1. Finetuning & Execution of the LLM

One of the main components of the Digital Stroke Doctor application is the LLM that determines the probability of a stroke given the input of the user. This section is about researching the most popular and top-of-the-notch technologies, evaluating their pros and cons and deciding which of these technologies fits best for this individual purpose.

1.1. Hauptteil

1.1.1. LLMs

Locally referes to running on a Nvidia GForce 3050 Mobile and Intel Core i5-11400H.

Name	Size	Response time	Coherency
Reflection-Llama-3.10-70B	~280GB	~260s	TODO: Response bekomen
gpt2-open-instruct-v1	~0.5GB	~10s	Random words as response
Llama-3.1-Storm-8B	~16GB	~60s	Clear response

1.1.1.1. Reflection-Llama-3.1-70B

Reflection-Llama-3.1-70B is a model with 70 Billion parameters created for complex natural language processing. It is used for tasks such as summarization, text generation, question answering and tasks that require reasoning.

1.1.1.2. gpt2-open-instruct-v1

1.1.1.3. Llama-3.1-Storm-8B

1.1.2. Languages

Name	Experience	Performance	Support for LLMs
Python			Most modules for LLMs
Julia			Some Modules for LLMs
R			A few Libraries for LLMs

1.1.2.1. Python

Hugging Face Transformers PyTorch TensorFlow

- Experience with the language and modules
- Most modules for LLMs

1.1.2.2. Julia

MLJ.jl

- No experience with language
- High performance

1.1.2.3. R

tensorflow for R keras for R

- Some experience with the language but no experience with the libraries
- Less mature than Python 1.1.2.1

1.1.3. Hosting for Finetuning

Most LLMs are too large to train locally which is why a server with enough VRAM is needed. Possible options are:

1.1.3.1. Google Colab

- Very fast GPUs including NVIDIA T4
- 10 Euros/Month

1.1.3.2. TGM

-

1.1.3.3. Huggingface

- Wide variety of GPUs
- Different prices also including Nvidia T4 - Medium for $0.60/hour