

# 06

## 基础组件之Evaluate

- (1) Evaluate简介
- (2) Evaluate基本使用
- (3) Evaluate模型微调代码优化

# 基础组件之Evaluate

## Evaluate

- 简介

- evaluate库是一个非常简单易用的机器学习模型评估函数库，只需要一行代码便可以加载各种任务的评估函数

- 函数库地址

- <https://huggingface.co/evaluate-metric>

- 文档地址

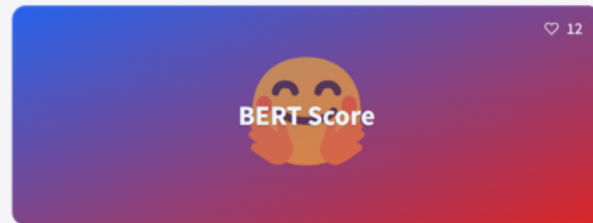
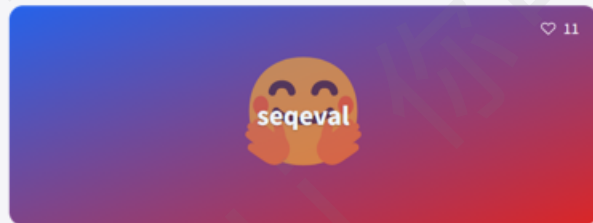
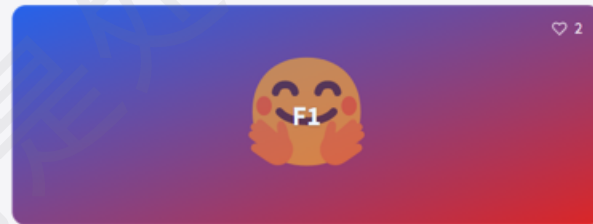
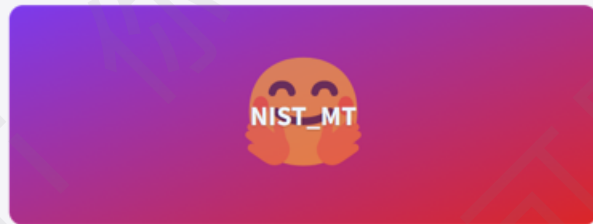
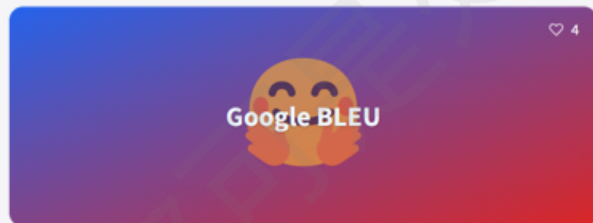
- <https://huggingface.co/docs/evaluate/index>



# 基础组件之Evaluate

## Evaluate基本使用

- 查看支持的评估函数 (list\_evaluation\_modules)
- 加载评估函数 (load)
- 查看评估函数说明 (inputs\_description)
- 评估指标计算 (compute)
  - 全局计算 (compute)
  - 迭代计算 (add / add\_batch)
- 计算多个评估指标 (combine)
- 评估结果对比可视化 (radar\_plot)



# 基础组件之Evaluate

## 模型微调代码优化

- 任务类型
  - 文本分类
- 使用模型
  - hfl/rbt3
- 优化内容
  - Evaluate使用多个评估函数

## Step7 训练与验证

```
import evaluate

clf_metric = evaluate.combine(["accuracy", "f1", "recall", "precision"])

def evaluate():
    model.eval()
    with torch.inference_mode():
        for batch in validloader:
            if torch.cuda.is_available():
                batch = {k: v.cuda() for k, v in batch.items()}
            output = model(**batch)
            pred = torch.argmax(output.logits, dim=-1)
            clf_metric.add_batch(pred.long(), batch["labels"].long())
    return clf_metric.compute()

def train(epoch=3, log_step=100):
    global_step = 0
    for ep in range(epoch):
        model.train()
        for batch in trainloader:
            if torch.cuda.is_available():
                batch = {k: v.cuda() for k, v in batch.items()}
            optimizer.zero_grad()
            output = model(**batch)
            output.loss.backward()
            optimizer.step()
            if global_step % log_step == 0:
                print(f"ep: {ep}, global_step: {global_step}, loss: {output.loss.item()}")
                global_step += 1
        clf_result = evaluate()
        print(f"ep: {ep}, result: {clf_result}")
```