

# Hugging Face

Hugging Face is a company and an open-source community focused on Natural Language Processing (NLP). It provides a platform for accessing, sharing, and collaborating on state-of-the-art NLP models and tools.

**Transformers Library:** Hugging Face's Transformers library is one of the most popular open-source libraries for NLP. It provides a wide range of pre-trained models (such as BERT, GPT, RoBERTa) and tools for working with them, including model architectures, tokenizers, and fine-tuning scripts.

First we will see how to use pre trained models from hugging face

## **Step-1:**

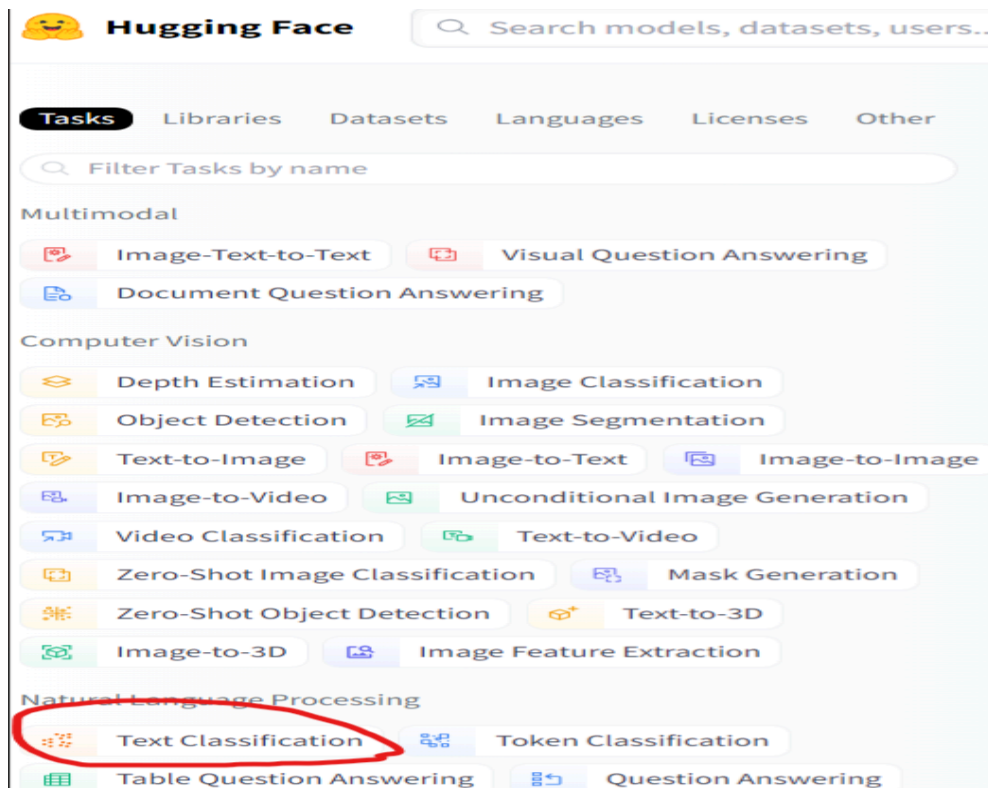
Navigate to the website

<https://huggingface.co/>.

## **Step-2:**

Once you're on the website, locate and click on the "Models" section from the menu.

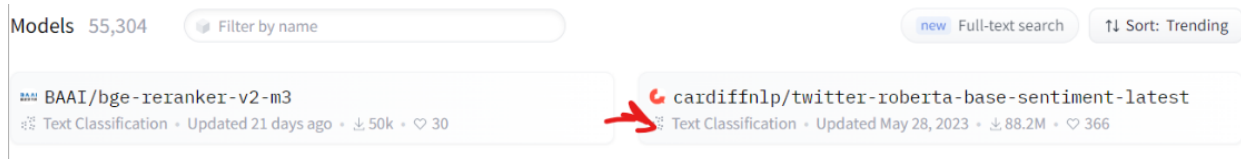
### Step-3:



On the left side there are different tasks available we will be choosing the text classification task

### Step-4:

After clicking that we can choose a model for task from a wide range of pre-trained models



I will be choosing this cardiff nlp model for sentiment analysis

## Step-5:

On the top right click on use in transformers to generate a code which will help us use the selected model in python

How to use from the **Transformers** library

```
# Use a pipeline as a high-level helper
from transformers import pipeline

pipe = pipeline("text-classification", model="cardiffnlp/twitter-roberta-base-sentiment-latest")

# Load model directly
from transformers import AutoTokenizer, AutoModelForSequenceClassification

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

Copy and run the first code which will help us in using a pipeline to use the model.

In Hugging Face's Transformers library, a pipeline is a high-level interface that simplifies the process of using pre-trained models for various NLP tasks. It abstracts away many of the complexities involved in loading models, tokenizing input, running inference, and post-processing results.

With a pipeline, users can perform common NLP tasks such as text generation, text classification, named entity recognition, question answering, and more with just a few lines of code.

## Step-6:

Use the code in python

```
In [1]: from transformers import pipeline
        model = pipeline("text-classification", model="cardiffnlp/twitter-roberta-base-sentiment-latest")

...

In [2]: model.predict("I liked the movie it was awesome")

Out[2]: [{'label': 'positive', 'score': 0.9864270091056824}]
```

**We have successfully performed predictions for our sentence.**

**Label:** The label represents the predicted sentiment category for the input text. In sentiment analysis, common labels include "positive," "negative," and "neutral." The label indicates the sentiment polarity assigned to the input text based on the model's prediction.

**Score:** The score is a numerical value associated with the predicted label. It represents the confidence or probability assigned by the model to the predicted sentiment category. Higher scores indicate higher confidence in the prediction, while lower scores suggest lower confidence.