
- Entity Mapping: Ein vorher definierter Mapping-Mechanismus legt fest, wie eine Entität ersetzt wird.
- Automatischer Austausch: Das Modell ersetzt die erkannten Entitäten entsprechend der gewählten Methode.
- Ausgabe: Das Netzwerk generiert eine anonymisierte Version des ursprünglichen Textes.



Anonymisierung von Texten - Entity Detection Beispiel

```
<|im_start|>system  
Entity Detection<|im_end|>  
<|im_start|>user  
We have a community picnic at Greenfield Park, it is on thursday at 11 AM.  
Write me an e-mail annoucment!<|im_end|>  
<|im_start|>assistant  
Greenfield Park--LOC--Sunnyvale Park--Local Park::3--Public Space::6--A Park::9  
thursday--DATETIME--Friday--A Day of the Week::3--Midweek::5--A Day::8  
11 AM--DATETIME--10 AM--Morning::2--A.M hours::4--Daytime::6<|im_end|>
```



Anonymisierung von Texten – Entity Swapping Beispiel

<|im_start|>system

Entity Swapping<|im_end|>

<|im_start|>user

Entities:

Greenfield Park : Sunnyvale Park

thursday : A Day of the Week

11 AM : Morning

Text:

We have a community picnic at Greenfield Park, it is on thursday at 11 AM.

Write me an e-mail annoucment!<|im_end|>

<|im_start|>assistant

We have a community picnic at Sunnyvale Park, it is on A Day of the Week at Morning. Write me an e-mail announcement <|im_end|>



Herausforderungen beim Training von KI-Modellen

1. Verwechslung von Sensitivity & Complexity Scores

Problem: Modell gibt manchmal falsche Werte aus (z. B. Sensitivity 4 & Complexity 0).

Lösung:

Mischung der Trainingsdaten, damit das Modell beide Werte besser differenzieren kann.

2. Ungleichmäßige Datenverteilung

Problem: bestimmte Kategorien (z. B. Score 4) kaum vorkommen

Lösung:

- Synthetische Datengenerierung, um fehlende Klassen aufzufüllen.
- Balancierung der Daten, sodass jede Klasse gleichmäßig vertreten ist.

3. Richtige Hyperparameter für das Training finden

Problem: Trainingsparameter wie Lernrate oder Batchgröße beeinflussen die Modellqualität.

Lösung:

- Orientierung an Hyperparametern von Open-Source-Modellen als Startpunkt.
- Anpassung der Parameter durch heuristische Methoden basierend auf vorhandenen Ressourcen.



Wo zu Finden

Hugging Face Search models, datasets, users...

Models Datasets Spaces Posts Docs Enterprise Pricing

Hugging Face is way more fun with friends and colleagues! 😊 Join an organization Dismiss this message

▲ metricspace/GDPR_Input_Detection_and_Anonymization_0.5B like 3 Following ▲ Metric Space 8

Safetensors qwen2 License: apache-2.0

Model card Files and versions Community

The GDPR Input Detection and Anonymization model

The **The GDPR Input Detection and Anonymization model** is designed to protect sensitive information locally before it is processed by larger AI models in external clouds.

Intended Use

The model is made to bridge the user inputs to external LLM input like a firewall or proxy.

The model analysis the user prompts and computes two scores.

The first score helps to identify if it needs a small or more cabable model to process the user input.

The second score rates the sensitivity of the prompt. When it detects sensitive information, the further cloud processing of the prompt can be blocked or at least be replaced by an anonymized version.

Complexity Scoring

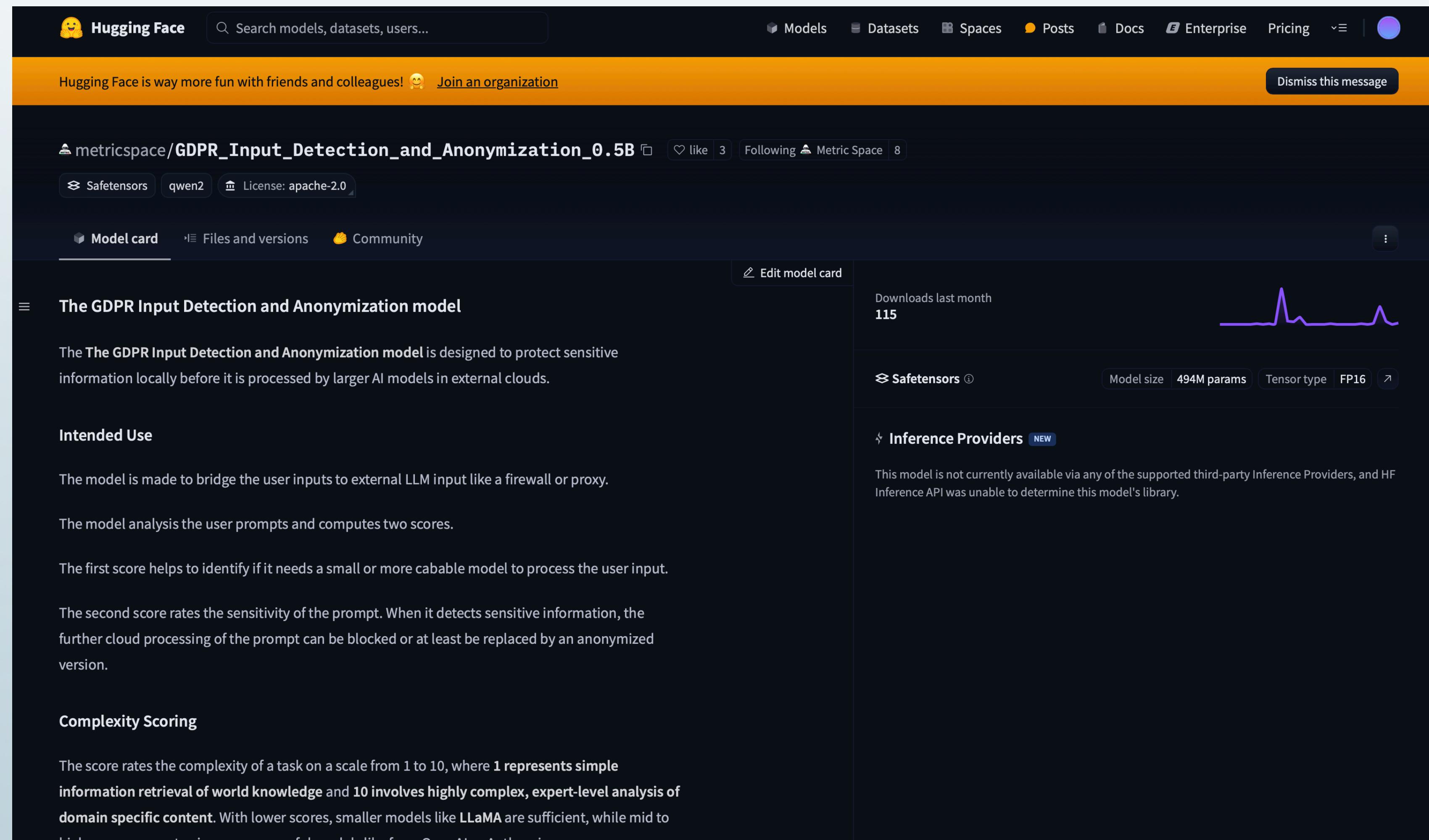
The score rates the complexity of a task on a scale from 1 to 10, where **1 represents simple information retrieval of world knowledge** and **10 involves highly complex, expert-level analysis of domain specific content**. With lower scores, smaller models like LLaMA are sufficient, while mid to high scores suggest using more powerful models like from OpenAI or Anthropic.

Downloads last month 115

Safetensors Model size 494M params Tensor type FP16

Inference Providers NEW

This model is not currently available via any of the supported third-party Inference Providers, and HF Inference API was unable to determine this model's library.



https://huggingface.co/metricspace/GDPR_Input_Detection_and_Anonymization_0.5B





Von Marcel Franz und Linnea Feddeck
Metric Space AI

