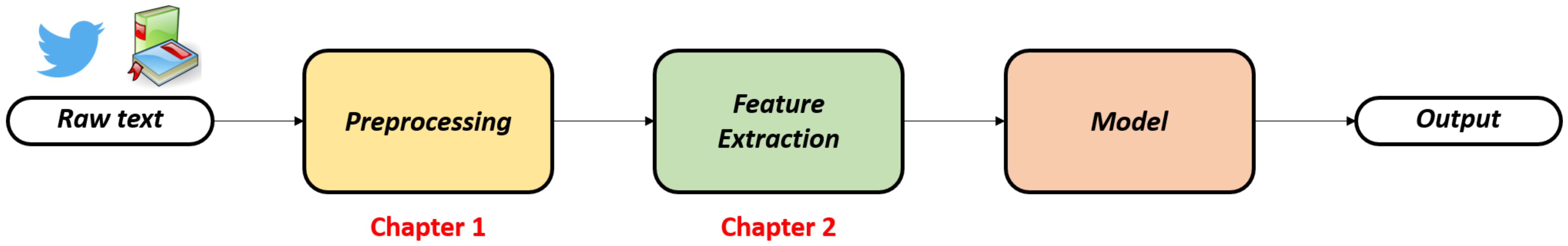
Hugging Face pipelines for sentiment analysis

NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON

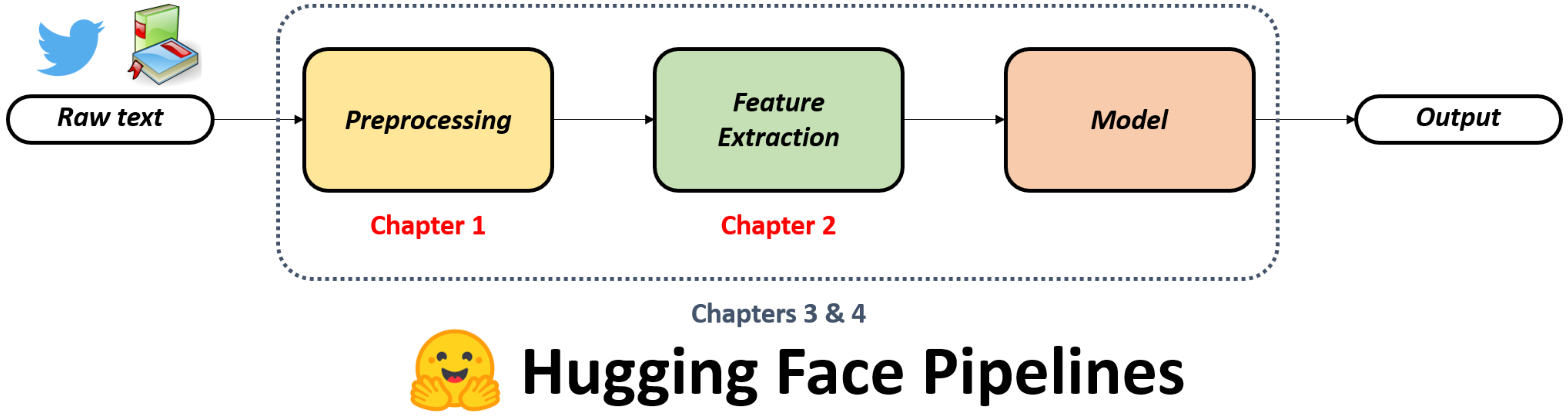
Fouad Trad
Machine Learning Engineer

datacamp

Recap: NLP workflow



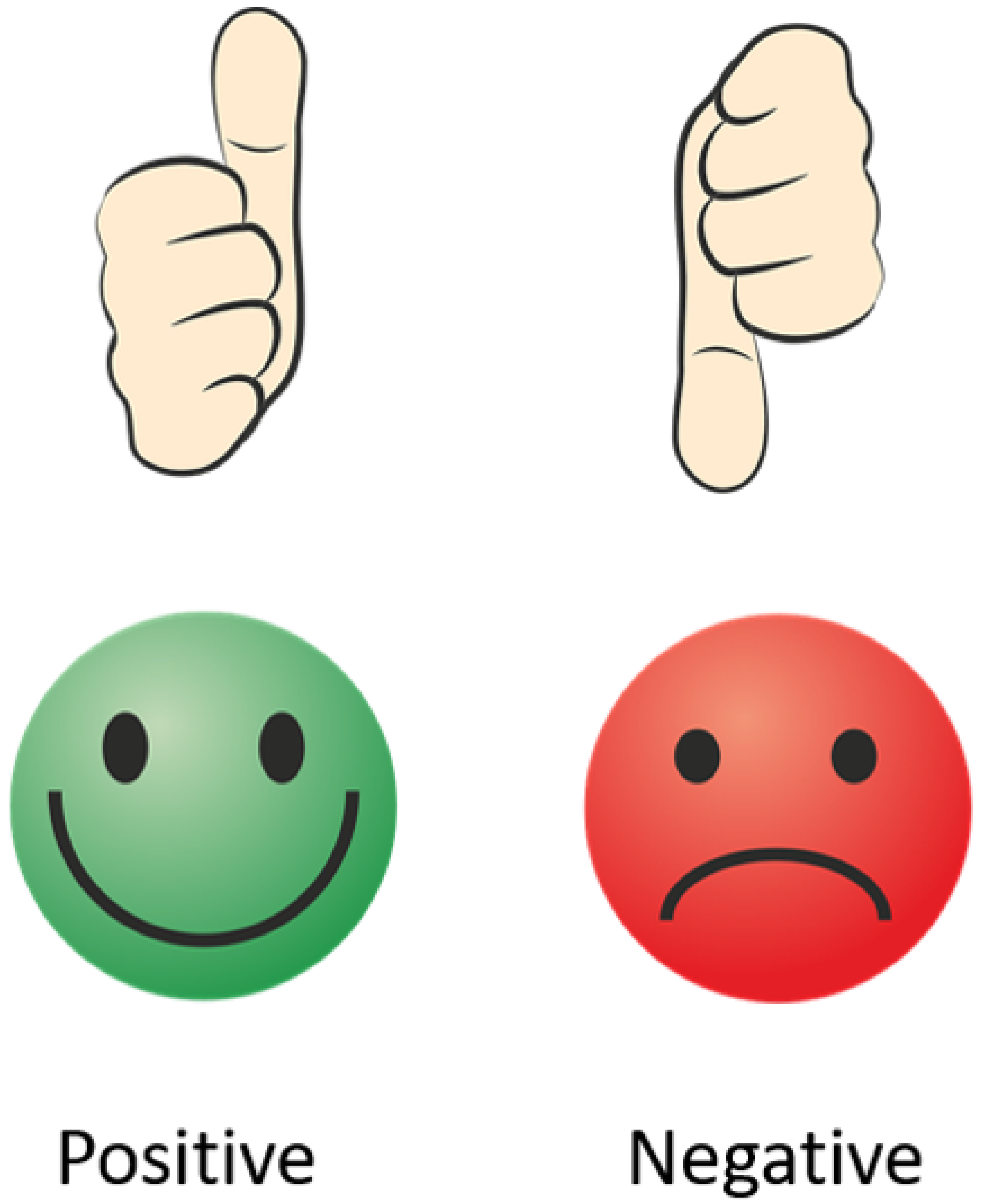
Hugging Face pipelines



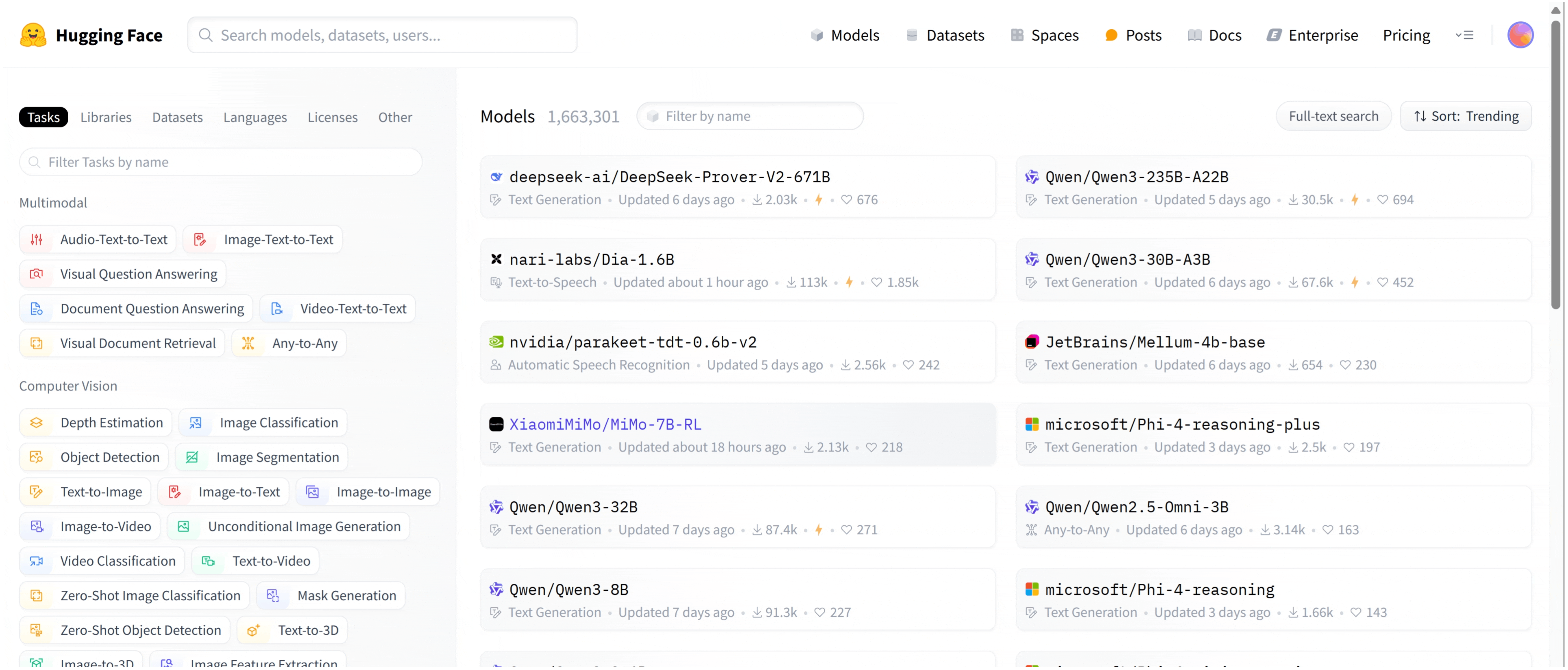
- Ready-made workflow that handles all steps in a function call
- Defining a pipeline requires:
  - NLP task
  - Model to perform the task

Pipelines for sentiment analysis

- Text classification task
- Predicts if text expresses positive or negative emotion



Models for text classification



deepseek-ai/DeepSeek-Prover-V2-671B
Text Generation • Updated 6 days ago • 2.03k • • 676

Qwen/Qwen3-235B-A22B
Text Generation • Updated 5 days ago • 30.5k • • 694

nari-labs/Dia-1.6B
Text-to-Speech • Updated about 1 hour ago • 113k • • 1.85k

Qwen/Qwen3-30B-A3B
Text Generation • Updated 6 days ago • 67.6k • • 452

nvidia/parakeet-tdt-0.6b-v2
Automatic Speech Recognition • Updated 5 days ago • 2.56k • 242

JetBrains/Mellum-4b-base
Text Generation • Updated 6 days ago • 654 • 230

XiaomiMiMo/MiMo-7B-RL
Text Generation • Updated about 18 hours ago • 2.13k • 218

microsoft/Phi-4-reasoning-plus
Text Generation • Updated 3 days ago • 2.5k • 197

Qwen/Qwen3-32B
Text Generation • Updated 7 days ago • 87.4k • • 271

Qwen/Qwen2.5-Omni-3B
Any-to-Any • Updated 6 days ago • 3.14k • 163

Qwen/Qwen3-8B
Text Generation • Updated 7 days ago • 91.3k • 227

microsoft/Phi-4-reasoning
Text Generation • Updated 3 days ago • 1.66k • 143

[1] https://huggingface.co/models

Pipelines in code

```python
from transformers import pipeline

classification_pipeline = pipeline(
  task="sentiment-analysis", # or text-classification
  model="distilbert/distilbert-base-uncased-finetuned-sst-2-english"
)
result = classification_pipeline("I really liked the movie!!")
print(result)
```

```
[{'label': 'POSITIVE', 'score': 0.9998093247413635}]
```

Sentiment analysis on a batch of texts

```python
texts = ["I really liked the movie!!",
         "Great job ruining my day.",
         "This product exceeded my expectations.",
         "Wow, just what I needed... another problem.",
         "Absolutely fantastic experience!"]
results = classification_pipeline(texts)
print(results)
```

```
[{'label': 'POSITIVE', 'score': 0.9998093247413635},
 {'label': 'NEGATIVE', 'score': 0.8666700124740601},
 {'label': 'POSITIVE', 'score': 0.998874843120575},
 {'label': 'POSITIVE', 'score': 0.98626708984375},
 {'label': 'POSITIVE', 'score': 0.9998812675476074}]
```

Assessing sentiment analysis models

```python
texts = ["I really liked the movie!!",
         "Great job ruining my day.",
         "This product exceeded my expectations.",
         "Wow, just what I needed... another problem.",
         "Absolutely fantastic experience!"]
true_labels = ["POSITIVE", "NEGATIVE", "POSITIVE", "NEGATIVE", "POSITIVE"]
results = classification_pipeline(texts)
predicted_labels = [result['label'] for result in results]
```

```python
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(true_labels, predicted_labels)
print(f"Accuracy: {accuracy}")
```

```
Accuracy: 0.80
```

Let's practice!

NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON

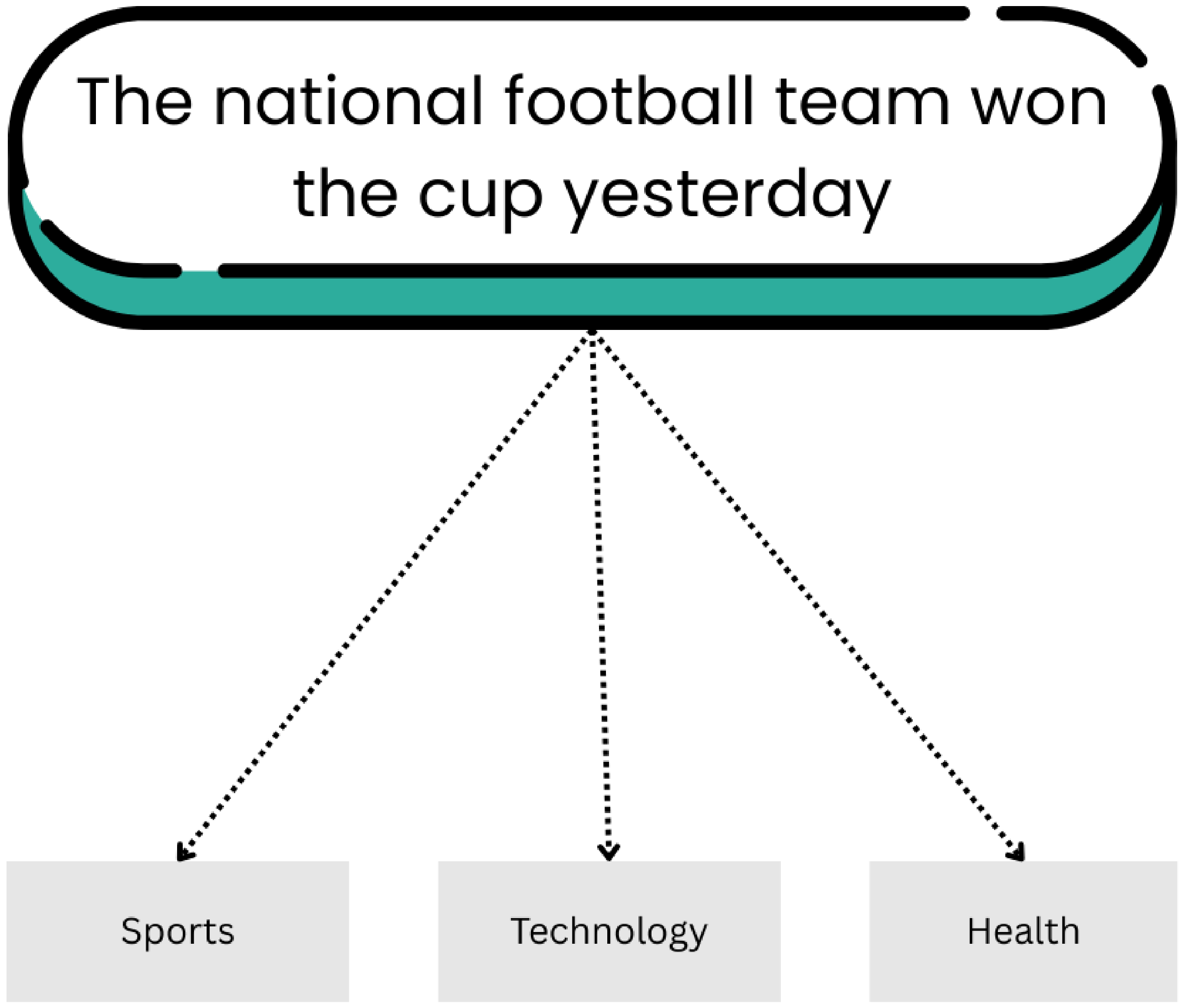
Zero-shot classification and QNLI

NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON

Fouad Trad
Machine Learning Engineer

Zero-shot classification

- Allows model to assign text to labels it hasn't seen before
- Uses natural language prediction get the output
- Useful for:
  - Content tagging
  - Customer support
  - Filtering news articles



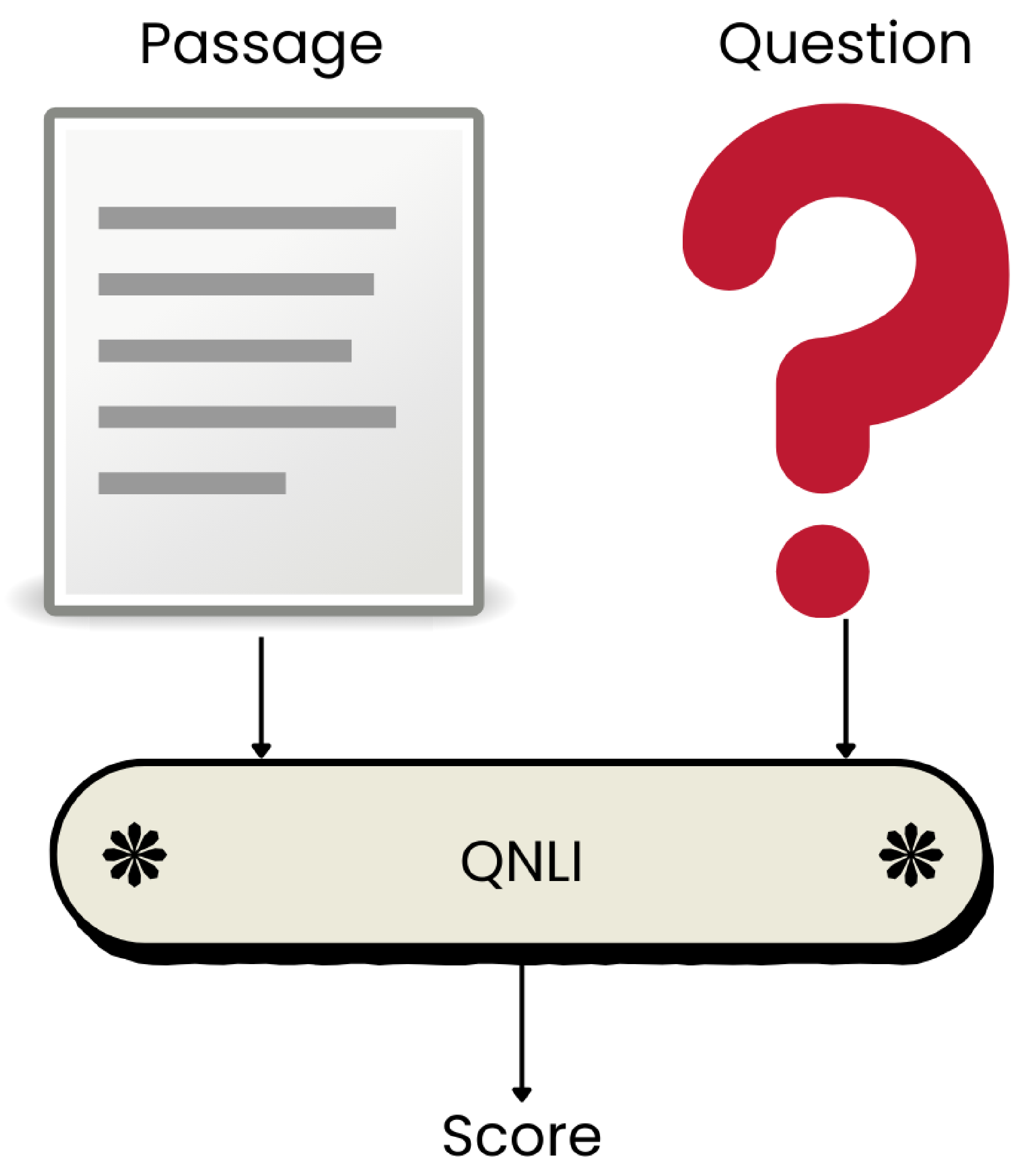
Zero-shot classification pipeline

```python
from transformers import pipeline
zero_shot_classifier = pipeline(
  task="zero-shot-classification",
  model="MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli"
)
text = "The national football team won the cup yesterday."
candidate_labels = ["sports", "technology", "health"]
result = zero_shot_classifier(text, candidate_labels)
print(result)
```

```
{'sequence': 'The national football team won the cup yesterday.',
 'labels': ['sports', 'technology', 'health'],
 'scores': [0.9948731064796448, 0.0029330444522202015, 0.0021938229911002934]}
```

Question natural language inference (QNLI)

- Determines whether the answer to a question can be found in a passage
- Useful for:
  - Document search
  - Chatbots
  - Information retrieval



QNLI pipeline

```python
from transformers import pipeline
qnli_pipeline = pipeline(
    task="text-classification",
    model="cross-encoder/qnli-electra-base"
    )
passage = "Penguins are found primarily in the Southern Hemisphere."
question = "Where do penguins live?"
result = qnli_pipeline({"text": question, "text_pair": passage})
print(result)
```

```
{'label': 'LABEL_0', 'score': 0.9951545000076294}
```

QNLI pipeline

```python
from transformers import pipeline
qnli_pipeline = pipeline(
    task="text-classification",
    model="cross-encoder/qnli-electra-base"
    )
passage = "Penguins are found primarily in the Southern Hemisphere."
question = "What is the capital of Paris?"
result = qnli_pipeline({"text": question, "text_pair": passage})
print(result)
```

```
{'label': 'LABEL_0', 'score': 0.008907231502234936}
```

Let's practice!

NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON

Question similarity and grammatical correctness

NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON

Fouad Trad
Machine Learning Engineer

datacamp

Question similarity

- Identifies when two questions are paraphrases
- Useful for:
  - Deduplication
  - Clustering similar questions
  - Improving search accuracy
- Done with models trained on the Quora Question Pairs (QQP) dataset

QQP pipeline

```python
from transformers import pipeline
qqp_pipeline = pipeline(
    task="text-classification",
    model="textattack/bert-base-uncased-QQP"
    )
question1 = "How can I learn Python?"
question2 = "What is the best way to study Python?"
result = qqp_pipeline({"text": question1, "text_pair": question2})
print(result)
```

```
{'label': 'LABEL_1', 'score': 0.6853412985801697}
```

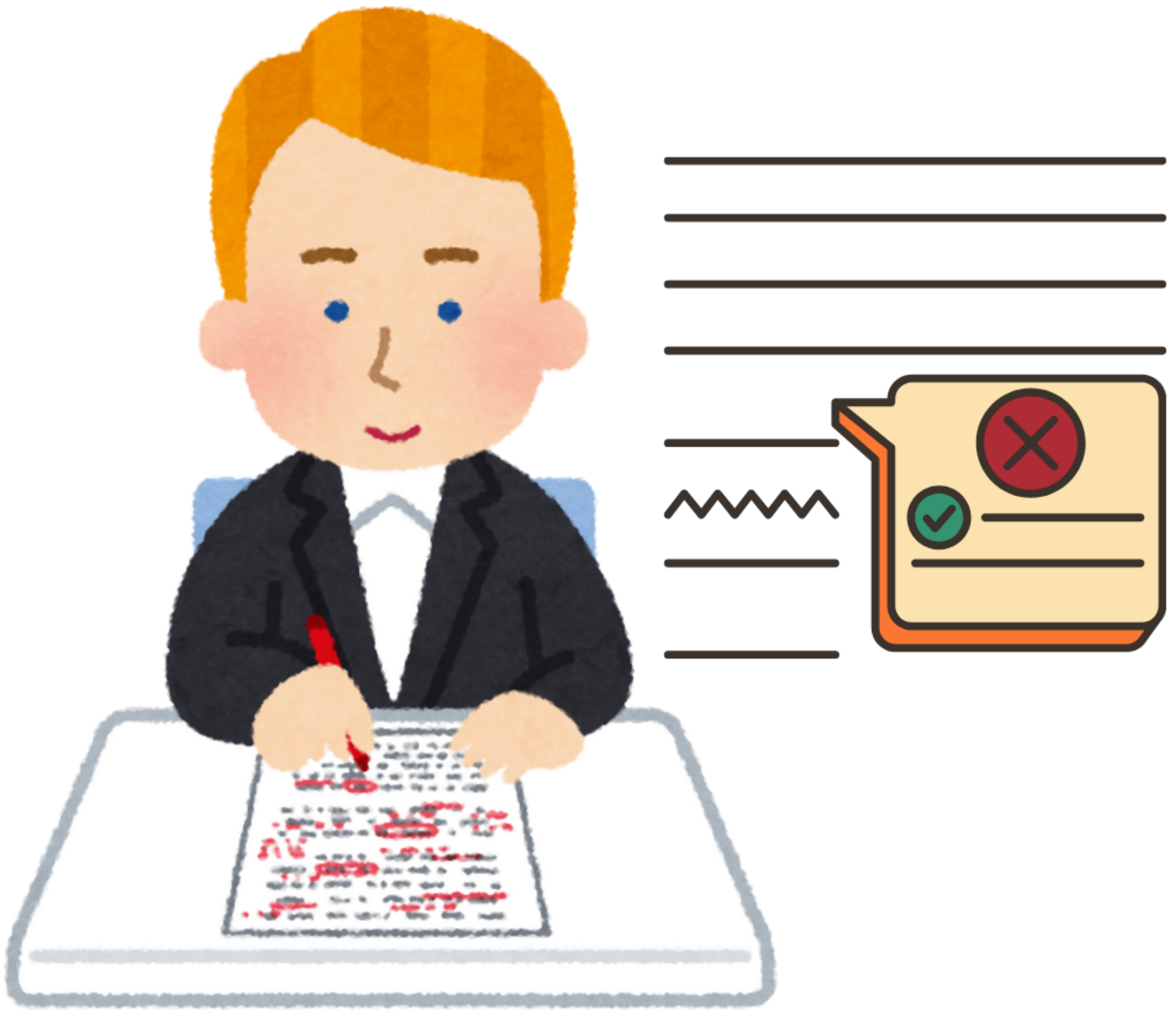
QQP pipeline

```python
from transformers import pipeline
qqp_pipeline = pipeline(
    task="text-classification",
    model="textattack/bert-base-uncased-QQP"
    )
question1 = "How can I learn Python?"
question2 = "What is the capital of France?"
result = qqp_pipeline({"text": question1, "text_pair": question2})
print(result)
```

```
{'label': 'LABEL_0', 'score': 0.9999338388442993}
```

Assessing grammatical correctness

- Assess how much a text is grammatically correct
- Useful for:
  - Educational tools
  - Grammar checkers
  - Writing assistants
- Done with models trained on the Corpus of Linguistic Acceptability (CoLA) dataset



CoLA pipeline

```python
from transformers import pipeline
cola_classifier = pipeline(
  task="text-classification",
  model="textattack/distilbert-base-uncased-CoLA"
)
result = cola_classifier("The cat sat on the mat.")
print(result)
```

```
[{'label': 'LABEL_1', 'score': 0.9918296933174133}]
```

CoLA pipeline

```python
from transformers import pipeline
cola_classifier = pipeline(
  task="text-classification",
  model="textattack/distilbert-base-uncased-CoLA"
)
result = cola_classifier("The cat on sat mat the.")
print(result)
```

```
[{'label': 'LABEL_0', 'score': 0.9628171324729919}]
```

Let's practice!

NATURAL LANGUAGE PROCESSING (NLP) IN PYTHON