

Amazon SageMaker JumpStart Workshop

2021/02/13

シニアエバンジェリスト

亀田治伸

1. Amazon SageMaker JumpStart の起動

1.1 SageMaker のマネージメントコンソールにアクセスし、左のペインから

[SageMaker Studio]を選択します。



1.2 [実行ロール]から[新しいロールの作成]を選びます。

実行ロール

SageMaker Studio には、Amazon SageMaker や Amazon S3 などの他の AWS サービスにアクセスするためのアクセス許可が必要です。実行ロールには、[AmazonSageMakerFullAccess policy](#) がアタッチされています。このポリシーがアタッチされたロールがない場合は、AWS で作成できます。

新しいロールの作成
新しいロールの作成
カスタム IAM ロールの ARN の入力
既存のロールの使用
AmazonSageMaker-ExecutionRole-20180703T173852
AmazonSageMaker-ExecutionRole-20180831T201100
AmazonSageMaker-ExecutionRole-20210108T152479
AmazonSageMakerServiceCatalogProductsUseRole

は、このオプションを選択する必要があります。

1.3 全てデフォルトのまま[ロールの作成]を押します。

IAM ロールを作成する

×

IAM ロールを渡すと、ユーザーに代わって他の AWS のサービスでアクションを実行するアクセス許可が Amazon SageMaker に与えられます。ここでロールを作成すると、[AmazonSageMakerFullAccess](#) 作成する IAM ポリシーで記述されたアクセス許可が付与されます。

作成する IAM ロールにより、以下へのアクセスが提供されます。

☒ 指定する S3 バケット - オプション

☒ 任意の S3 バケット
自分のノートブックインスタンスにアクセスできるユーザーが、アカウント内の任意のバケットとそのコンテンツにアクセスすることを許可します。

☐ 特定の S3 バケット

例: `bucket-name-1`、`bucket-name-2`、`...`

カンマ区切り。ARN、「*」、および「/」はサポートされません。

☐ なし

☒ 名前に「sagemaker」が含まれる任意の S3 バケット

☒ 名前に「sagemaker」が含まれる任意の S3 オブジェクト

☒ タグ「sagemaker」と値「true」が含まれる任意の S3 オブジェクト

[オブジェクトのタグ付けの表示](#)

☒ SageMaker へのアクセスを許可するバケットポリシーを持つ S3 バケット

[S3 バケットポリシーの表示](#)

キャンセル

ロールの作成

1.4 以下で IAM ロールが正しく作成されたことを確認します。その後[送信]を押します。

実行ロール

SageMaker Studio には、Amazon SageMaker や Amazon S3 などの他の AWS サービスにアクセスするためのアクセス許可が必要です。実行ロールには、[AmazonSageMakerFullAccess policy](#) がアタッチされています。このポリシーがアタッチされたロールがない場合は、AWS で作成できます。

AmazonSageMaker-ExecutionRole-20210208T153561

☒ 成功! IAM ロールを作成しました。
[AmazonSageMaker-ExecutionRole-20210208T153561](#)

×

1.5 以下のダイアログが出ますので、VPC を選びます。(VPC が 1 個の場合、ダイアログが出ない場合がありますが、その際はそのまま進みます)

VPC を選択

×

複数の VPC があります。SageMaker Studio で使用する VPC を選択します。

☐ HandsOn
 ☐ vpc-071ef70c7b103769f

キャンセル

保存して続行

1.6 以下のドロップダウンから Subnet を指定します。 (**Private で OK?**)

VPC を選択

×

複数の VPC があります。SageMaker Studio で使用する VPC を選択します。

subnet-2a2d7d73

subnet-0c78c5f011427e6d9

subnet-2a2d7d73

×

キャンセル

保存して続行

1.7 起動が開始しますので、しばらく待ちます。

SageMaker Studio を準備する

×

Studio が必要とするクイックスタートユーザー [default-1612765795871] とリソースを設定しています。これは 1 回限りの設定であり、数分かかる場合があります。

Amazon SageMaker > SageMaker Studio > コントロールパネル

SageMaker Studio コントロールパネル

ユーザー名を選択し、[Studio を開く] を選択して開始する

ユーザーを追加

< 1 >

⚙

ユーザー名	最終更新日時	作成済み
ユーザーなし		

ユーザーを追加するには、[Add user] を選択し、ユーザー名を入力します。

▼ Studio のサマリー

Studio を削除する方法

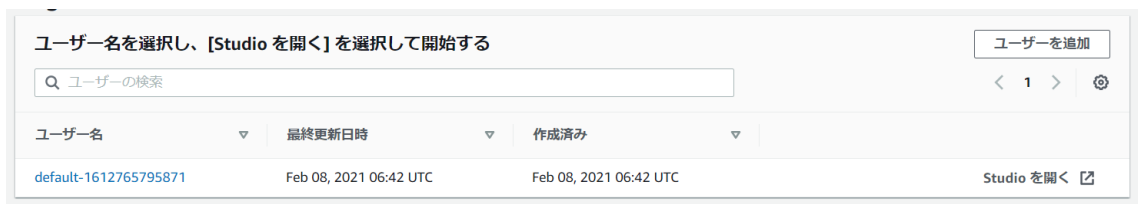
Studio を削除

設定を編集

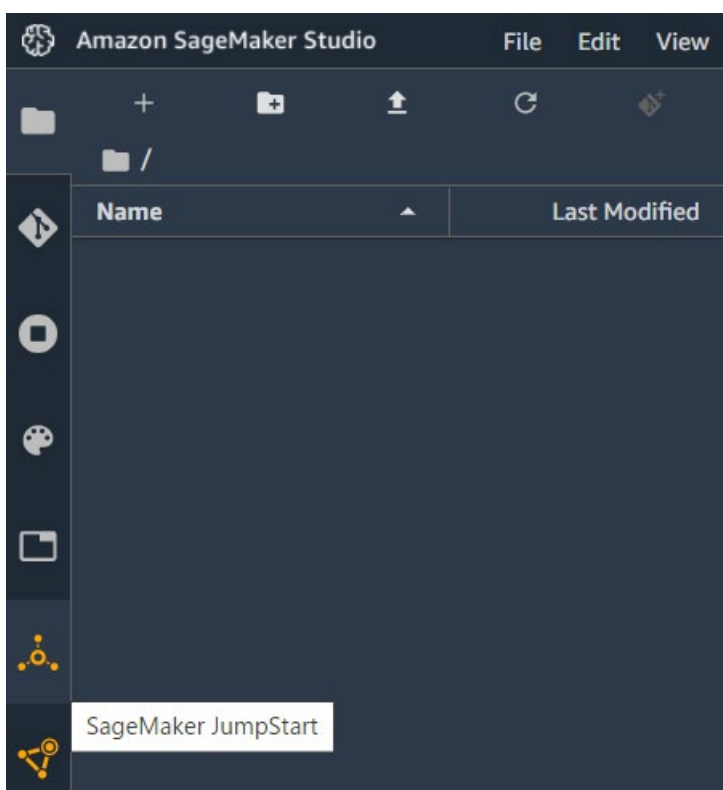
<div>ステータス</div> <div>⌚ Pending</div>	<div>Studio ID</div> <div>d-ptzyvedo9jz5</div>	<div>実行ロール</div> <div>arn:aws:iam::294963776963:role/service-role/AmazonSageMaker-ExecutionRole-20210208T153561</div>	<div>認証方法</div> <div>AWS Identity and Access Management (IAM)</div>
---------------------------------------	--	---	---

1.8 構築が完了したら[Studio を開く]を押します。起動に少し時間がかかりますの

で、数分間待ちます。

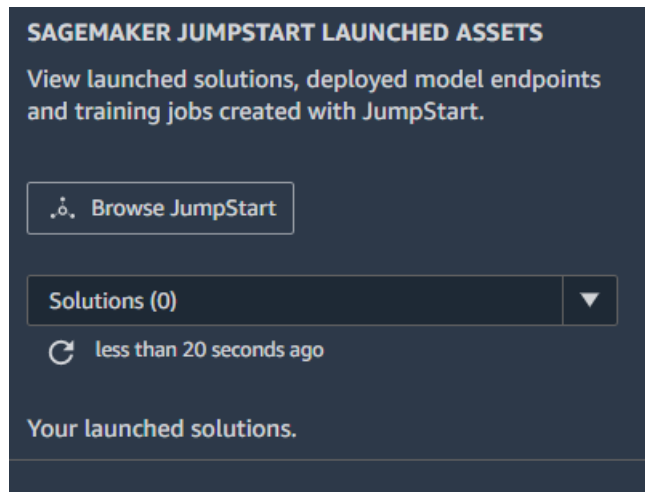


1.9 Jupyter Lab が起動したら、左のペインから JumpStart のボタンを押します。



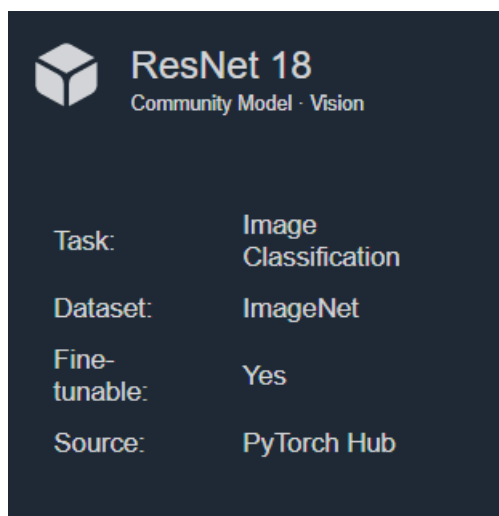
1.10 [Browse JumpStart]のボタンを押します。右側の画面にすぐ利用できるソリ

ューションやモデル群がそろっています。



2 画像解析モデル(Vision models)の起動

2.1 [ResNet 18]をクリックします。



2.2 デフォルトのまま[Deploy]ボタンを押します。

MODEL

ResNet 18

Vision · Image Classification

[Browse JumpStart](#)

Get Started

Deploy Model

Deploy a pretrained model to an endpoint for inference. Deploying on SageMaker hosts the model on the specified compute instance and creates an internal API endpoint. JumpStart will provide you an example notebook to access the model after it is deployed. [Learn more.](#)

▼ Deployment Configuration

Customize the machine type and endpoint name. [Learn more.](#)

Machine Type
ml.m5.large ▼

Endpoint Name
pt-ic-resnet18

[Reset to default](#)

[Deploy](#)

2.3 学習済モデルの起動中となります。

MODEL

jumpstart-pt-ic-resnet18

Preparing your Model

We are preparing the model to attach to your endpoint. This may take a few minutes.


Endpoint details

Model	not available
Endpoint arn	not available
Endpoint name	jumpstart-pt-ic-resnet18
Instance	ml.m5.large
Number of instances	1

エラーが出る場合、AWS アカウントの開設直後など、起動しようとしているインスタンス[ml.m5.large]の利用が許可されていないことが考えられます。AWS サポートへ上限緩和申請を行う必要があります。(<https://aws.amazon.com/jp/aws-jp-faq/>)

2.4 起動が完了すると[In Service]と表示されます。

Endpoint Status

In Service  less than a minute ago
This endpoint is operational and ready to respond to inference requests.

Model	ResNet 18
Endpoint arn	arn:aws:sagemaker:ap-northeast-1:294963776963:endpoint/jumpstart-pt-ic-resnet18
Endpoint name	jumpstart-pt-ic-resnet18
Instance	mLm5.large
Number of instances	1

2.5 [Open Notebook]を押して推論環境へアクセスを行うテスト環境を起動します。

Use Endpoint from Studio

Open a demo notebook to query your endpoint from Studio.

[Open Notebook](#)

注意点：以下の作業を行う前に、先にブラウザの別タブで 2.13 の作業を先にお願ひします。後ほど転移学習の使う花の画像セットを S3 バケットへアップロードするのですが、画像枚数が多く、数十分かかりますので、先に作業を行ってください。アップロード中に以下の作業を続けてください。

2.6 ソースコード部分を選び、1 個つつ上の▶ボタンを押して実行します。(必ず前の部分の実行が終わってから次を実行します。)

Unknown Python 3 (Data Science) Kernel: CPU: -% MEM: -% Share

The class_id_to_label_mapping file has been downloaded from TensorFlow. Apache 2.0 License

```
[ ]: import boto3
from IPython.core.display import HTML

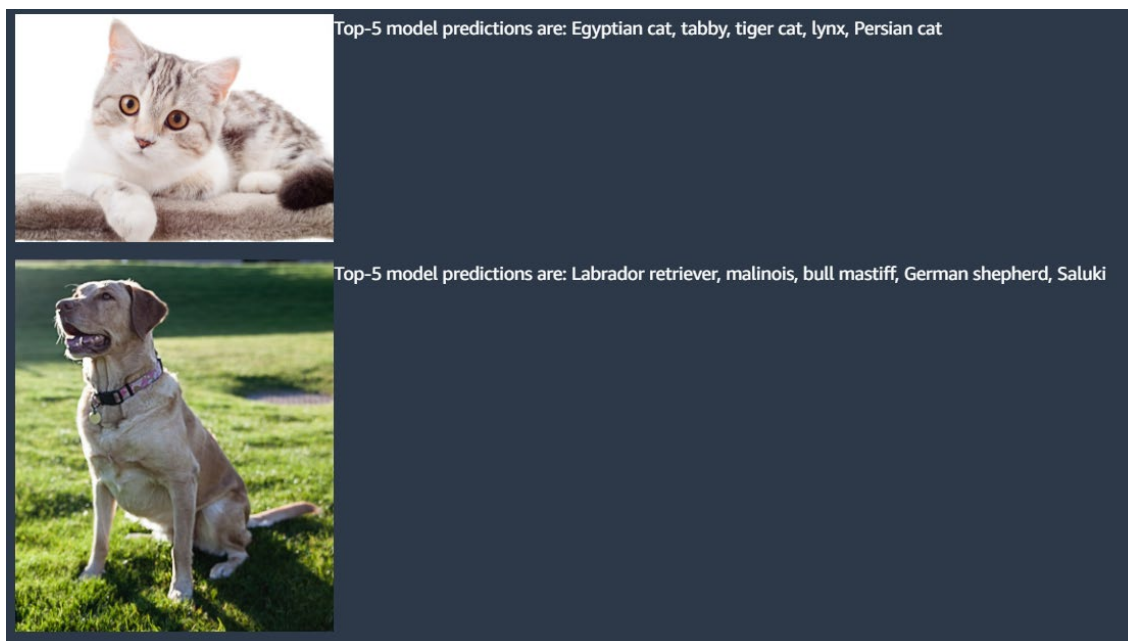
region = boto3.Session().region_name
s3_bucket = f"jumpstart-cache-prod-{region}"
key_prefix = "inference-notebook-assets"
s3 = boto3.client("s3")

def download_from_s3(key_filenames):
    for key_filename in key_filenames:
        s3.download_file(s3_bucket, f"{key_prefix}/{key_filename}", key_filename)

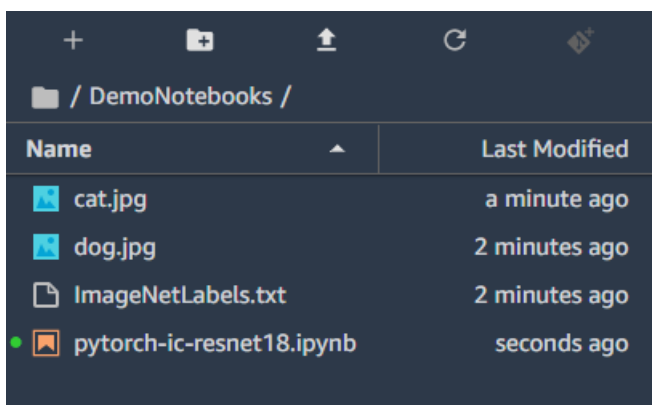
cat_jpg, dog_jpg, ImageNetLabels = "cat.jpg", "dog.jpg", "ImageNetLabels.txt"
download_from_s3(key_filenames=[cat_jpg, dog_jpg, ImageNetLabels])

HTML('<table><tr><td>  <figcaption>cat.jpg</figcaption>'
    '</td><td>  <figcaption>dog.jpg</figcaption>'
    '</td></tr></table>')
```

2.7 犬と猫が正しく推論できていることがわかります。



2.8 適当な鳥の画像をダウンロードします。その画像を左のペインからアップロードしてください。(bird.jpg)としてください。



2.9 [Open the downloaded images and load in memory. You can upload any image from your local computer in the directory and open them here.]の部分を書き換え実行します。

```
images = {}
```

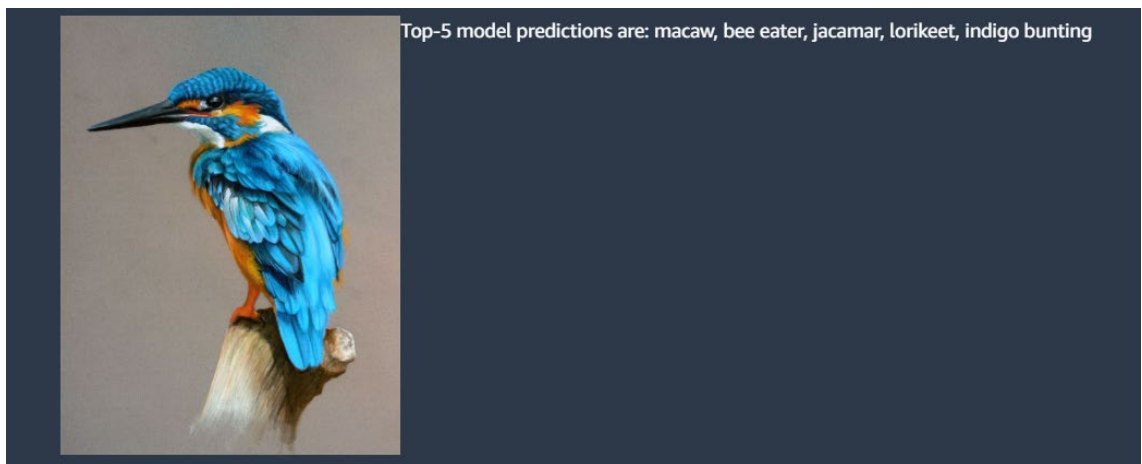


```
with open("bird.jpg", 'rb') as file: images["bird.jpg"] = file.read()

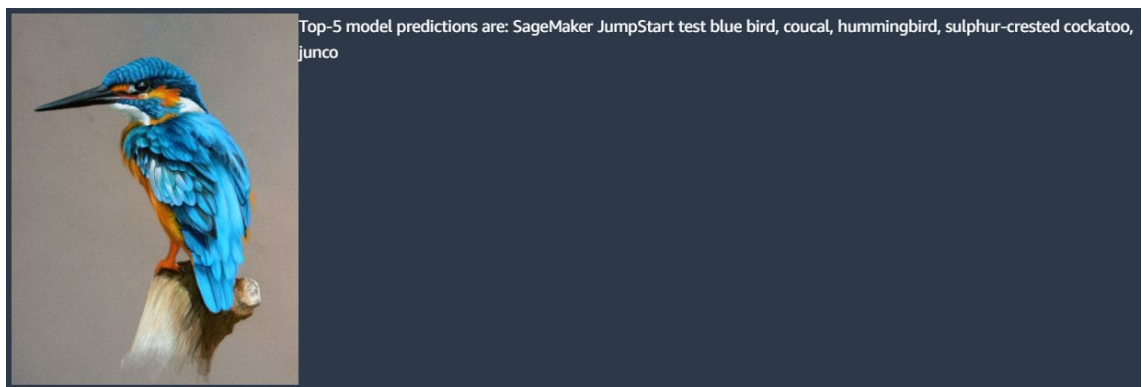
with open("ImageNetLabels.txt", 'r') as file: class_id_to_label =

file.read().splitlines()
```

2.10 以下のように正しく推論が実行されます。



2.11 [ImageNetLabels.txt]を開き、鳥の推論時に出てきた文字列を検索し適当に書き換えて、[Open the downloaded...]の下の部分から再度実行します。パラメータが書き換わったことが確認できます。



2.12 今度は再度 JumpStart のトップ画面から ResNet 50 の起動画面にいきます。

先ほど起動した[Deploy Mode]の下に[Fine-tune Model]のボタンがあります。こちらで、転移学習を行い追加の判断を行う画像を学習させることができます。

2.13 [https://filessharingharunobukameda.s3-ap-northeast-](https://filessharingharunobukameda.s3-ap-northeast-1.amazonaws.com/flower_photos.zip)

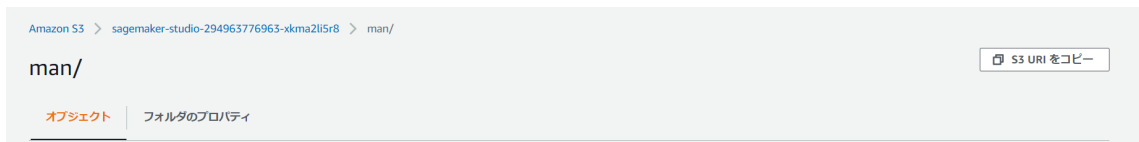
[1.amazonaws.com/flower_photos.zip](https://filessharingharunobukameda.s3-ap-northeast-1.amazonaws.com/flower_photos.zip) から

ダウンロードした[flower_photos]をフォルダ毎 S3 バケットにアップロードします。今までの作業の中で S3 バケットができていますので、[sagemaker-studio-xxxxxxx]となっているバケットを選んでフォルダをアップロードしてください。[flower_photos]の下に書く花の名前のフォルダがあり、その中に複数の jpg ファイルが含まれています。

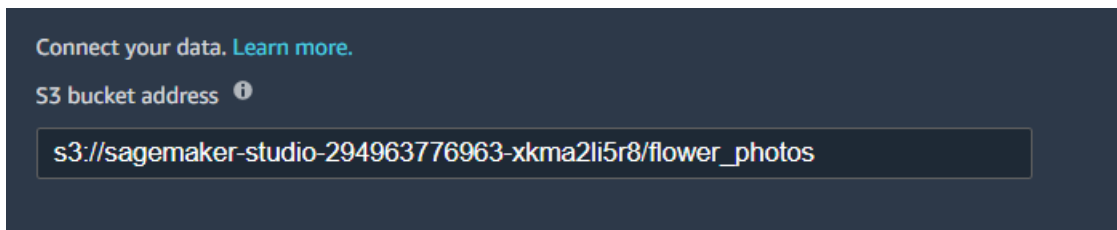
ファイルとフォルダ (3681 合計, 222.6 MB)					
このテーブル内のすべてのファイルとフォルダがアップロードされます。					
<div><div>削除</div><div>ファイルを追加</div><div>フォルダの追加</div></div>					
<div>🔍 名前で検索</div> <div>< 1 2 3 4 5 6 7 ... 369 ></div>					
<input type="checkbox"/>	名前 ▲	フォルダ ▼	タイプ ▼	サイズ ▼	
<input type="checkbox"/>	100080576_f52e8ee070_n.jpg	flower_photos/daisy/	image/jpeg	26.2 KB	
<input type="checkbox"/>	10043234166_e6dd915111_n.jpg	flower_photos/dandelion/	image/jpeg	26.1 KB	
<input type="checkbox"/>	1008566138_6927679c8a.jpg	flower_photos/sunflowers /	image/jpeg	139.9 KB	
<input type="checkbox"/>	10090824183_d02c613f10_m.jpg	flower_photos/roses/	image/jpeg	19.7 KB	
<input type="checkbox"/>	100930342_92e8746431_n.jpg	flower_photos/tulips/	image/jpeg	25.6 KB	

(アップロード前のフォルダ指定画面)

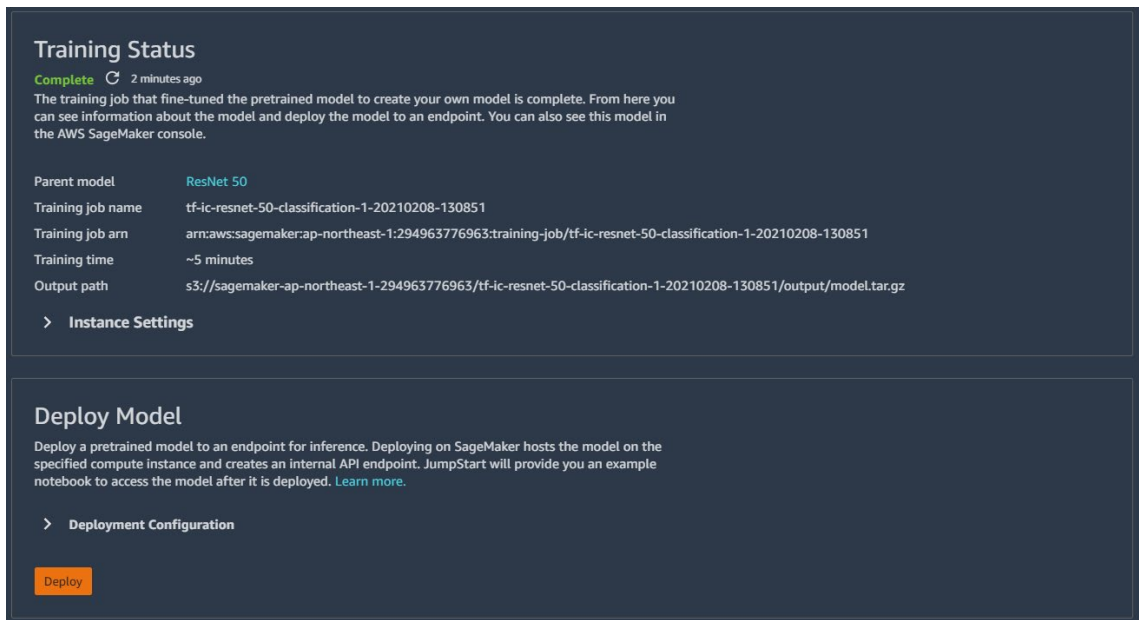
2.14 [flower_photos]のフォルダをクリックして、[S3 URI のコピー]を押してください。



2.15 以下のようにコピーした URI を入力します。一番最後の[/]は削除してください。[Train]ボタンを押すと学習が開始します。




2.16 学習が完了すると[Deploy]ボタンが出てきますので、それをクリックします。



2.17 Endpoint の作成が完了すると[Open Notebook]のボタンが表示されますのでそちらをクリックします。

Endpoint Status

In Service  less than 20 seconds ago
This endpoint is operational and ready to respond to inference requests.

Model	ResNet 50
Endpoint arn	arn:aws:sagemaker:ap-northeast-1:294963776963:endpoint/jumpstart-tf-ic-resnet-50-classification-1
Endpoint name	jumpstart-tf-ic-resnet-50-classification-1
Instance	ml.m5.large
Number of instances	1

Use Endpoint from Studio


















Open a demo notebook to query your endpoint from Studio.

[Open Notebook](#)

2.18 Notebook が起動したら先ほどと同じように順番にクリックして犬と猫が推論さ

れることを確認します。

2.19 推論写真フォルダの jpg ファイルを 1 枚ずつアップロードします。

+    	
/ DemoNotebooks /	
Name	Last Modified
 bird.jpg	a day ago
 cat.jpg	15 minutes ago
 daisy.jpg	22 minutes ago
 dandelion.jpg	21 minutes ago
 dog.jpg	15 minutes ago
 ImageNetLabels.txt	15 minutes ago
 kameda.jpg	16 hours ago
 label.txt	13 minutes ago
 pytorch-ic-resnet18_1.ipyn...	an hour ago
 roses.jpg	21 minutes ago
 sunflowers.jpg	21 minutes ago
 tensorflow-ic-resnet-50-c...	11 minutes ago
 tulips.jpg	20 minutes ago

2.20 [Open the downloaded images...]の下プログラムを以下に書き換えます。

```
images = {}

with open("kameda.jpg", 'rb') as file: images["kameda.jpg"] = file.read()

with open("daisy.jpg", 'rb') as file: images["daisy.jpg"] = file.read()

with open("dandelion.jpg", 'rb') as file: images["dandelion.jpg"] = file.read()

with open("roses.jpg", 'rb') as file: images["roses.jpg"] = file.read()

with open("cat.jpg", 'rb') as file: images["cat.jpg"] = file.read()

with open("sunflowers.jpg", 'rb') as file: images["sunflowers.jpg"] =

file.read()

with open("tulips.jpg", 'rb') as file: images["tulips.jpg"] = file.read()

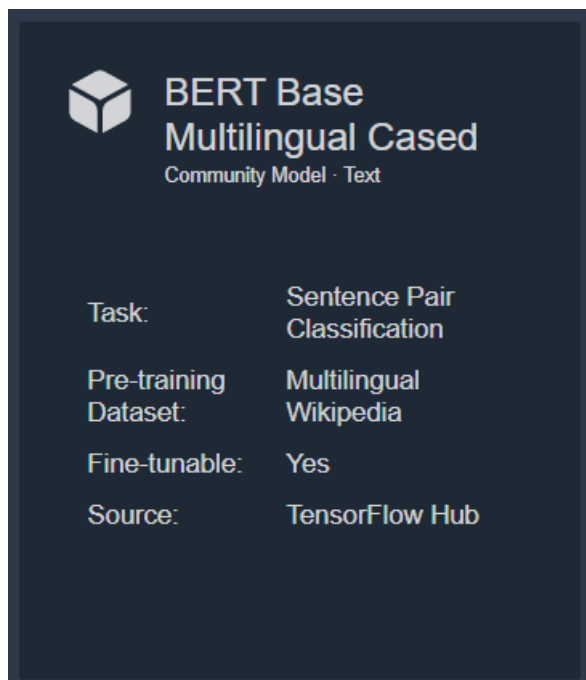
with open("ImageNetLabels.txt", 'r') as file: class_id_to_label =

file.read().splitlines()
```

2.21 推論を実行して結果を見ます。この状態ではラベル用テキストが正しく設定されていないため、花の違いを正しく区別しているものの、表示が正しくされていません。[label.txt]をアップロードし上記プログラムの[ImageNetLabels.txt]を書き換えます。再度推論を実行すると花の違いやラベル出力が異なることがわかります。

3. 自然言語解析モデルの起動

3.1. BERT Base Multilingual Cased を選びます。



3.2 先ほどと同じ手順でモデルを起動し、notebook を開きます。その際[Endpoint Name]を適当な短い名前に変更します。(testyyyyymmdd 等) BERT は一般的にモデルが巨大であるため、先ほどよりは時間がかかります。

3.3 起動したら、上から 1 個ずつ実行し、QA が動作することを確認します。

3.4 2 つ目のスクリプトの[sentence_pair1]の 2 つの文字列を関連する日本語文字列に置き換えます。以下はその一例です。

```
sentence_pair1 = ["日本の首都は?", "日本には多くの街がありますが、首都は東京です"]
```

サイド上から順番に実行し、2 つの文に相関があると推論される場合、

[entailment]、相関が無いと推論される場合[no-entailment]と出力されます。

4. お疲れ様でした。

以下を必ず削除してください。

- EFS
- S3 バケット (sagemaker から始まるもの)
- SageMaker 推論エンドポイント
- SageMaker モデル
- SageMaker Studio のアプリ (Studio からユーザーをクリックで出てきます)
- SageMaker Studio のユーザー
- SageMaker Studio