

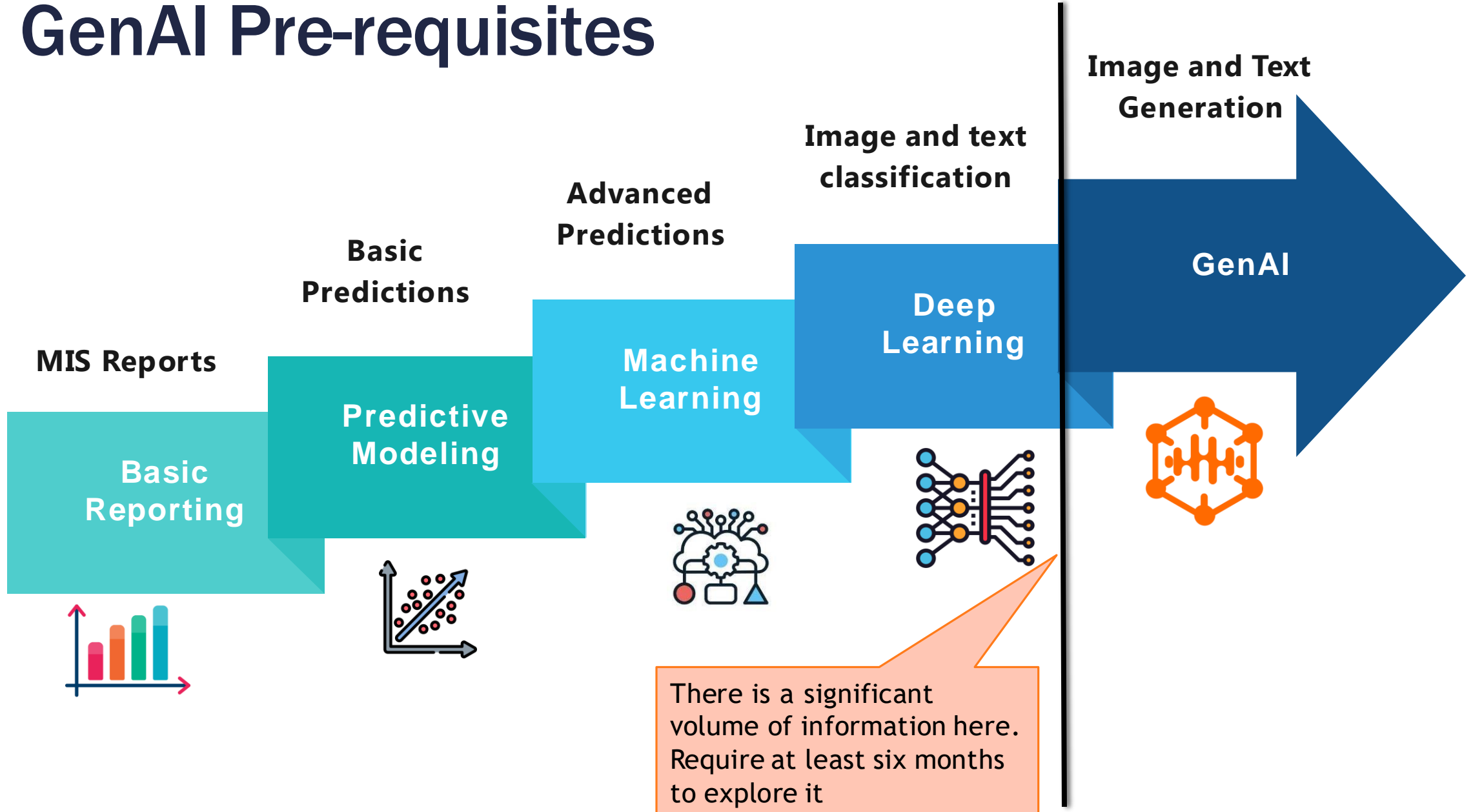
# Hugging Face Models

---

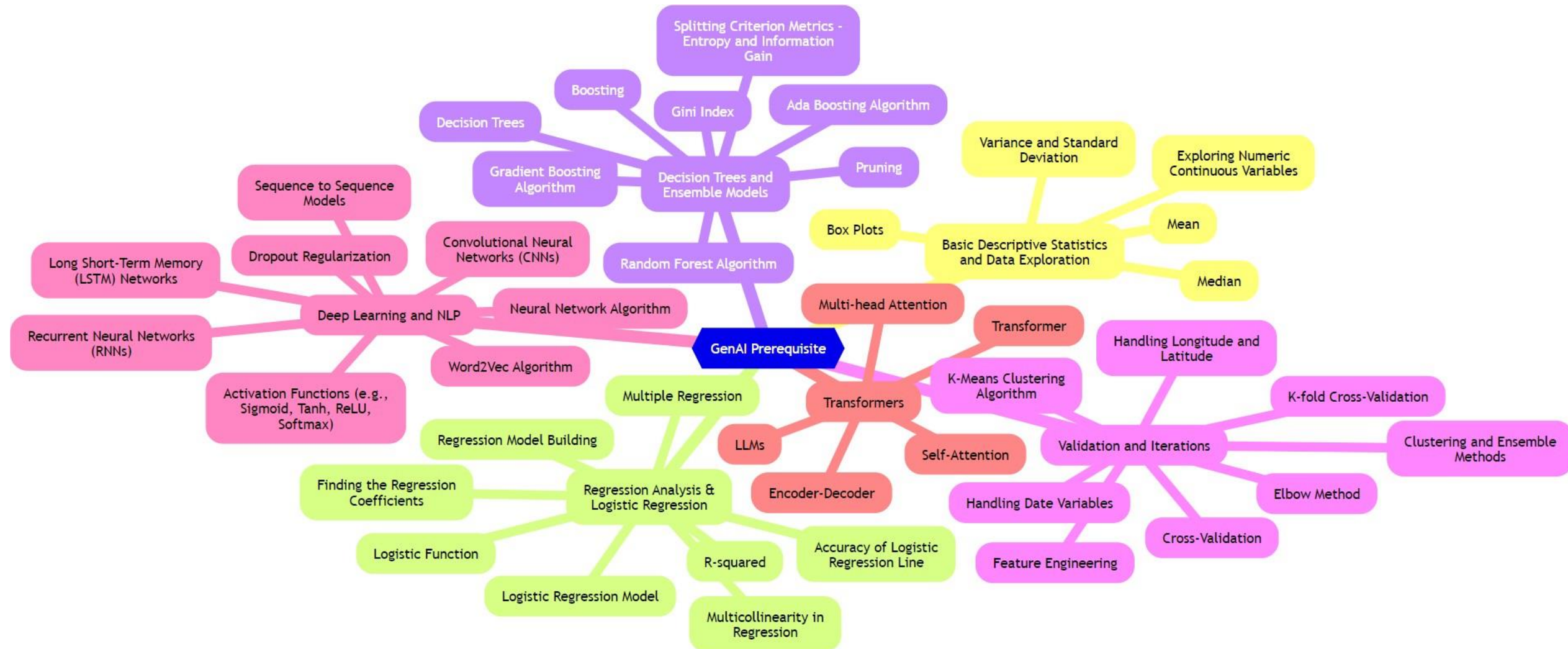
# Contents

- Open source LLMs
- Model Hub on Hugging face
- Model from Hugging face
- pipeline() function
- Document Classification model
- Q&A Model
- NER Model
- Saving the Model on HuggingFace hub
- Final app - Bank Customers Complaints categorization App

# GenAI Pre-requisites

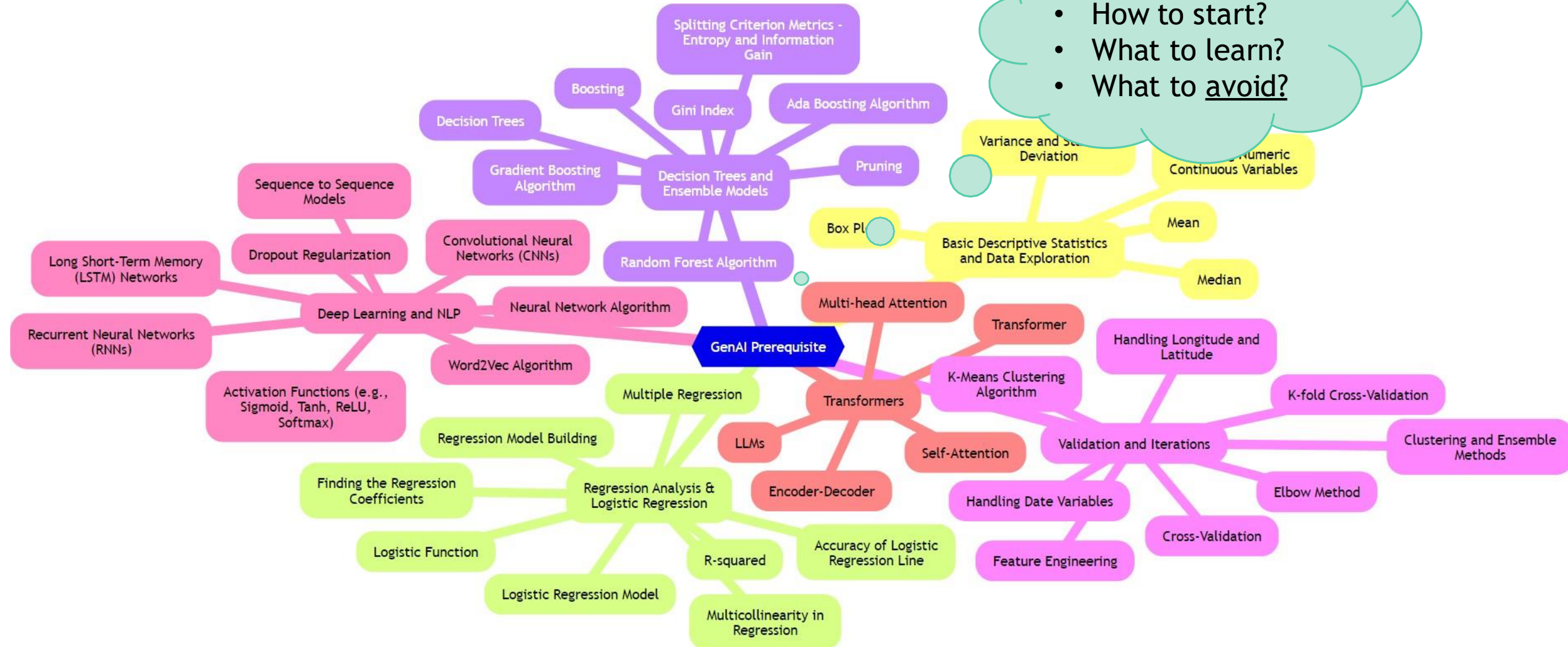


# GenAI Pre-requisites



# GenAI Pre-requisites

- Where to start?
- How to start?
- What to learn?
- What to avoid?



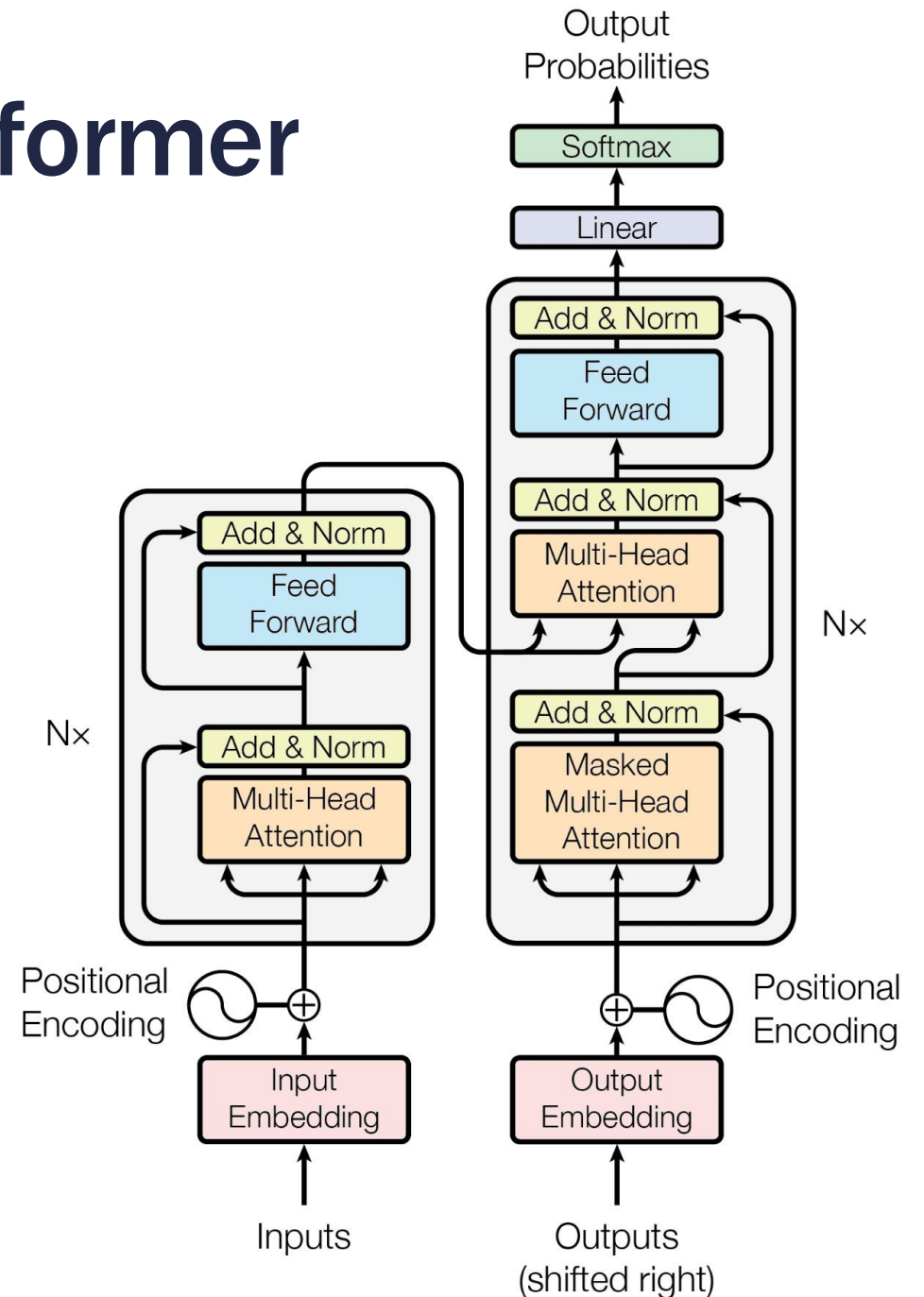


# GenAI Pre-requisites



# Almost every LLM is a Transformer

- Transformer is an advanced Deep Neural Network model
- Transformers revolutionize NLP, forming the basis for modern LLMs.
- Designed to process and understand sequential data like text.
- Capture relationships between words regardless of their position.
- Attention mechanism allows models to focus on all input parts simultaneously.



# What are the opensource LLMs?

- What are the opensource LLMs?
- How to access them using APIs?
- Where to find them?
- Where can we find the documentation ?
- Is there any repository where I can find all the LLMs ? - Yes
- **Hugging Face**



# Hugging Face

- If you are working with language models, you have to be comfortable with Hugging Face hub
- Hugging face is the repository for several pre-trained ML, NLP, Vision based models and datasets.
- You can access various models and datasets.
- You can also upload your own model into hugging face platform

# Repository of all ML and AI models

- Designed for sharing, collaborating, and managing machine learning models, datasets, etc.
- Trying to democratize machine learning, especially in NLP and computer vision. Make the models available for everyone
- Contains good documentation and example codes
- **You must learn** how to use Hugging Face models before jumping on to LLMs



Hugging Face

Search models, datasets, users...

Models

Datasets

Spaces

Posts

Docs

Pricing

**NEW** Create Assistants in HuggingChat



# The AI community building the future.

The platform where the machine learning community collaborates on models, datasets, and applications.

Tasks Libraries Datasets Languages Licenses Other

Filter Tasks by name

Multimodal

- Text-to-Image Image-to-Text
- Text-to-Video Visual Question Answering
- Document Question Answering Graph Machine Learning

Computer Vision

- Depth Estimation Image Classification
- Object Detection Image Segmentation
- Image-to-Image Unconditional Image Generation
- Video Classification Zero-Shot Image Classification

Natural Language Processing

- Text Classification Token Classification
- Table Question Answering Question Answering
- Zero-Shot Classification Translation
- Summarization Conversational
- Text Generation Text2Text Generation
- Sentence Similarity

Audio

- Text-to-Speech Automatic Speech Recognition
- Audio-to-Audio Audio Classification

Models 469,541

meta-llama/Llama-2-70b-chat-hf  
Text Generation • Updated 6 days ago

stabilityai/stable-diffusion-2-1  
Updated 6 days ago • 2.01k

openchat/openchat  
Text Generation • Updated 2 days ago

lillyasviel/ControlNet-v1.1  
Updated Apr 26 • 1.87k

cerspense/zeroscope\_v2\_560m  
Updated 3 days ago • 2.66k • 33

meta-llama/Llama-2-13b-chat-hf  
Text Generation • Updated 4 days ago

tiiuae/falcon-40b-instruct  
Text Generation • Updated 27 days ago

WizardLM/WizardCoder-15B-V1  
Text Generation • Updated 3 days ago

CompVis/stable-diffusion-v1-4  
Text-to-Image • Updated about 17 hours

# Sign-up


Signup and create a login-id

The image is a composite showing the Hugging Face website. At the top, a navigation bar includes the Hugging Face logo, a search bar, and links for Models, Datasets, Spaces, Posts, Docs, Solutions, Pricing, Log In, and a prominent Sign Up button. A speech bubble points to the Sign Up button with the text 'Signup and create a login-id'. Below the navigation bar, the main content area is split. On the left, a dark banner features the Hugging Face logo and the text 'The AI community building the future.' with a 'NEW Try Cohere Command R+ on HuggingChat' badge. On the right, a sidebar displays a list of models under the heading 'Models 469,541'. The list includes 'meta-llama/Llama-2-70b', 'stabilityai/stable-diffusion-xl-base-0.9', 'openchat/openchat', 'llyasviel/ControlNet-v1-1', 'cerspense/zeroscope\_v2\_XL', 'meta-llama/Llama-2-13b', and 'tiiuae/falcon-40b-instruct', each with details on updates, downloads, and likes. The main content area also shows a 'Tasks' section with filters for Multimodal, Computer Vision, and Natural Language Processing.

**Hugging Face** Search models, datasets, users...

Models Datasets Spaces Posts Docs Solutions Pricing Log In **Sign Up**

**NEW** Try Cohere Command R+ on HuggingChat

 **The AI community building the future.**

**Models** 469,541 Filter by name

- meta-llama/Llama-2-70b  
Text Generation • Updated 4 days ago • 25.2k • 64
- stabilityai/stable-diffusion-xl-base-0.9  
Updated 6 days ago • 2.01k • 393
- openchat/openchat  
Text Generation • Updated 2 days ago • 1.3k • 136
- llyasviel/ControlNet-v1-1  
Updated Apr 26 • 1.87k
- cerspense/zeroscope\_v2\_XL  
Updated 3 days ago • 2.66k • 334
- meta-llama/Llama-2-13b  
Text Generation • Updated 4 days ago • 328 • 64
- tiiuae/falcon-40b-instruct  
Text Generation • Updated 27 days ago • 288k • 899

**Tasks** Libraries Datasets Languages Licenses Other

Filter Tasks by name

**Multimodal**

- Text-to-Image Image-to-Text
- Text-to-Video Visual Question Answering
- Document Question Answering Graph Machine Learning

**Computer Vision**

- Depth Estimation Image Classification
- Object Detection Image Segmentation
- Image-to-Image Unconditional Image Generation
- Video Classification Zero-Shot Image Classification

**Natural Language Processing**

- Text Classification Token Classification
- Table Question Answering Question Answering
- Zero-Shot Classification Translation
- Summarization Conversational

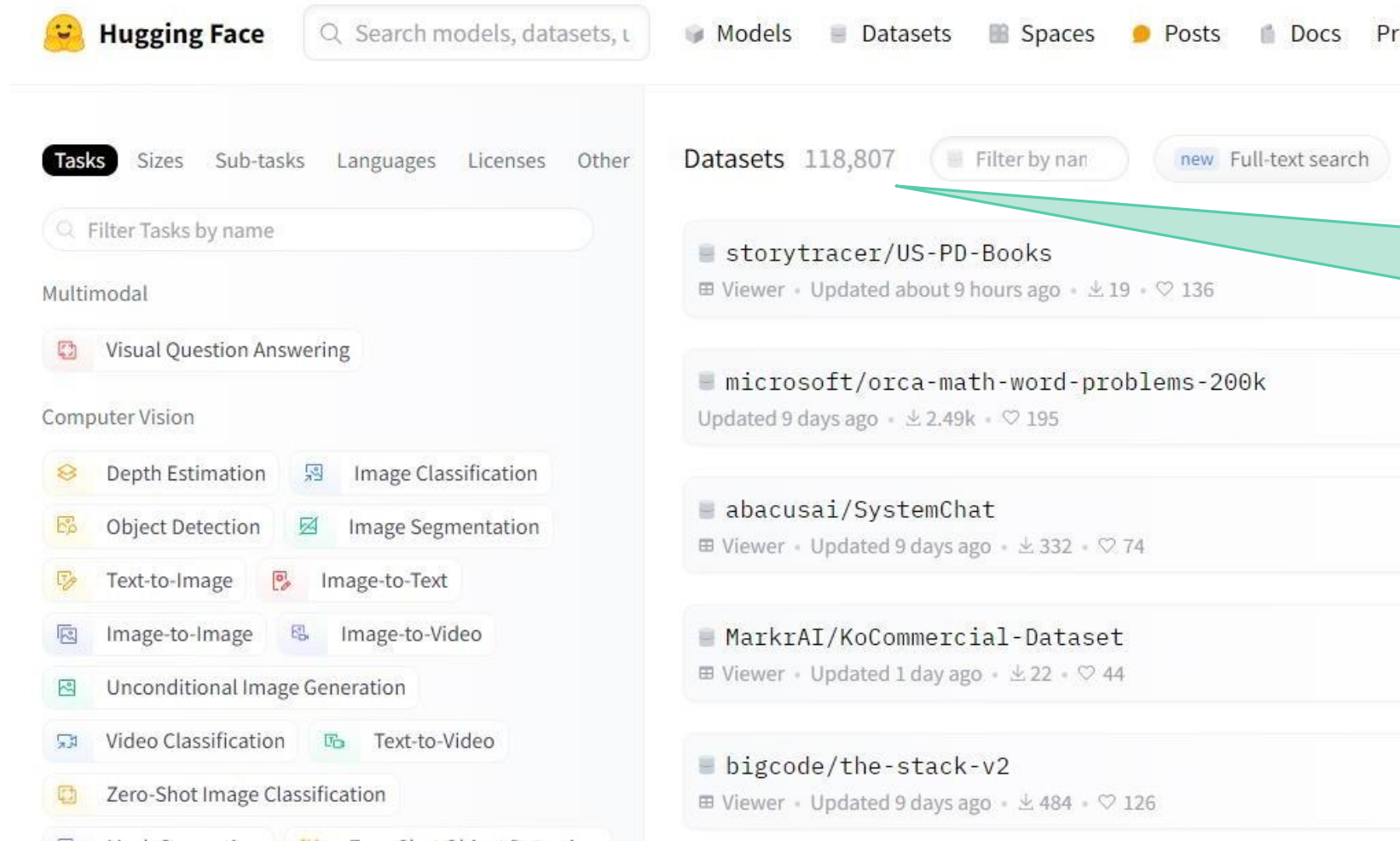
# Model Hub contains a huge collection of pre-trained models

The screenshot displays the Hugging Face Model Hub interface. At the top, the Hugging Face logo is on the left, followed by a search bar labeled "Search models, datasets, users...". To the right of the search bar are navigation links for "Models", "Datasets", "Spaces", "Posts", and "Docs". Below the navigation bar, the left sidebar contains a "Tasks" tab and a list of categories: "Libraries", "Datasets", "Languages", "Licenses", and "Other". A "Filter Tasks by name" search bar is also present. The main content area shows a list of models under the "Models" tab, with a count of "547,770" and a "Filter by name" search bar. The list includes the following models:

- CohereForAI/c4ai-command-r-v01**: Text Generation • Updated about 9 hours ago • 1.14k • 421
- google/gemma-7b**: Text Generation • Updated 14 days ago • 243k • 2.23k
- NousResearch/Genstruct-7B**: Text Generation • Updated 6 days ago • 1.15k • 236
- ByteDance/SDXL-Lightning**: Text-to-Image • Updated 10 days ago • 454k • 1.23k
- 01-ai/Yi-9B**: Text Generation • Updated 5 days ago • 2.62k • 147
- stabilityai/TripoSR**: Image-to-3D • Updated 8 days ago • 20.1k • 216

More than 500,000  
pretrained models

# A huge collection of datasets also



The screenshot shows the Hugging Face website interface. At the top, the Hugging Face logo is on the left, followed by a search bar labeled "Search models, datasets, spaces, posts, docs, and profiles". Navigation links for "Models", "Datasets", "Spaces", "Posts", "Docs", and "Profiles" are in the center. On the left sidebar, the "Tasks" tab is selected, showing a list of tasks categorized by "Multimodal" (Visual Question Answering) and "Computer Vision" (Depth Estimation, Image Classification, Object Detection, Image Segmentation, Text-to-Image, Image-to-Text, Image-to-Image, Image-to-Video, Unconditional Image Generation, Video Classification, Text-to-Video, Zero-Shot Image Classification). The main content area displays the "Datasets" section with a count of 118,807. A "Filter by name" button and a "Full-text search" button are visible. A list of datasets is shown, including "storytracer/US-PD-Books", "microsoft/orca-math-word-problems-200k", "abacusai/SystemChat", "MarkrAI/KoCommercial-Dataset", and "bigcode/the-stack-v2". Each dataset entry includes a "Viewer" link, the update time, download count, and heart count.

**Hugging Face** Search models, datasets, spaces, posts, docs, and profiles

Models Datasets Spaces Posts Docs Profiles

**Tasks** Sizes Sub-tasks Languages Licenses Other

Filter Tasks by name

Multimodal

Visual Question Answering

Computer Vision

Depth Estimation Image Classification

Object Detection Image Segmentation

Text-to-Image Image-to-Text

Image-to-Image Image-to-Video

Unconditional Image Generation

Video Classification Text-to-Video

Zero-Shot Image Classification

**Datasets** 118,807 Filter by name new Full-text search

storytracer/US-PD-Books  
Viewer • Updated about 9 hours ago • 19 • 136

microsoft/orca-math-word-problems-200k  
Updated 9 days ago • 2.49k • 195

abacusai/SystemChat  
Viewer • Updated 9 days ago • 332 • 74

MarkrAI/KoCommercial-Dataset  
Viewer • Updated 1 day ago • 22 • 44







bigcode/the-stack-v2  
Viewer • Updated 9 days ago • 484 • 126

More than 100,000  
datasets









# Model Hub on Hugging Face

Models 2,735 meta Full-text search







-  [meta-llama/Meta-Llama-3-8B](#)  
Text Generation • Updated May 13 •  $\downarrow$  1.12M •  $\heartsuit$  4.84k
-  [meta-llama/Meta-Llama-3-8B-Instruct](#)  
Text Generation • Updated 23 days ago •  $\downarrow$  2.71M •  $\heartsuit$  2.72k
-  [meta-llama/Meta-Llama-3-70B-Instruct](#)  
Text Generation • Updated 23 days ago •  $\downarrow$  497k •  $\heartsuit$  1.19k
-  [eastwind/meta-chameleon-7b](#)  
Updated 2 days ago •  $\heartsuit$  23
-  [meta-llama/Llama-2-7b-chat-hf](#)  
Text Generation • Updated Apr 17 •  $\downarrow$  1.03M •  $\heartsuit$  3.63k
-  [meta-llama/Llama-2-7b-hf](#)

Models 651 openai Full-text search







-  [openai/whisper-large-v3](#)  
Automatic Speech Recognition • Updated 11 days ago •  $\downarrow$  3.4M •  $\heartsuit$  2.7
-  [openai-community/gpt2](#)  
Text Generation • Updated Feb 19 •  $\downarrow$  7.56M •  $\heartsuit$  2.03k
-  [openai/clip-vit-large-patch14](#)  
Zero-Shot Image Classification • Updated Sep 15, 2023 •  $\downarrow$  37.3M •  $\heartsuit$  1
-  [openai/clip-vit-base-patch32](#)  
Zero-Shot Image Classification • Updated Feb 29 •  $\downarrow$  17.4M •  $\heartsuit$  400
-  [openai/clip-vit-large-patch14-336](#)  
Zero-Shot Image Classification • Updated Oct 4, 2022 •  $\downarrow$  6.74M •  $\heartsuit$  14
-  [openai/whisper-large-v2](#)

# Model Hub on Hugging Face

Models 2,244 google Full-text search

-  **google-bert/bert-base-uncased**  
Fill-Mask • Updated Feb 19 • ⬇ 67.4M • ❤ 1.62k
-  **google/gemma-7b**  
Text Generation • Updated May 1 • ⬇ 193k • ❤ 2.96k
-  **google/recurrentgemma-9b**  
Text Generation • Updated 10 days ago • ⬇ 436 • ❤ 39
-  **google/flan-t5-base**  
Text2Text Generation • Updated Jul 17, 2023 • ⬇ 1.54M • ❤ 722
-  **google/gemma-2b**  
Text Generation • Updated Apr 16 • ⬇ 381k • ❤ 767
-  **google/gemma-2b-it**

Models 702 microsoft Full-text search

-  **microsoft/Florence-2-large**  
Image-to-Text • Updated 2 days ago • ⬇ 3.22k • ❤ 282
-  **microsoft/Florence-2-large-ft**  
Image-to-Text • Updated 2 days ago • ⬇ 1.73k • ❤ 125
-  **microsoft/Florence-2-base**  
Image-to-Text • Updated 2 days ago • ⬇ 738 • ❤ 50
-  **microsoft/Florence-2-base-ft**  
Image-to-Text • Updated 2 days ago • ⬇ 1.5k • ❤ 41
-  **microsoft/Phi-3-vision-128k-instruct**  
Text Generation • Updated 10 days ago • ⬇ 235k • ❤ 741
-  **microsoft/Phi-3-mini-4k-instruct**

# Sample Model from Hugging face


Use this model name  
in your code

This is “task”. It is an  
important parameter

Scroll down to see  
more details

Model Size.  
Anything more than  
100 M parameters is  
huge

# Model from Hugging face

google/gemma-7b  like 2.24k

[Text Generation](#) [Transformers](#) [Safetensors](#) [GGUF](#) [gemma](#) [Inference Endpoints](#) [text-generation](#)

[License: gemma-terms-of-use \(other\)](#)

[Model card](#) [Files](#) [Community](#) 69 [Train](#) [Deploy](#)

Click on Files and Community for additional resources

main v gemma-7b 13 contributors History: 11 commits + Contribute v

suryabhupa ybelkada <b>HF STAFF</b>	Update README.md (#51)	359f554	14 days ago
examples	update examples/notes		17 days ago
.gitattributes	1.62 kB	Squashing commit	21 days ago
README.md	21.1 kB	Update README.md (#51)	14 days ago
config.json	629 Bytes	Squashing commit	21 days ago
gemma-7b.gguf	34.2 GB	LFS Squashing commit	21 days ago
generation_config.json	428 Bytes	Squashing commit	21 days ago

Click and see some examples

# Model from Hugging face

⚡ Inference API ⓘ

📄 Text Generation

Example 4 ▼

Question: Give a four-line summary of WW1

Answer: The First World War was a global conflict that began in 1914 and lasted until 1918. It was fought between the Allies, which included the United Kingdom, France, Russia, and the United States, and the Central Powers, which included Germany, Austria-Hungary, and the Ottoman Empire. The war was marked by the use of new technologies, such as tanks and airplanes, and was characterized by massive loss of life and destruction. The war ended with the defeat

Compute

ctrl+Enter

0.5

Interreact with LLM immediately

Click on Compute multiple times

# Model from Hugging face

## Running the model on a CPU

```
from transformers import AutoTokenizer, AutoModelForCausalLM

tokenizer = AutoTokenizer.from_pretrained("google/gemma-7b-it")
model = AutoModelForCausalLM.from_pretrained("google/gemma-7b-it")

input_text = "Write me a poem about Machine Learning."
input_ids = tokenizer(input_text, return_tensors="pt")

outputs = model.generate(**input_ids)
print(tokenizer.decode(outputs[0]))
```

## Running the model on a single / multi GPU

```
# pip install accelerate
from transformers import AutoTokenizer, AutoModelForCausalLM

tokenizer = AutoTokenizer.from_pretrained("google/gemma-7b-it")
model = AutoModelForCausalLM.from_pretrained("google/gemma-7b-it")

input_text = "Write me a poem about Machine Learning."
input_ids = tokenizer(input_text, return_tensors="pt")

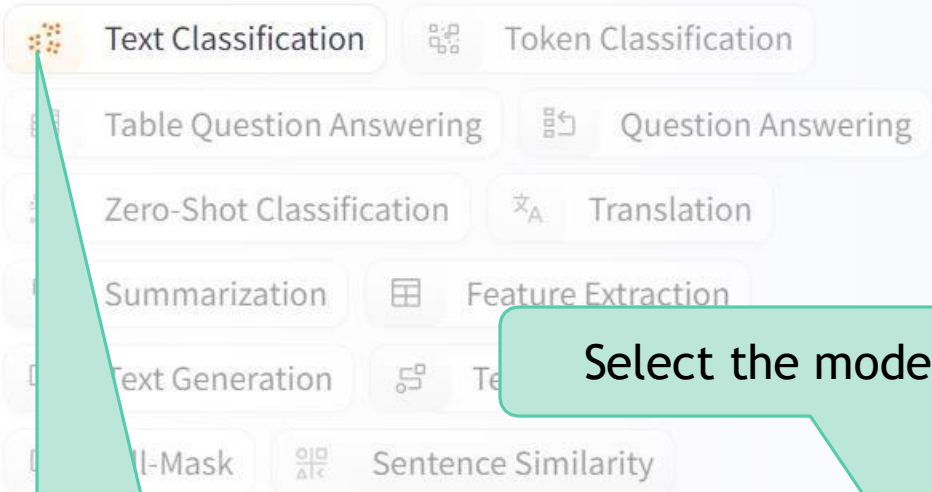
outputs = model.generate(**input_ids)
print(tokenizer.decode(outputs[0]))
```

Code samples  
provided as part of  
documentation



# Text Classification - sentiment analysis

Natural Language Processing



Select the task

Select the model

Models 64,133

Filter by name

Full-text search

Sort: Trending

RLHFlow/ArmoRM-Llama3-8B-v0.1

Text Classification • Updated 2 days ago • 6.38k • 69

BAAI/bge-reranker-v2-m3

Text Classification • Updated Mar 19 • 1.11M • 127

mrm8488/distilroberta-finetuned-financial-news-sentiment-an...

Text Classification • Updated Jan 21 • 4.56M • 269

HuggingFaceFW/fineweb-edu-classifier

Text Classification • Updated 16 days ago • 1.37M • 62

j-hartmann/emotion-english-distilroberta-base

Text Classification • Updated Jan 2, 2023 • 809k • 313

cardiffnlp/twitter-roberta-base-sentiment-latest

Text Classification • Updated May 28, 2023 • 11.2M • 452

# distilbert model for sentiment analysis

distilbert/distilbert-base-uncased-finetuned-sst-2-english

like 415

Text Classification

Transformers

PyTorch

TensorFlow

Rust

ONNX

Safetensors

sst2

glue

English

doi:10.57967/hf/0181

distilbert

Eval Results

Inference Endpoints

arxiv:1910.01108

License: apache-2.0

Model card

Files and versions

Community 29

Train

Deploy

Use in Transformers

DistilBERT base uncased finetuned SST-2

Downloads last month  
12,564,166

Safetensors

Model size 67M params

Tensor type F32

Inference API

Text Classification

Examples

This product is too good

Compute

Computation time on cpu: 0.336 s

POSITIVE 0.999

Table of Contents


- Model Details
- How to Get Started With the Model
- Uses
- Risks, Limitations and Biases
- Training

Model Details

Model Description: This model is a fine-tuned checkpoint

Sentiment

# deepset/roberta-base-squad2

deepset/**roberta-base-squad2**  like 582

Question Answering

Transformers

PyTorch

TensorFlow

JAX

Rust

Safetensors

squad\_v2

English

roberta

Eval Results

Inference Endpoints

License: cc-by-4.0

Model card

Files and versions

Community 17

⋮

Train

Deploy

Use in Transformers

Edit model card

## roberta-base for QA


This is the [roberta-base](#) model, fine-tuned using the [SQuAD2.0](#) dataset. It's been trained on question-answer pairs, including unanswerable questions, for the task of Question Answering.

### Overview

**Language model:** roberta-base  
**Language:** English  
**Downstream-task:** Extractive QA  
**Training data:** SQuAD 2.0  
**Eval data:** SQuAD 2.0  
**Code:** See [an example QA pipeline on Haystack](#)  
**Infrastructure:** 4x Tesla v100

..

Downloads last month  
**1,012,062**



Safetensors ⓘ

Model size 124M params

Tensor type F32 · I64

↗

⚡ Inference API ⓘ

Question Answering

Examples

Which name is also used to describe the Amazon rainfr

Compute

Context

The Amazon rainforest (Portuguese: Floresta Amazônica or Amazônia; Spanish: Selva Amazónica, Amazonía or usually Amazonia; French: Forêt amazonienne; Dutch: Amazoneregenwoud), also known in English as Amazonia or the Amazon Jungle, is a moist broadleaf forest that covers most of the Amazon basin of South America. This basin encompasses 7,000,000 square kilometres (2,700,000 sq mi), of which 5,500,000 square kilometres (2,100,000

# Hugging face pipeline()

## To get started quickly

```
from transformers import AutoTokenizer, AutoModelForCausalLM

tokenizer = AutoTokenizer.from_pretrained("google/gemma-7b")
model = AutoModelForCausalLM.from_pretrained("google/gemma-7b")

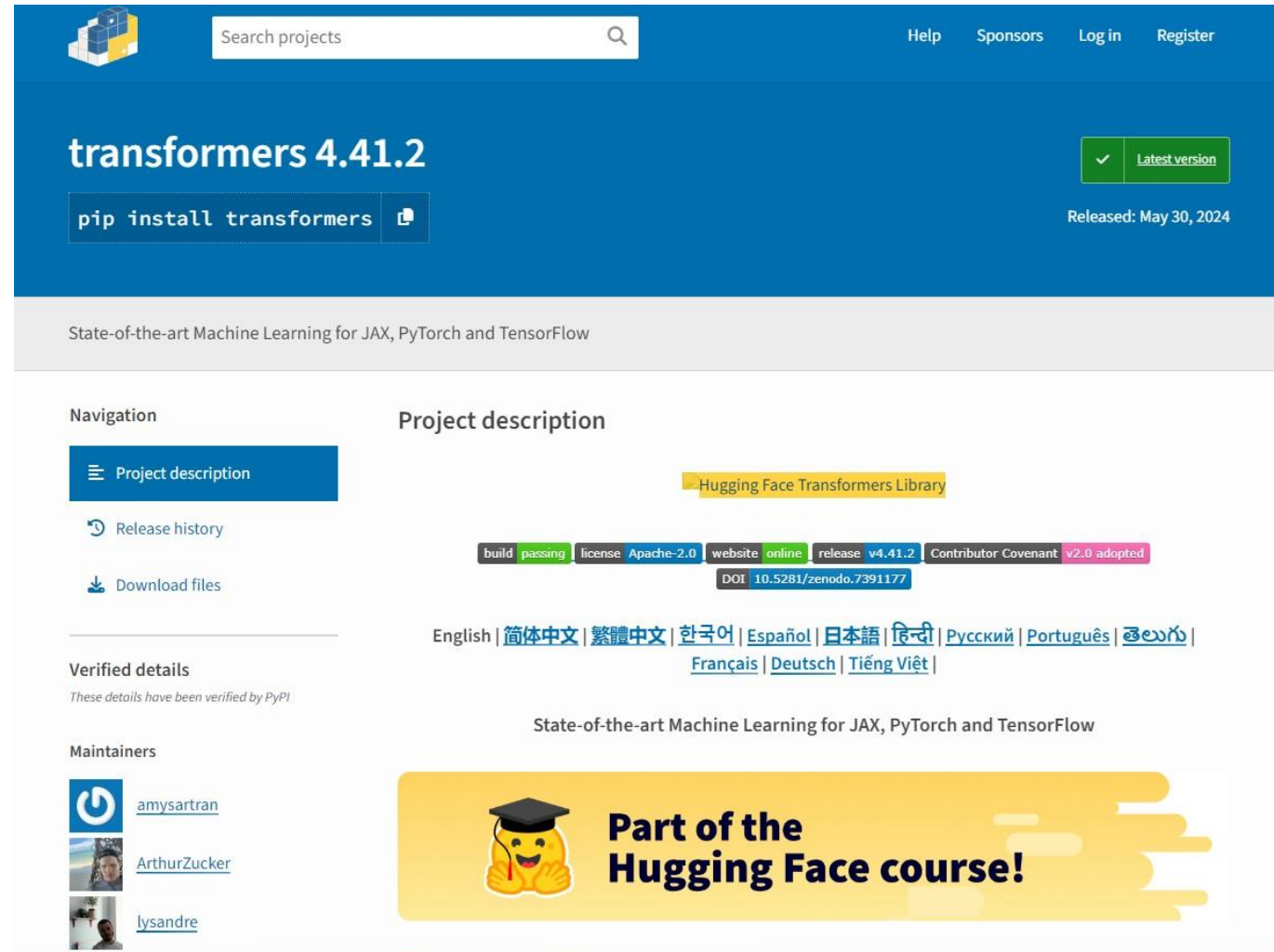
input_text = "Write me a poem about Machine Learning."
input_ids = tokenizer(input_text, return_tensors="pt")

outputs = model.generate(**input_ids)
print(tokenizer.decode(outputs[0]))
```

- This is the original code, but there is an easy way to call this model
- We will use pipeline() function

# About the transformers package

- Transformers is a community of projects and the Hugging Face Hub.
- It supports developers, researchers, students, professors, and engineers.
- Enables users to build their dream projects with ease.
- Provides thousands of pre-trained models for various tasks.
- Supports different modalities like text, vision, and audio.



The screenshot shows the Hugging Face Hub page for the transformers 4.41.2 package. The page has a blue header with a search bar, navigation links (Help, Sponsors, Log in, Register), and the package name 'transformers 4.41.2' with a 'Latest version' badge. Below the header, there's a 'pip install transformers' button and a release date of May 30, 2024. The main content area is divided into two columns: 'Navigation' on the left and 'Project description' on the right. The 'Navigation' column includes links for 'Project description', 'Release history', and 'Download files'. The 'Project description' column features the 'Hugging Face Transformers Library' badge, a row of status badges (build passing, license Apache-2.0, website online, release v4.41.2, Contributor Covenant v2.0 adopted, DOI 10.5281/zenodo.7391177), and a list of languages (English, 简体中文, 繁體中文, 한국어, Español, 日本語, हिन्दी, Русский, Português, తెలుగు, Français, Deutsch, Tiếng Việt). Below the languages, there's a 'Verified details' section with a note 'These details have been verified by PyPI' and a 'Maintainers' section listing amysartran, ArthurZucker, and lysandre. At the bottom right, there's a yellow banner with a graduation cap icon and the text 'Part of the Hugging Face course!'.

transformers 4.41.2

pip install transformers

Released: May 30, 2024

State-of-the-art Machine Learning for JAX, PyTorch and TensorFlow

Navigation

- Project description
- Release history
- Download files

Project description

Hugging Face Transformers Library

build passing license Apache-2.0 website online release v4.41.2 Contributor Covenant v2.0 adopted DOI 10.5281/zenodo.7391177

English | 简体中文 | 繁體中文 | 한국어 | Español | 日本語 | हिन्दी | Русский | Português | తెలుగు | Français | Deutsch | Tiếng Việt

State-of-the-art Machine Learning for JAX, PyTorch and TensorFlow

Verified details

These details have been verified by PyPI

Maintainers

- amysartran
- ArthurZucker
- lysandre

Part of the Hugging Face course!

# About the `pipeline()` function

- The pipeline package in the transformers library simplifies the process of using pre-trained NLP models
- It helps us by abstracting away complex steps like tokenization, model loading, and post-processing.
- Almost similar code can be used for multiple pre-trained models.
- The `pipeline()` function takes care of the necessary steps, including tokenizing input text, passing it through the model, and returning the results in a user-friendly format.

```
from transformers import pipeline
```



# Sentiment Analysis Model

```
senti_model = pipeline(task="sentiment-analysis")
```

- There are multiple models for sentiment analysis
- We have not specified any model name
- The default model will be considered

No model was supplied, defaulted to `distilbert/distilbert-base-uncased-finetuned-sst-2-english` and revision `af0f99b` (<https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english>).

Using a pipeline without specifying a model name and revision in production is not recommended.

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88: UserWarning:  
The secret `HF_TOKEN` does not exist in your Colab secrets.
```

To authenticate with the Hugging Face Hub, create a token in your settings tab (<https://huggingface.co/settings/tokens>), set it as secret in your Google Colab and restart your session.

You will be able to reuse this secret in all of your notebooks.

Please note that authentication is recommended but still optional to access public models or datasets.

```
warnings.warn(  

```

- Hugging face provides an API key. We have not mentioned it here.
- It still works for some models

# Sentiment Analysis Model - Result

```
senti_model("This movie is damn good. I loved it")
```

```
[{'label': 'POSITIVE', 'score': 0.9998770952224731}]
```

```
senti_model("This is a bad phone. The screen and battery are of poor quality.")
```

```
[{'label': 'NEGATIVE', 'score': 0.9998168349266052}]
```

# Choose your model

The screenshot shows the Hugging Face Models interface. On the left, the 'Tasks' tab is selected, showing a list of tasks under 'Natural Language Processing'. 'Text Classification' is highlighted. On the right, the 'Models' tab shows a list of models. A dropdown menu is open for the 'Sort' option, showing 'Trending', 'Most likes', 'Most downloads', 'Recently created', and 'Recently updated'. Three blue callout boxes provide instructions: one points to 'Text Classification' in the tasks list, another points to the first model 'mixedbread-ai/mxbai-rerank-large-v1', and a third points to the 'Sort' dropdown menu.

**Tasks** Libraries Datasets Languages  
Licenses Other

Filter Tasks by name Reset Tasks

Multimodal

Natural Language Processing

Text Classification Token Classification

Table Question Answering

Question Answering

Zero-Shot Classification Translation

Summarization Feature Extraction

Text Generation Text2Text Generation

Fill-Mask Sentence Similarity

Zero-Shot

Models 52,174 Filter by name

new Full-text search Sort: Trending

Trending

Most likes

Most downloads

Recently created

Recently updated

mixedbread-ai/mxbai-rerank-large-v1  
Text Classification • Updated 12 days ago • 25k • 48

cardiffnlp/twitter-roberta-base-sentiment-latest  
Text Classification • Updated May 28, 2023 • 94.4M • 332

mim8488/distilroberta-finetuned-financial-news-sentiment-analysis  
Text Classification • Updated Jan 21 • 30.9M • 201

BAAI/bge-reranker-large  
Text Classification • Updated 21 days ago • 369k • 180

FacebookAI/roberta-large  
Text Classification • Updated 24

ProsusAI/finbert  
Text Classification • Updated May 23, 2023 • 1M • 482

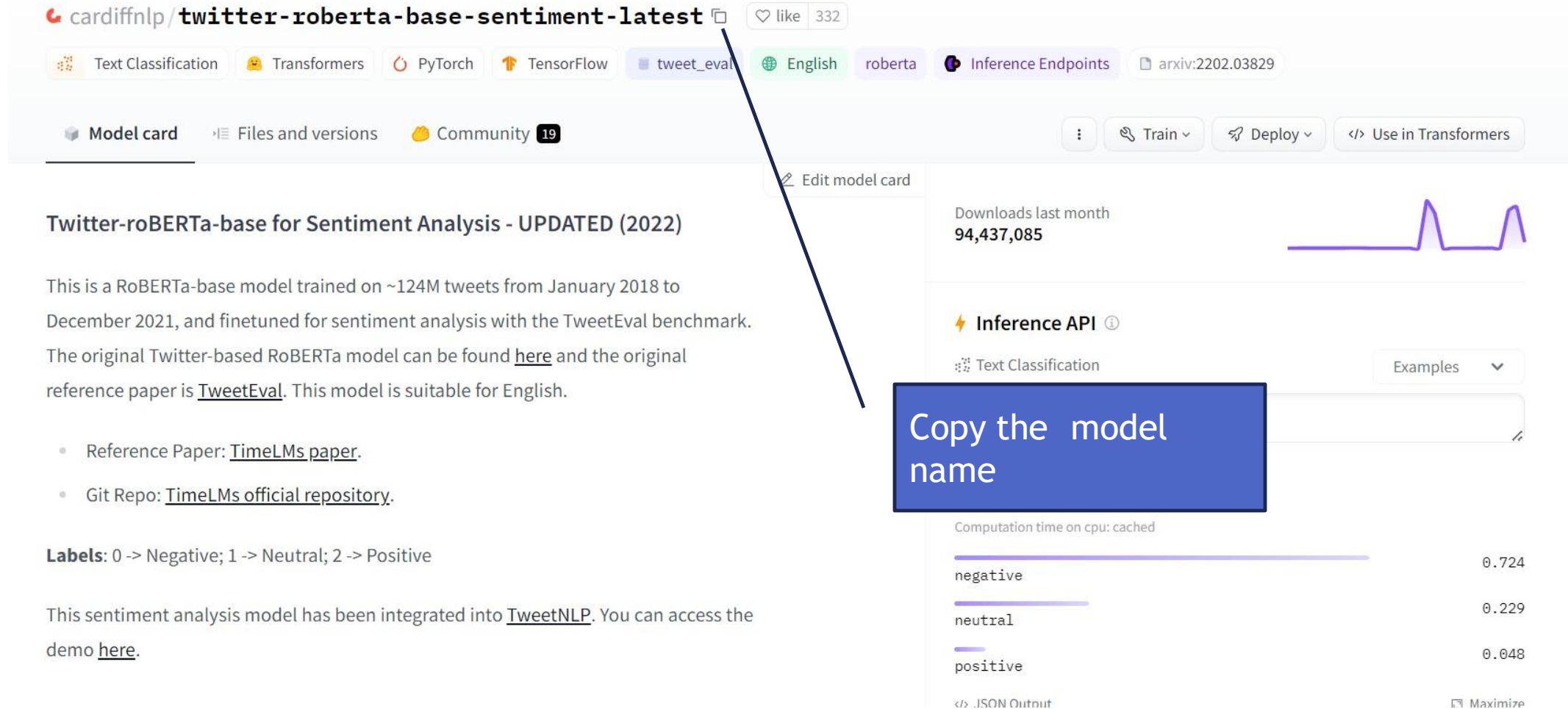
-base-multilingual-cased-sentiments-student  
Updated Jun 24, 2023 • 10.8M • 156

Several models available

- Select the task
- Sentiment Analysis comes under text classification

Sort and choose the most liked or most downloaded or any other model of your choice

# Choose your model



The screenshot shows the Hugging Face interface for the model 'cardiffnlp/twitter-roberta-base-sentiment-latest'. A blue callout box with the text 'Copy the model name' points to the model name in the header. The model card includes a description, a list of references, and a table of inference results.

**Twitter-roBERTa-base for Sentiment Analysis - UPDATED (2022)**

This is a RoBERTa-base model trained on ~124M tweets from January 2018 to December 2021, and finetuned for sentiment analysis with the TweetEval benchmark. The original Twitter-based RoBERTa model can be found [here](#) and the original reference paper is [TweetEval](#). This model is suitable for English.

- Reference Paper: [TimeLMs paper](#).
- Git Repo: [TimeLMs official repository](#).

**Labels:** 0 -> Negative; 1 -> Neutral; 2 -> Positive

This sentiment analysis model has been integrated into [TweetNLP](#). You can access the demo [here](#).

**Inference API**

Text Classification

Category	Score
negative	0.724
neutral	0.229
positive	0.048

Computation time on cpu: cached

Maximize



# Choose your model

```
Senti_model_2 = pipeline(task="sentiment-analysis",
                           model="cardiffnlp/twitter-roberta-base-sentiment-latest")
```

```
Senti_model_2("Over heating issue don't by this product camera was good")
```

```
[{'label': 'neutral', 'score': 0.4210317134857178}]
```

```
Senti_model_2("Waste of money")
```

```
[{'label': 'negative', 'score': 0.7434294819831848}]
```

```
Senti_model_2("Nice product under 24k .... overall good")
```

```
[{'label': 'positive', 'score': 0.9655607342720032}]
```

1 ★ Horrible

Waste of money

Avinash Kumar  Certified Buyer, Gurugram 6 days ago

5★ Super!

Nice product under 24k .... overall good 👍

Flipkart Customer  Certified Buyer, Habra 3 days ago

# Zero shot prediction

- Zero-shot prediction means asking the model to understand and carry out a task without having been given any specific examples of how to do it during its training.
- It depends on the LLM's broad understanding of language and concepts to make educated guesses.
- **Example:** If an LLM is given a task to classify customer reviews into categories like "service", "quality", "price", without ever being trained on these specific categories, it uses its general knowledge to infer which category a review belongs to based on the content of the review.



# Using this model on your dataset

```
import pandas as pd
user_review_data=pd.read_csv("https://raw.githubusercontent.com/...")
user_review_data=user_review_data.sample(50)
user_review_data["Review"]
```

```
23      I have yet to run this new battery below two b...
917          Leopard Print is wonderfully wild!.
243      No additional ear gels provided, and no instru...
1049     My side Greek salad with the Greek dressing wa...
235          One of my favorite purchases ever.
1245     -Drinks took close to 30 minutes to come out a...
348      This little device has transformed my organiza...
542      Perhaps my phone is defective, but people cann...
595      What possessed me to get this junk, I have no i...
1708     Service is quick and even "to go" orders are j...
304      Everything worked on the first try.The device ...
325          good item, low price.
489      This results in the phone being either stuck a...
949          I had to purchase a different case.
87       The construction of the headset is poor
```



Actual Reviews Data

# Sentiment Analysis on a Data frame

```
user_review_data["Predicted_Sentiment"] = user_review_data["Review"].apply(lambda x: Senti_model_2(x)[0]["label"])
user_review_data
```

	Id	Review	Sentiment	Predicted_Sentiment
0	1	So there is no way for me to plug it in here i...	0	negative
1	2	Good case, Excellent value.	1	positive
2	3	Great for the jawbone.	1	positive
3	4	Tied to charger for conversations lasting more...	0	negative
4	5	The mic is great.	1	positive
...	...	...	...	...
1995	1996	I think food should have flavor and texture an...	0	negative
1996	1997	Appetite instantly gone.	0	negative
✓ 1m 5s completed at 10:42AM				

Prediction takes a little longer if we load the model on CPU

# Load the model on GPU

```
Senti_model_2_gpu = pipeline(task="sentiment-analysis",
                             model="cardiffnlp/twitter-roberta-base-sentiment-latest",
                             device="cuda")
```

device="cuda" for GPU

	Id	Review	Sentiment	Predicted_Sentiment
0	1	So there is no way for me to plug it in here i...	0	negative
1	2	Good case, Excellent value.	1	positive
2	3	Great for the jawbone.	1	positive
3	4	Tied to charger for conversations lasting more...	0	negative
4	5	The mic is great.	1	positive
...	...	...	...	...
1995	1996	I think food should have flavor and texture an...	0	negative
1996	1997	Appetite instantly gone.	0	negative
1997	1998	Overall I was not impressed and would not go b...	0	negative

Takes less time to complete it

✓ 20s completed at 10:40 AM

# Other models on Hugging Face

## Computer Vision

Depth Estimation   Image Classification  
Object Detection   Image Segmentation  
Text-to-Image   Image-to-Text  
Image-to-Image   Image-to-Video  
Unconditional Image Generation  
Video Classification   Text-to-Video  
Zero-Shot Image Classification  
Mask Generation  
Zero-Shot Object Detection   Text-to-3D  
Image-to-3D   Image Feature Extraction

## Natural Language Processing

Text Classification   Token Classification  
Table Question Answering  
Question Answering  
Zero-Shot Classification   Translation  
Summarization   Feature Extraction  
Text Generation   Text2Text Generation  
Fill-Mask   Sentence Similarity

Models for various tasks.

## Audio

Text-to-Speech   Text-to-Audio  
Automatic Speech Recognition  
Audio-to-Audio   Audio Classification  
Voice Activity Detection

## Tabular

Tabular Classification   Tabular Regression

## Reinforcement Learning

Reinforcement Learning   Robotics

## Other

Graph Machine Learning



- 🎵 "audio-classification": Returns a **AudioClassificationPipeline**.
- 🗣️ "automatic-speech-recognition": Returns a **AutomaticSpeechRecognitionPipeline**.
- 💬 "conversational": Returns a **ConversationalPipeline**.
- 📏 "depth-estimation": Returns a **DepthEstimationPipeline**.
- 📞 "document-question-answering": Returns a **DocumentQuestionAnsweringPipeline**.
- Q "feature-extraction": Returns a **FeatureExtractionPipeline**.
- 🗑️ "fill-mask": Returns a **FillMaskPipeline**.
- 🖼️ "image-classification": Returns a **ImageClassificationPipeline**.
- 🖼️ Q "image-feature-extraction": Returns an **ImageFeatureExtractionPipeline**.
- 🖼️ "image-segmentation": Returns a **ImageSegmentationPipeline**.
- 🖼️ → 🖼️ "image-to-image": Returns a **ImageToImagePipeline**.
- 🖼️ 🗣️ "image-to-text": Returns a **ImageToTextPipeline**.
- 🗑️ "mask-generation": Returns a **MaskGenerationPipeline**.
- 🦾 "object-detection": Returns a **ObjectDetectionPipeline**.

- Some of the currently accepted tasks
- More on next page

- ? 🗣️ **"question-answering"**: Returns a **QuestionAnsweringPipeline**.
- 📄 Q **"summarization"**: Returns a **SummarizationPipeline**.
- 📄 ? **"table-question-answering"**: Returns a **TableQuestionAnsweringPipeline**.
- 🗣️ → 🗣️ **"text2text-generation"**: Returns a **Text2TextGenerationPipeline**.
- 🗣️ Q **"text-classification"** (alias **"sentiment-analysis"**): Returns a **TextClassificationPipeline**.
- 🗣️ → **"text-generation"**: Returns a **TextGenerationPipeline**.
- 🗣️ → 🔊 **"text-to-audio"** (alias **"text-to-speech"**): Returns a **TextToAudioPipeline**.
- 🔍 **"token-classification"** (alias **"ner"**): Returns a **TokenClassificationPipeline**.
- 🌐 → 10 **"translation"**: Returns a **TranslationPipeline**.
- - **"translation xx to yy"**: Returns a **TranslationPipeline**.
- 📺 Q **"video-classification"**: Returns a **VideoClassificationPipeline**.
- 🖼️ **"visual-question-answering"**: Returns a **VisualQuestionAnsweringPipeline**.
- 🎯 🗣️ **"zero-shot-classification"**: Returns a **ZeroShotClassificationPipeline**.
- 🎯 🖼️ **"zero-shot-image-classification"**: Returns a **ZeroShotImageClassificationPipeline**.
- 🎯 🎵 **"zero-shot-audio-classification"**: Returns a **ZeroShotAudioClassificationPipeline**.
- 🎯 🖼️ **"zero-shot-object-detection"**: Returns a **ZeroShotObjectDetectionPipeline**.

We are interested in these NLP based tasks



Python 3 Google Compute Engine backend (GPU)

Showing resources from 10:18 AM to 12:01 PM

System RAM  
6.8 / 51.0 GB



GPU RAM  
5.8 / 15.0 GB



Disk  
28.0 / 201.2 GB



Clear cache in GPU

```
import torch  
torch.cuda.empty_cache()
```

Python 3 Google Compute Engine backend (GPU)

Showing resources from 10:18 AM to 12:02 PM

System RAM  
6.9 / 51.0 GB



GPU RAM  
1.1 / 15.0 GB



Disk  
28.0 / 201.2 GB



# Language Translation Model

```
translator_model = pipeline(task="translation_en_to_fr",  
                             model="google-t5/t5-small",  
                             device="cuda")  
translator_model("Good bye")
```

```
[{'translation_text': 'Bon droit'}]
```

# Q&A Model

```
qa_model = pipeline(task="question-answering",  
                    model="deepset/roberta-base-squad2",  
                    device="cuda")
```

Our model

```
#Importing computer_scientists.txt document from github  
!wget https://raw.githubusercontent.com/venkatarreddykonasani/  
document=open("computer_scientists.txt").read()
```

Input Document

# Q&A based on a document

```
qa_model({'question': "What did Yann LeCun contribute?",  
| | | | | 'context': document})
```

```
{ 'score': 0.24149088561534882,  
  'start': 330,  
  'end': 377,  
  'answer': 'Revolutionized AI for image and text processing' }
```

```
qa_model({'question': "Who is the father of deep learning?",  
| | | | | 'context': document})
```

```
{ 'score': 0.7754606008529663,  
  'start': 1421,  
  'end': 1444,  
  'answer': 'Geoffrey Everest Hinton' }
```

# NER Model

```
ner_model = pipeline(task="ner",  
                      model="dslim/bert-base-NER",  
                      device="cuda",  
                      aggregation_strategy="simple")  
#aggregation_strategy = "Simple" ; simplifies the output and makes it easy to read
```

# NER Model

```
sample_doc="""
```

```
Hello,
```

```
I, John Smith, a member of the Tech Innovators team, would like to schedule a meeting with you,  
Mary Johnson, from the Quantum Solutions group, on Tuesday, February 8th, 2024, at 10:00 AM.  
We can meet at your office in San Francisco or, if more convenient, at the Cafe Bella in New York City.  
Please let me know if this date and time work for you.
```

```
"""
```

NER output

```
entities = ner_model(sample_doc)
```

```
print(entities)
```

```
[{'entity_group': 'PER', 'score': 0.9994373, 'word': 'John Smith', 'start': 13, 'end': 23}, {'entity_group': 'ORG', 'score':  
0.9968871, 'word': 'Tech Innovators', 'start': 41, 'end': 56}, {'entity_group': 'PER', 'score': 0.99906284, 'word': 'Mary  
Johnson', 'start': 108, 'end': 120}, {'entity_group': 'ORG', 'score': 0.9988469, 'word': 'Quantum Solutions', 'start': 131,  
'end': 148}, {'entity_group': 'LOC', 'score': 0.9993695, 'word': 'San Francisco', 'start': 233, 'end': 246},  
{ 'entity_group': 'ORG', 'score': 0.63540447, 'word': 'Cafe Bella', 'start': 278, 'end': 288}, {'entity_group': 'LOC',  
'score': 0.9994677, 'word': 'New York City', 'start': 292, 'end': 305}]
```



# NER Model

```
# Convert the above output into a dataframe and print it with the entity name
NER_result = pd.DataFrame(entities, columns=["word", "entity_group"])

# Print the DataFrame
print(NER_result)
```

	word	entity_group
0	John Smith	PER
1	Tech Innovators	ORG
2	Mary Johnson	PER
3	Quantum Solutions	ORG
4	San Francisco	LOC
5	Cafe Bella	ORG
6	New York City	LOC



Output stored in a data frame

# Text Summarization Model

```
summarizer_model = pipeline(task="summarization",  
                             model="google/pegasus-xsum",  
                             device="cuda")
```

```
Book_essay = """  
The 7 Habits of Highly Effective People" is a timeless self-help book by Ste  
Covey's philosophy centers on the idea that true success is achieved by alig  
The next three habits delve into the concept of interdependence, emphasizing  
The seventh habit, "Sharpen the Saw," encourages continuous self-renewal and  
Throughout the book, Covey provides practical advice and real-life examples  
"""
```

```
print(summarizer_model(Book_essay, max_length=120, min_length=30))
```

```
[{'summary_text': '"The 7 Habits of Highly Effective People" is a timeless self-help book by Stephen R. Covey that offers  
holistic approach to personal and professional effectiveness.'}]
```

# Text Generation Model

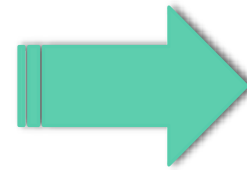
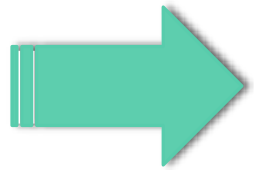
```
text_generator_model = pipeline(task="text-generation",  
                                model="gpt2",  
                                device="cuda")
```

```
# Generate text starting with the given prompt  
text_result = text_generator_model("The best way to start a presentation is")  
print(text_result)
```

Setting `pad\_token\_id` to `eos\_token\_id`:50256 for open-end generation.

```
[{'generated_text': 'The best way to start a presentation is to find the most exciting thing about any subject imaginable—  
how exciting a subject it is, and the value it creates," says Mark Bivens, a researcher as well as a research associate at  
the Massachusetts Institute'}]
```

# pipeline() function



Raw input

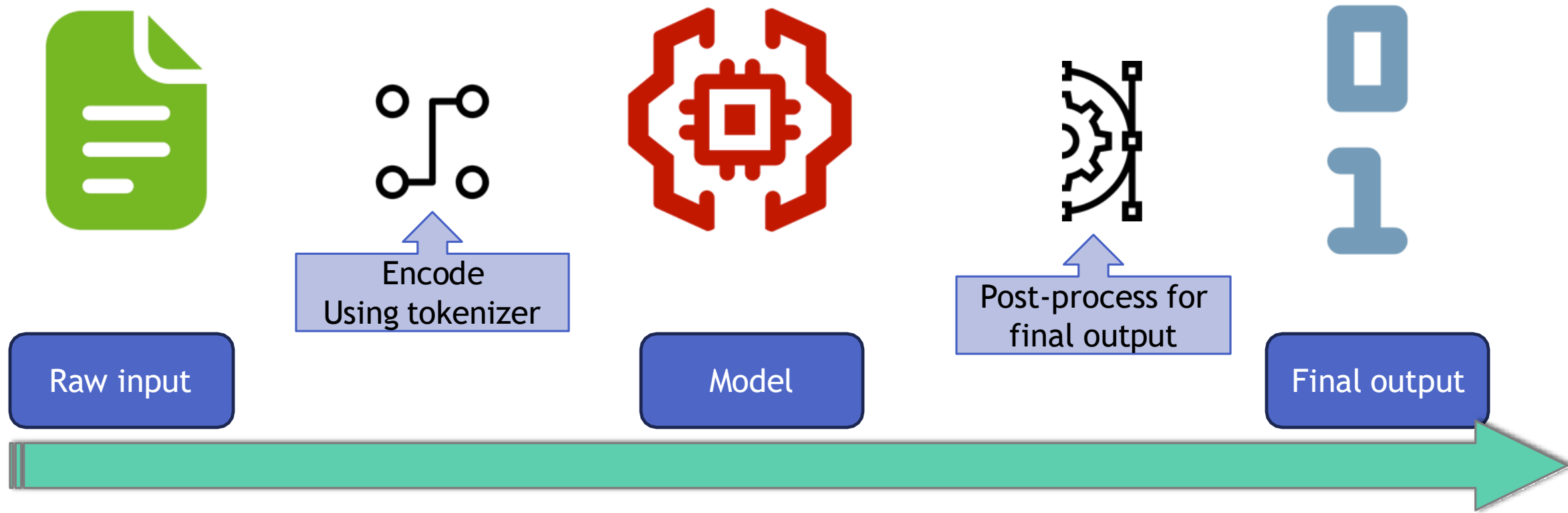
Model

Final output

```
Senti_model_2 = pipeline(task="sentiment-analysis",  
                          model="cardiffnlp/twitter-roberta-base-sentiment-latest")
```

```
Senti_model_2("Over heating issue don't by this product camera was good")
```

# Without pipeline() function



- We have to convert raw input text into encoded values using a tokenizer
- Get the output from the model. The output will be in the form of tensors. We need to process it to display the original classes

# Without pipeline() function

```
from transformers import AutoTokenizer, AutoModelForSequenceClassification

tokenizer = AutoTokenizer.from_pretrained("cardiffnlp/twitter-roberta-base-sentiment")

model = AutoModelForSequenceClassification.from_pretrained("cardiffnlp/twitter-roberta-base-sentiment")
```

- Every model has its own tokenizer function.
- Auto tokenizer and Auto model functions work for almost all the models



# Without pipeline() function

```
from transformers import AutoTokenizer, AutoModelForSequenceClassification

tokenizer = AutoTokenizer.from_pretrained("cardiffnlp/twitter-roberta-base-sentiment")

model = AutoModelForSequenceClassification.from_pretrained("cardiffnlp/twitter-roberta-base-sentiment")
```

```
import numpy as np
raw_text = "This is a great book"
encoded_input = tokenizer(raw_text, return_tensors='pt')
output = model(**encoded_input)
logits = output.logits.detach().numpy()
y_pred = np.argmax(logits)
y_pred
```

- Model output will be in the form of tensors
- This post processing step will extract the logits from the output
- We use those logits to decide our final predicted class

- Labels: 0 -> Negative; 1 -> Neutral; 2 -> Positive