
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

BOA304

Building a product review classifier with transfer learning

Banjo Obayomi (he/him)

Senior Developer Advocate
AWS

James Yi (he/him)

Senior Partner SA, AI/ML
AWS

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Solution overview

Data preparation

Model training

Model evaluation

Model deployment

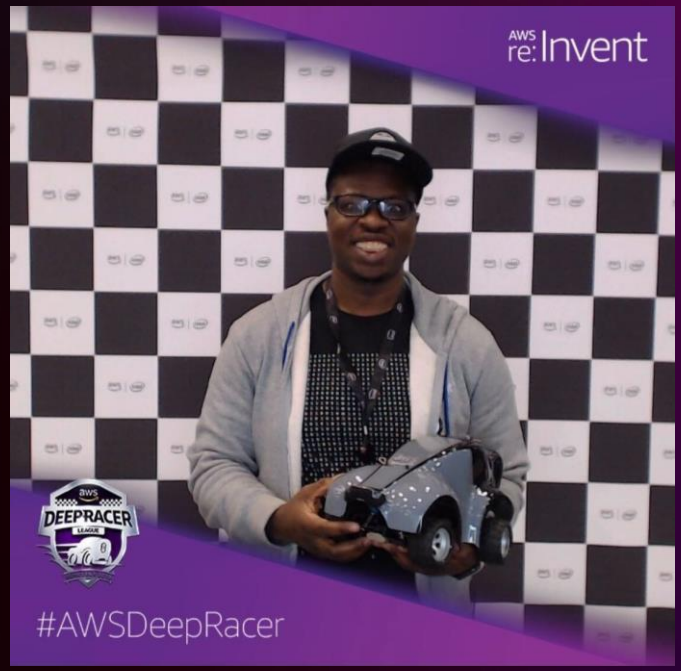
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Solution overview

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Product review classifier





★★★★★ **Easier than I thought.**

Reviewed in the United States on March 25, 2020

Style: DeepRacer | **Verified Purchase**

Pretty awesome that my 6yo son and I - neither of whom can code - could "teach a brain" in the cloud, "put it in the body" of the car and watch it move around on its in less than a week. Does anyone know where I can print different tracks out that could fit in 150 sq foot space? Have B&W printer, scotch tape, green paint, blue paint, garage floor & walls in need of a makeover. :-)

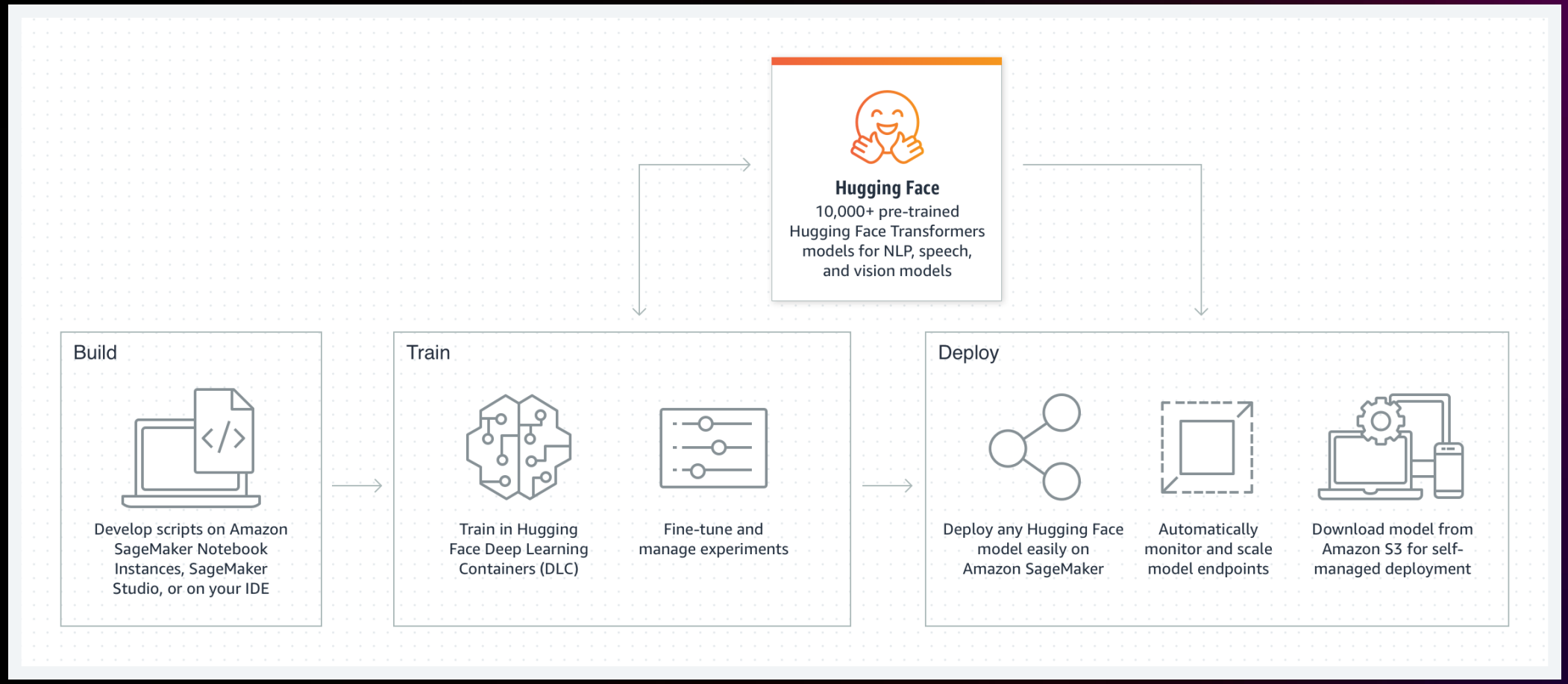
7 people found this helpful

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How do we classify helpful reviews at scale?

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Natural language processing overview



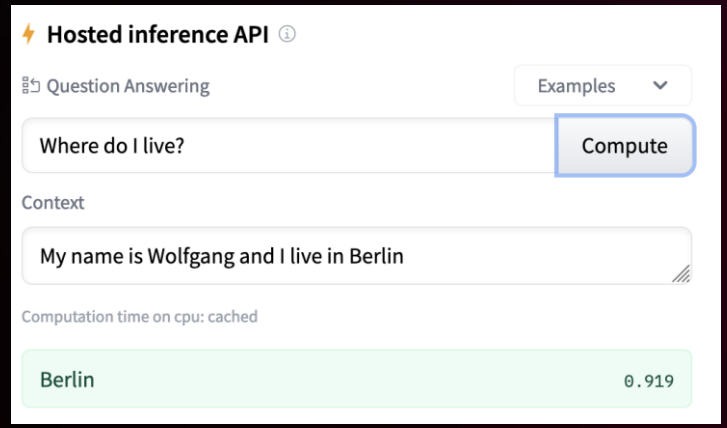
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

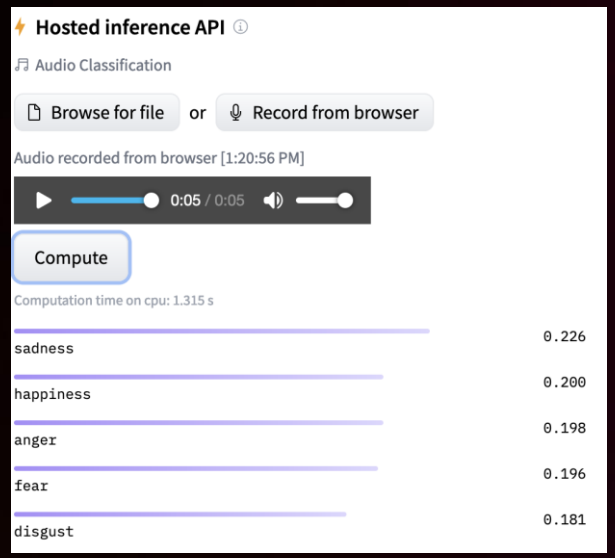
_DALL·E_Mini_2021, author = {Dayma, Boris and Patil, Suraj and Cuenca, Pedro and Saifullah, Khalid and Abraham, Tanishq and Lê Khắc, Phúc and Melas, Luke and Ghosh, Ritobrata

Hugging Face Transformers

STATE-OF-THE-ART MACHINE LEARNING FOR JAX, PYTORCH, AND TENSORFLOW

- Download and train state-of-the-art pretrained models
- Reduce your compute costs
- Reduce your carbon footprint
- Saves time from training using transfer learning





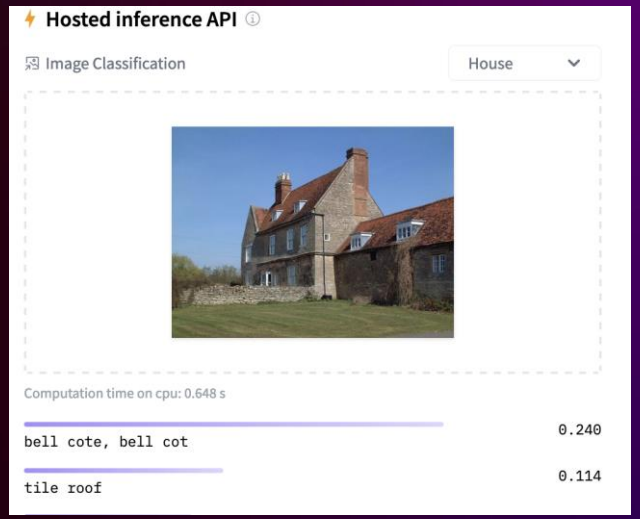




Image: DALL-E Mini, 2021, authors Boris Dayma, Suraj Patil, Pedro Cuenca, Khalid Saifullah, Tanishq Abraham, Phuc Le Khac, Luke Melas, and Ritobrata Ghosh

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Data preparation

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Data Exchange

AWS Data Exchange is a service that makes it easy for customers to find, subscribe to, and use third-party data in the AWS Cloud.

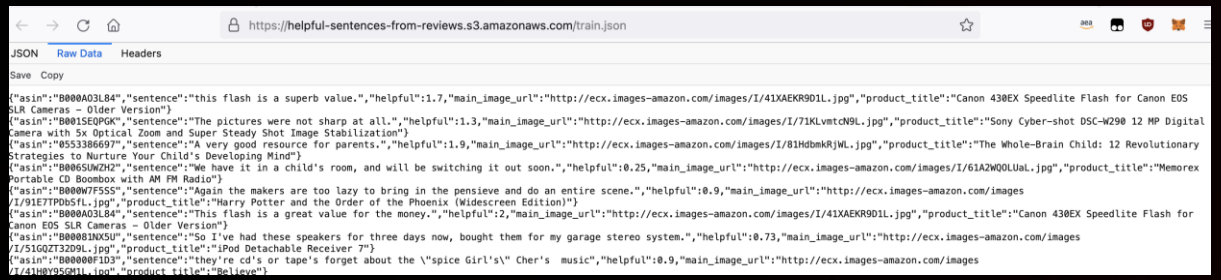
Open Data on AWS Data Exchange

Helpful Sentences from Reviews

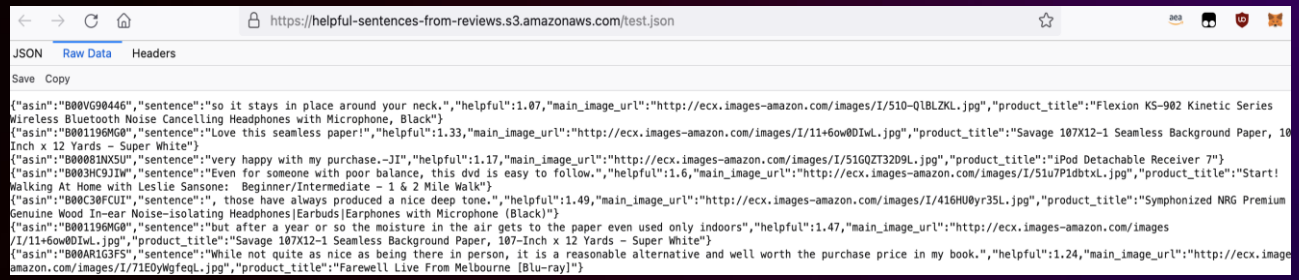
Provided by: Amazon, part of the Open Data

This product is part of the Open Data and contains data sets that are publicly available for anyone to access and use. No subscription is required. Unless specifically stated in the applicable data set documentation, data sets available through the Open Data are not provided and maintained by AWS.

train.json



test.json




```
{"asin":"B000AO3L84","sentence":"this flash is a superb value.","helpful":1.7,"main_image_url":"http://ecx.images-amazon.com/images/I/41XAEKR9D1L.jpg","product_title":"Canon 430EX Speedlite Flash for Canon EOS SLR Cameras - Older Version"}
```


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

```
import json
import requests
import pandas as pd

# Raw data
train_data_raw = "https://helpful-sentences-from-reviews.s3.amazonaws.com/train.json"
test_data_raw = "https://helpful-sentences-from-reviews.s3.amazonaws.com/test.json"

# Download data
train_data = get_data(train_data_raw)
test_data = get_data(test_data_raw)

# Convert into dataframe
train_df = data_to_df(train_data)
test_df = data_to_df(test_data)

# Save dataframe
train_df.to_csv("helpful_sentences_train.csv", index=False)
test_df.to_csv("helpful_sentences_test.csv", index=False)
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

{"asin":"B000AO3L84","sentence":"this flash is a superb value.","helpful":1.7,"main_image_url":"http://ecx.images-amazon.com/images/I/41XAEKR9D1L.jpg","product_title":"Canon 430EX Speedlite Flash for Canon EOS SLR Cameras - Older Version"}
{"asin":"B001SEQPGK","sentence":"The pictures were not sharp at all.","helpful":1.3,"main_image_url":"http://ecx.images-amazon.com/images/I/71KLvmtcN9L.jpg","product_title":"Sony Cyber-shot DSC-W290 12 MP Digital Camera with 5x Optical Zoom and Super Steady Shot Image Stabilization"}
{"asin":"0553386697","sentence":"A very good resource for parents.","helpful":1.9,"main_image_url":"http://ecx.images-amazon.com/images/I/81HdbmkRjWL.jpg","product_title":"The Whole-Brain Child: 12 Revolutionary Strategies to Nurture Your Child's Developing Mind"}

```
def get_data(url: str):
    """
    Purpose:
        gets the json data
    Args:
        url - location of data
    Returns:
        data_array: array of the data
    """

    data = requests.get(url)
    raw_data = data.text
    data_array = raw_data.split("\n")

    return data_array
```

[{"asin":"B000AO3L84","sentence":"this flash is a superb value.","helpful":1.7,"main_image_url":"http://ecx.images-amazon.com/images/I/41XAEKR9D1L.jpg","product_title":"Canon 430EX Speedlite Flash for Canon EOS SLR Cameras - Older Version"},
{"asin":"B001SEQPGK","sentence":"The pictures were not sharp at all.","helpful":1.3,"main_image_url":"http://ecx.images-amazon.com/images/I/71KLvmtcN9L.jpg","product_title":"Sony Cyber-shot DSC-W290 12 MP Digital Camera with 5x Optical Zoom and Super Steady Shot Image Stabilization"},
{"asin":"0553386697","sentence":"A very good resource for parents.","helpful":1.9,"main_image_url":"http://ecx.images-amazon.com/images/I/81HdbmkRjWL.jpg","product_title":"The Whole-Brain Child: 12 Revolutionary Strategies to Nurture Your Child's Developing Mind"}]

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

```python
def data_to_df(data_array):
    """
    Purpose:
        turn data to a df
    Args:
        data_array - array of data
    Returns:
        df: df of the data
    """

    df_map = {}
    df_map["text"] = []
    df_map["label"] = []

    for item in data_array:

        try:
            item_json = json.loads(item)
        except:
            continue

        df_map["text"].append(item_json["sentence"])

        # Get helpful label from score
        helpful_label = get_helpful_label(item_json["helpful"])
        df_map["label"].append(helpful_label)

    # Create dataframe
    df = pd.DataFrame.from_dict(df_map)

    return df
```

```python
def get_helpful_label(helpful_score):
    """
    Purpose:
        gets a label based on score
    Args:
        helpful_score: score of how helpful
    Returns:
        helpful_label: label of how helpful
    """

    if helpful_score < 1:
        return 0
    else:
        return 1
```

```
1  text,label
2  this flash is a superb value.,1
3  The pictures were not sharp at all.,1
4  A very good resource for parents.,1
5  "We have it in a child's room, and will be switching it out soon.",0
6  Again the makers are too lazy to bring in the pensieve and do an entire scene.,0
7  This flash is a great value for the money.,1
8  "So I've had these speakers for three days now, bought them for my garage stereo system.",0
9  "they're cd's or tape's forget about the ""spice Girl's"" Cher's  music",0
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Model training

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Machine learning frameworks



PyTorch, the PyTorch logo and any related marks are trademarks of Facebook, Inc.

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Hugging Face provides a developer-first experience to build, train, and deploy machine learning models

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Model training

```
from transformers import TrainingArguments, Trainer
from transformers import AutoModelForSequenceClassification
from transformers import AutoTokenizer
from transformers import DataCollatorWithPadding
import pandas as pd
from datasets import Dataset
from datasets import load_dataset

# Create tokenizer
tokenizer = AutoTokenizer.from_pretrained("distilbert-base-uncased")
data_collator = DataCollatorWithPadding(tokenizer=tokenizer)

# Load Base Model
model = AutoModelForSequenceClassification.from_pretrained(
    "distilbert-base-uncased", num_labels=2
)
*def* preprocess_function(examples):
    return tokenizer(examples["text"], truncation=True)

# Load Data
test_data = pd.read_csv("helpful_sentences_test.csv")
train_data = pd.read_csv("helpful_sentences_train.csv")

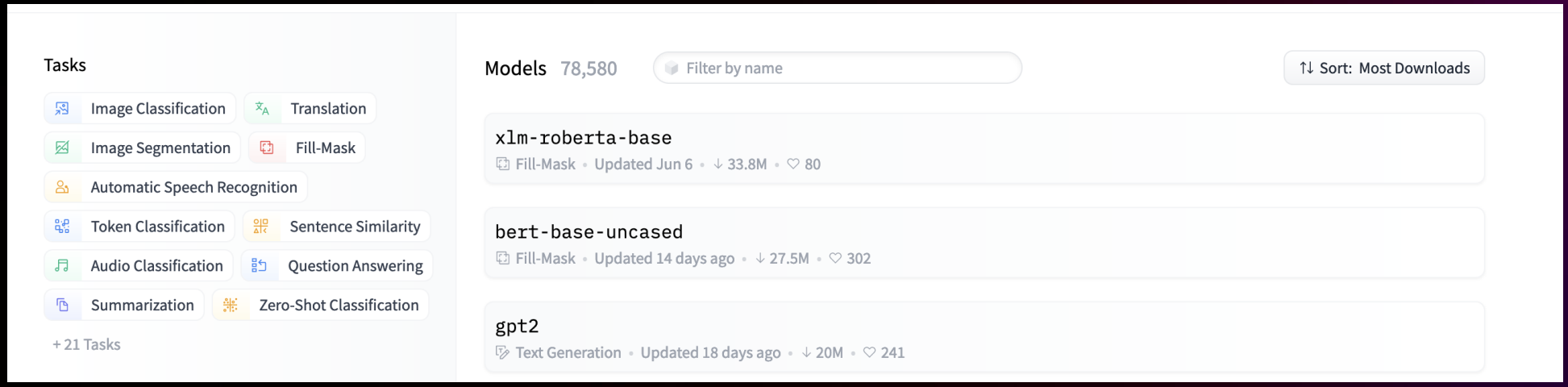
# Create Dataset
test_dataset = Dataset.from_pandas(test_data)
train_dataset = Dataset.from_pandas(train_data)

tokenized_test = test_dataset.map(preprocess_function, batched=True)
tokenized_train = train_dataset.map(preprocess_function, batched=True)
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Retraining a pretrained model on a dataset specific to your task is known as transfer learning



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Model training: Local

```python
from datasets import load_metric
import numpy as np

# Metrics
metric = load_metric("accuracy")
def compute_metrics(eval_pred):
    logits, labels = eval_pred
    predictions = np.argmax(logits, axis=-1)
    return metric.compute(predictions=predictions, references=labels)

# Training args
training_args = TrainingArguments(
    output_dir="./results",
    learning_rate=2e-5,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    num_train_epochs=5,
    weight_decay=0.01,
    evaluation_strategy="epoch"
)

# Set up Training job
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_train,
    eval_dataset=tokenized_test,
    tokenizer=tokenizer,
    data_collator=data_collator,
    compute_metrics=compute_metrics,
)

# Train the model
trainer.train()
```

Local checkpointing

Configurable checkpoints based on training epochs

```
(base) [bobayomi@3c22fb0a93f3]results> ls
checkpoint-1000                 checkpoint-3500                 checkpoint-5500
checkpoint-1500                 checkpoint-4000                 checkpoint-6000
checkpoint-2000                 checkpoint-4500                 distilbert-base-uncased-helpful-amazon
checkpoint-2500                 checkpoint-500
checkpoint-3000                 checkpoint-5000
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Model training: Amazon SageMaker

```
from sagemaker.huggingface import HuggingFace

# hyperparameters, which are passed into the training job
hyperparameters={'epochs': 1,
                 'train_batch_size': 32,
                 'model_name':'distilbert-base-uncased'
                 }

huggingface_estimator = HuggingFace(entry_point='train.py',
                            source_dir='./scripts',
                            instance_type='ml.p3.2xlarge',
                            instance_count=1,
                            role=role,
                            transformers_version='4.12',
                            pytorch_version='1.9',
                            py_version='py38',
                            hyperparameters = hyperparameters)

# starting the train job with our uploaded datasets as input
huggingface_estimator.fit({'train': training_input_path, 'test': test_input_path})
```



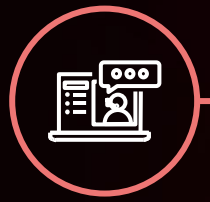
Experiment management and model tuning

Save weeks of effort by automatically tracking training runs and tuning hyperparameters



Debug and profile training runs

Use real-time metrics to correct performance problems



Distributed training

Complete distributed training up to 40% faster



Training compiler

Accelerate training times by up to 50% through more efficient use of GPUs



Managed spot training

Reduce the costs of training by up to 90%

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Model parallelism

Automated and efficient model partitioning

Minimal code change

Optimized for training large models

Distributed training on SageMaker

https://aws.amazon.com/sagemaker/distributed-training/

Data parallelism

Reduced training time

Optimized for AWS network

Support for popular ML framework APIs

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Model evaluation

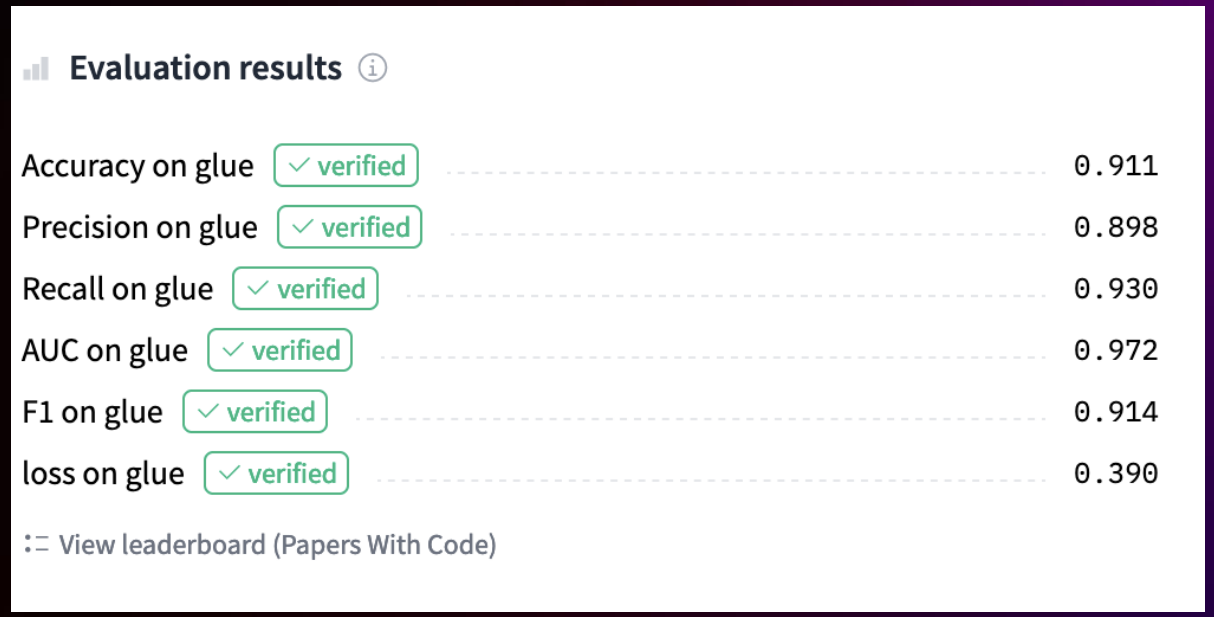
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Hugging Face Evaluate



MAKES EVALUATING AND COMPARING MODELS AND REPORTING THEIR PERFORMANCE EASIER AND MORE STANDARDIZED

- Implementations of dozens of popular metrics
- Comparisons and measurements
- Community metrics



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Model evaluation

```python
from transformers import (
    pipeline,
)
import pandas as pd
from datasets import Dataset
from evaluate import evaluator
import evaluate

# Define model
pipe = pipeline(
    "text-classification", model="results/checkpoint-6000/"
)
# Define dataset
test_data = pd.read_csv("helpful_sentences_test.csv")
test_dataset = Dataset.from_pandas(test_data)

# Define evaluator
accuracy = evaluate.load("accuracy")
```

```python
# Evaluate accuracy
eval = evaluator("text-classification")
result = eval.compute(
    model_or_pipeline=pipe,
    data=test_dataset,
    metric=accuracy,
    label_mapping={"LABEL_0": 0, "LABEL_1": 1},
    strategy="bootstrap",
    n_resamples=200,
)
print(result)
# {'accuracy': {'confidence_interval': (0.7991824857067859, 0.837), 'standard_error': 0.009116387310110274, 'score':
0.823}}
```

```python
# Evaluate F1 score
f1_metric = evaluate.load("f1")
result = eval.compute(
    model_or_pipeline=pipe,
    data=test_dataset,
    metric=f1_metric,
    label_mapping={"LABEL_0": 0, "LABEL_1": 1},
    strategy="bootstrap",
    n_resamples=200,
)
print(result)
#{'f1': {'confidence_interval': (0.8619904693250219, 0.8852541552022207), 'standard_error': 0.006116815390939261,
'score': 0.8729361091170136}}
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Model evaluation

Debug and profile training runs

Use real-time metrics to correct performance problems

In [13]:

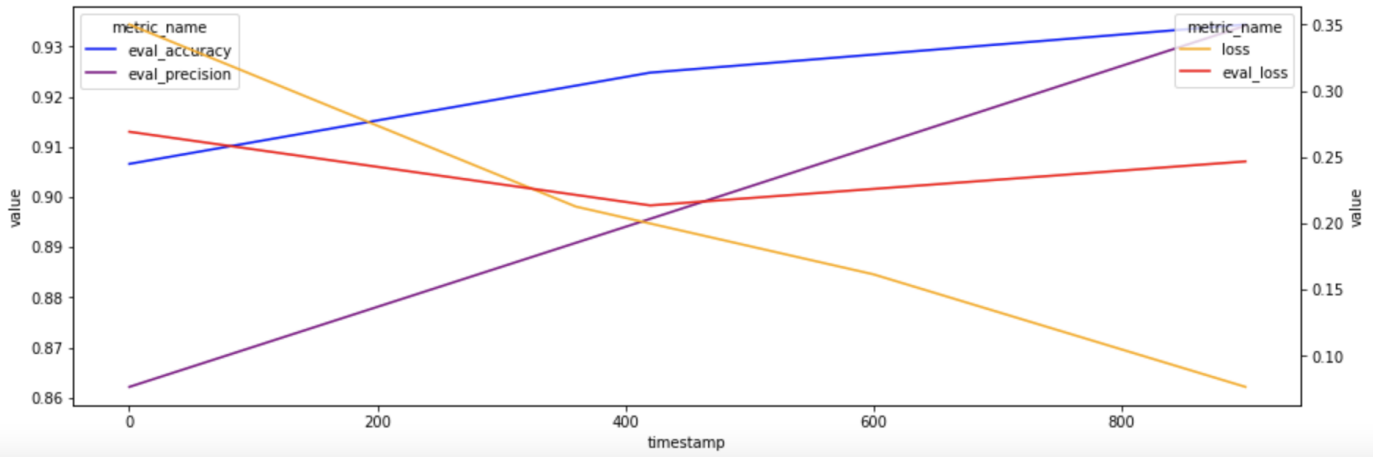
```
evals = df[df.metric_name.isin(['eval_accuracy','eval_precision'])]
losses = df[df.metric_name.isin(['loss', 'eval_loss'])]

sns.lineplot(
    x='timestamp',
    y='value',
    data=evals,
    hue='metric_name',
    palette=['blue', 'purple'])

ax2 = plt.twinx()
sns.lineplot(
    x='timestamp',
    y='value',
    data=losses,
    hue='metric_name',
    palette=['orange', 'red'],
    ax=ax2)
```

Out[13]:

```
<AxesSubplot:label='8b8c5c4f-4666-47d2-aaa5-520ae383cfc4', xlabel='timestamp', ylabel='value'>
```



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Model deployment

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Test locally

```
from transformers import (
    AutoTokenizer,
    AutoModelForSequenceClassification,
    TextClassificationPipeline,
)

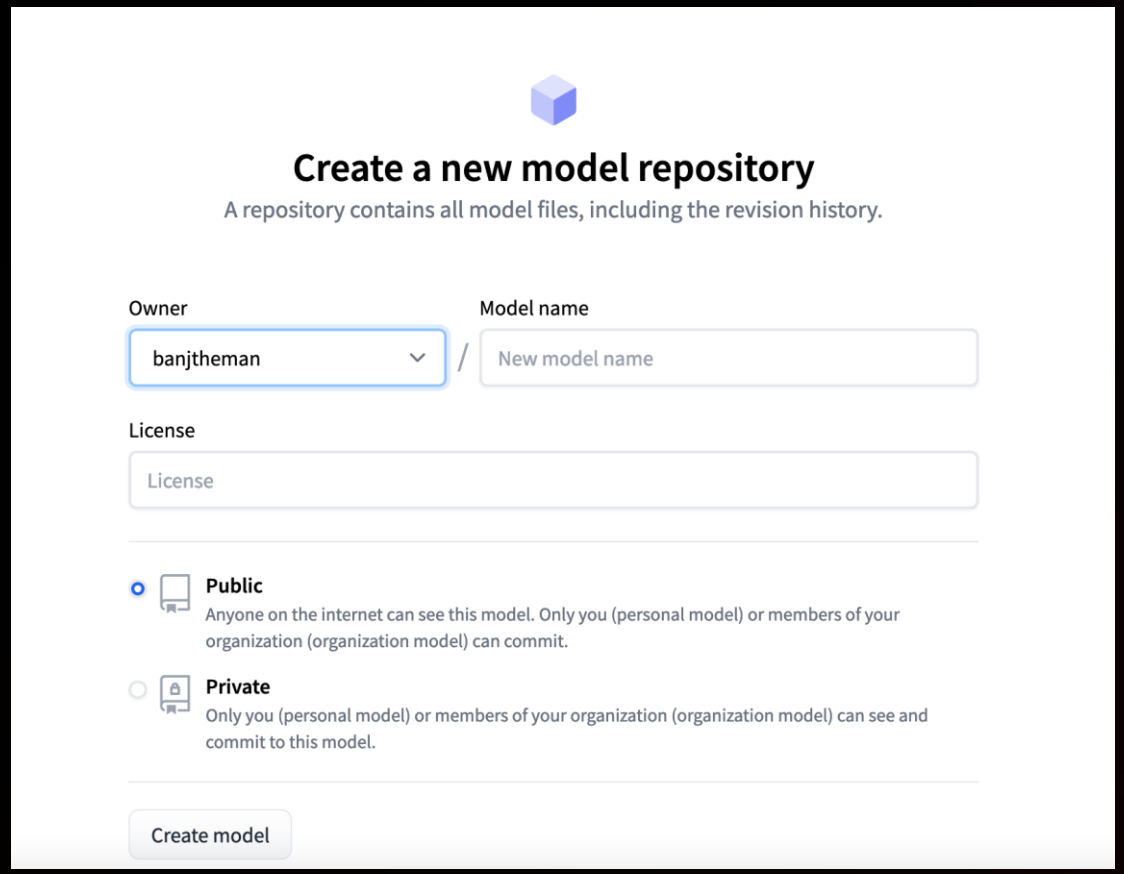
tokenizer = AutoTokenizer.from_pretrained("results/checkpoint-6000/")
model = AutoModelForSequenceClassification.from_pretrained(
    "results/checkpoint-6000/"
)
pipe = TextClassificationPipeline(model=model, tokenizer=tokenizer)

result = pipe("This was a Christmas gift for my grandson.")
print(result)
#[{'label': 'LABEL_0', 'score': 0.998775064945221}]# This is NOT A HELPFUL comment
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

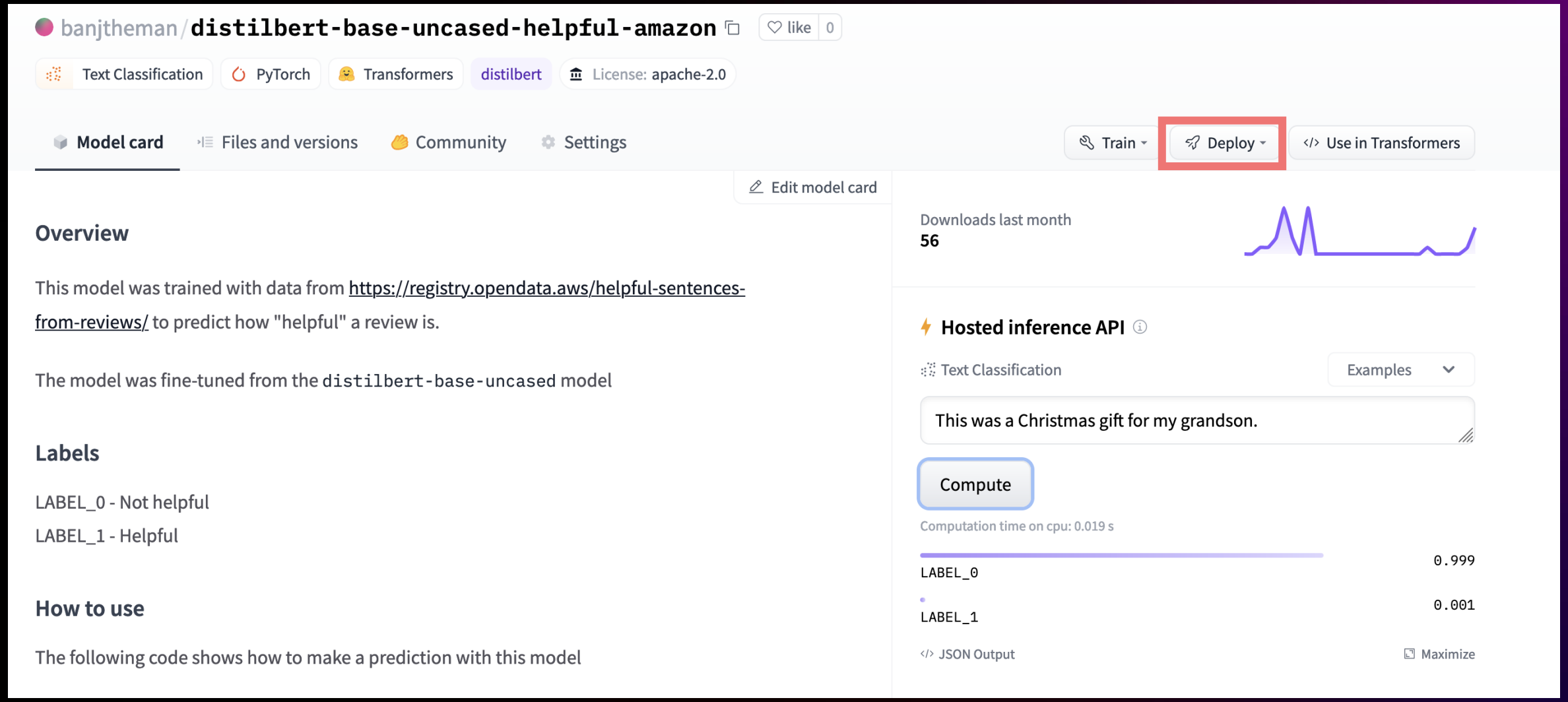
Hugging Face Hub




```
git lfs install
git clone https://huggingface.co/banjtheman/distilbert-base-uncased-helpful-amazon
cp -r PATH_TO_MODEL/results/checkpoint-6000/* distilbert-base-uncased-helpful-amazon/
cd distilbert-base-uncased-helpful-amazon/
git add .
git commit -am "initial commit"
git push
```

banjtheman	Update README.md	53364fd	
.gitattributes	1.18 kB		initial commit
README.md	1.01 kB		Update README.md
config.json	615 Bytes		initial commit
optimizer.pt	LFS	536 MB	initial commit
pytorch_model.bin	LFS	268 MB	initial commit
rng_state.pth	LFS	13.5 kB	initial commit
scheduler.pt	LFS	623 Bytes	initial commit
special_tokens_map.json	112 Bytes		initial commit
tokenizer.json	711 kB		initial commit
tokenizer_config.json	333 Bytes		initial commit

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How to deploy this model using Amazon SageMaker

Task

Select the task you want to use this model for

Text Classification

Configuration

Deploy from a local machine or in AWS?

AWS

Copy

```
from sagemaker.huggingface import HuggingFaceModel
import sagemaker

role = sagemaker.get_execution_role()
# Hub Model configuration. https://huggingface.co/models
hub = {
    'HF_MODEL_ID':'banjtheman/distilbert-base-uncased-helpful-amazon',
    'HF_TASK':'text-classification'
}

# create Hugging Face Model Class
huggingface_model = HuggingFaceModel(
    transformers_version='4.17.0',
    pytorch_version='1.10.2',
    py_version='py38',
    env=hub,
    role=role,
)

# deploy model to SageMaker Inference
predictor = huggingface_model.deploy(
    initial_instance_count=1, # number of instances
    instance_type='ml.m5.xlarge' # ec2 instance type
)
```

Quick Links

- Transformers: Run training on Amazon SageMaker
- Examples

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

```python
import sagemaker
from sagemaker.huggingface.model import HuggingFaceModel
from sagemaker.serverless import ServerlessInferenceConfig

# Specify Model Image_uri
image_uri='763104351884.dkr.ecr.us-east-1.amazonaws.com/huggingface-pytorch-inference:1.9.1-transformers4.12.3-cpu-py38-ubuntu20.04'

# Hub Model configuration. <https://huggingface.co/models>
hub = {
    'HF_MODEL_ID':'banjtheman/distilbert-base-uncased-helpful-amazon',
    'HF_TASK':'text-classification'
}

# create Hugging Face Model Class
huggingface_model = HuggingFaceModel(
   env=hub,                      # configuration for loading model from Hub
   role=sagemaker.get_execution_role(),
                                 # iam role with permissions
   transformers_version="4.17",  # transformers version used
   pytorch_version="1.10",        # pytorch version used
   py_version='py38',            # python version used
   image_uri=image_uri,          # image uri
)

# Specify MemorySizeInMB and MaxConcurrency
serverless_config = ServerlessInferenceConfig(
    memory_size_in_mb=4096, max_concurrency=10,
)

# deploy the serverless endpoint
predictor = huggingface_model.deploy(
    serverless_inference_config=serverless_config
)
```

```python
data = {
  "inputs": "I got this at reinvent",
}

res = predictor.predict(data=data)
print(res)
#[{'label': 'LABEL_0', 'score': 0.9419971704483032}]
```

```python
import boto3
import json


data = {
  "inputs": "I got this at reinvent",
}

# Encode data
user_encode_data = json.dumps(data, indent=2).encode('utf-8')

client = boto3.client('sagemaker-runtime')
endpoint_name = predictor.endpoint_name

# Invoke endpoint
response = client.invoke_endpoint(
    EndpointName=endpoint_name,
    ContentType="application/json",
    Body=user_encode_data)
result = json.loads(response['Body'].read().decode())
print(result)
# [{'label': 'LABEL_0', 'score': 0.9419971704483032}]
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SageMaker inference options

NEW

Real-time inference	Batch transform	Asynchronous inference	Serverless Inference
Low latency	Process large datasets	Near real time	Automatically scale in and out depending on traffic
Ultra-high throughput	Job-based system	Large payloads (1 GB)	Can tolerate cold starts
Multi-model endpoints		Long timeouts (15 minutes)	Pay only for what you use
A/B testing			

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Recap

- Extract, transform, and load datasets from AWS Data Exchange
- Train a Hugging Face model
- Evaluate the model
- Upload the model to Hugging Face Hub
- Create a SageMaker endpoint for the model

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Resources

- https://huggingface.co/banjtheman/distilbert-base-uncased-helpful-amazon
- https://github.com/huggingface/notebooks/tree/main/sagemaker
- https://huggingface.co/docs/sagemaker/main
- https://docs.aws.amazon.com/sagemaker/latest/dg/hugging-face.html

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Banjo Obayomi
@banjtheman

James Yi
@James33179713



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.