

Tutoriel AWS SageMaker

Thomas - Pereg - Paul - Guillaume - Julien

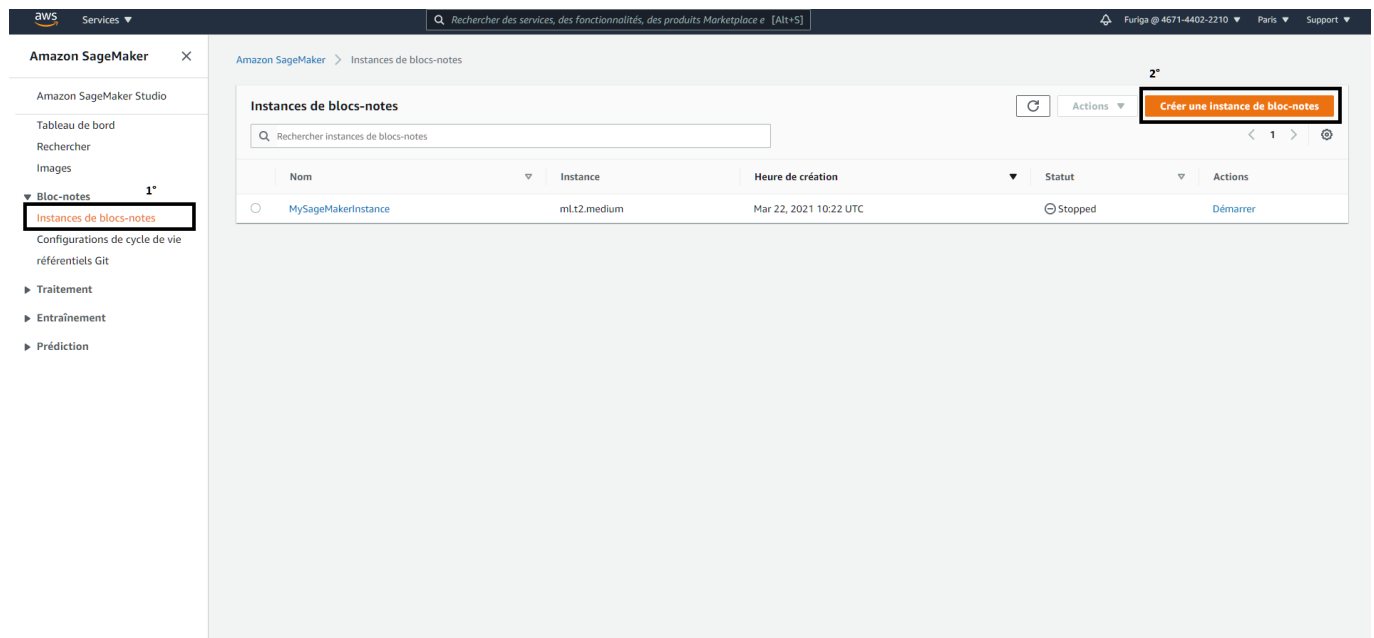
Creation d'un environnement de travail

1. Connection à la console Amazon SageMaker

Lorsque vous cliquez [ici](#), AWS Management Console s'ouvre dans une nouvelle fenêtre. Vous pouvez ainsi garder ce guide pas à pas ouvert. Commencez à saisir SageMaker dans la barre de recherche et sélectionnez Amazon SageMaker pour ouvrir la console de service.

2. Créez une instance de notebook (bloc-notes)

Ouvrez le menu déroulant "bloc-notes", cliquez sur "Instances de blocs-notes" (1), puis sur "Créer une instance de "bloc-notes" (2)



Suivez ensuite les étapes suivantes :

- Saisissez un nom pour votre bloc-notes (3)
- Dans le champ rôle IAM, choisissez Créer un nouveau rôle pour demander à Amazon SageMaker de créer un rôle avec les autorisations requises et de l'attribuer à votre instance. Vous pouvez également choisir un rôle IAM existant dans votre compte à cette fin. (4)

Amazon SageMaker fournit des instances de blocs-notes entièrement gérées et préconçues exécutant les blocs-notes Jupyter. Les instances de blocs-notes incluent un exemple de code pour les exercices courants d'hébergement et d'entraînement de modèles. [En savoir plus](#)

Paramètres d'instances de blocs-notes

Nom de l'instance de bloc-notes 3*
 Tutoriel AWS Simplon

Type d'instance de bloc-notes
 ml.t2.medium

► Configuration supplémentaire

Autorisations et chiffrement

Rôle IAM
 Les instances de blocs-notes nécessitent des autorisations pour appeler d'autres services, y compris SageMaker et S3. Choisissez un rôle ou laissez-nous en créer un avec la [AmazonSageMakerFullAccess](#) stratégie IAM attachée.

AmazonSageMaker-ExecutionRole-20210322T102218

Créer un rôle 4*
 Entrer un ARN de rôle IAM personnalisé
 Utiliser le rôle existant
 AmazonSageMaker-ExecutionRole-20210322T102218
 Choisissez vos données de blocs-notes : choisissez une clé KMS existante ou saisissez un ARN de clé.
 Aucun chiffrement personnalisé

► Réseau - facultatif

- Dans la boîte Créer un rôle IAM, sélectionnez "Tout compartiment S3" (5). Cela permet à votre instance Amazon SageMaker d'accéder à tous les compartiments S3 de votre compte. Plus loin dans ce tutoriel, vous créerez un nouveau compartiment S3. Toutefois, si vous avez un compartiment que vous souhaitez utiliser, sélectionnez Compartiments S3 spécifiques et indiquez le nom du compartiment. Sélectionnez Créer le rôle. (6)

Créer un rôle IAM

La transmission d'un rôle IAM donne à Amazon SageMaker l'autorisation d'exécuter des actions dans d'autres services AWS à votre place. La création d'un rôle ici accordera les autorisations décrites par la [AmazonSageMakerFullAccess](#) stratégie IAM pour le rôle que vous créez. Le rôle IAM que vous créez fournira un accès aux éléments suivants :

☒ **Compartiments S3 que vous spécifiez - facultatif**

☒ **Tout compartiment S3**
 Autorisez les utilisateurs qui ont accès à votre instance de blocs-notes à accéder à n'importe quel compartiment et à son contenu dans votre compte.

☐ **Compartiments S3 spécifiques**
 Exemple : nom-compartiment-1, nom-com2
 Délimités par des virgules. ARN, « * » et « / » ne sont pas pris en charge.

☐ **Aucun**

☒ **Tout compartiment S3 comportant « sagemaker » dans son nom**

☒ **Tout objet S3 comportant « sagemaker » dans son nom**

☒ **Tout objet S3 comportant la balise « sagemaker » et la valeur « vrai »** [Afficher le balisage des objets](#)

☒ **Compartiment S3 avec une stratégie de compartiment autorisant l'accès à SageMaker** [Afficher les stratégies de compartiments S3](#)

Annuler **Créer un rôle** 6*

- Notez qu'Amazon SageMaker a créé un rôle appelé AmazonSageMaker-ExecutionRole-*** pour vous. Pour les besoins de ce tutoriel, nous utiliserons les valeurs par défaut des autres champs. Choisissez Créer une instance de blocs-notes. (7)

Services

Rechercher des services, des fonctionnalités, des produits Marketplace e [Alt+S]

Furiga @ 4671-4402-2210 Paris Support

Configuration supplémentaire

Autorisations et chiffrement

Rôle IAM
Les instances de bloc-notes nécessitent des autorisations pour appeler d'autres services, y compris SageMaker et S3. Choisissez un rôle ou laissez-nous en créer un avec la [AmazonSageMakerFullAccess](#) stratégie IAM attachée.

AmazonSageMaker-ExecutionRole-20210322T133160

Réussite ! Vous avez créé un rôle IAM.
[AmazonSageMaker-ExecutionRole-20210322T133160](#)

Accès racine - facultatif
☒ Enable (Activer): accorder aux utilisateurs l'accès racine au bloc-notes
☐ Disable (Désactiver): ne pas accorder aux utilisateurs l'accès racine au bloc-notes
Les configurations de cycle de vie ont toujours un accès racine.

Clé de chiffrement - facultatif
Chiffrez vos données de bloc-notes. Choisissez une clé KMS existante ou saisissez un ARN de clé.

Aucun chiffrement personnalisé

Réseau - facultatif

référentiels Git - facultatif

Balises - facultatif

Annuler **Créer une instance de bloc-notes** 7

Il faut ensuite patienter quelques minutes afin que la création du bloc-notes se fasse.

Services

Rechercher des services, des fonctionnalités, des produits Marketplace e [Alt+S]

Furiga @ 4671-4402-2210 Paris Support

Amazon SageMaker

Amazon SageMaker Studio

Tableau de bord

Rechercher

Images

Bloc-notes

Instances de bloc-notes

Configurations de cycle de vie

référentiels Git

Traitement

Entraînement

Prédiction

Réussite ! Votre instance de bloc-notes est en cours de création.
Ouvrez l'instance de bloc-notes lorsque l'état est défini sur En service et ouvrez un bloc-notes de modèle pour démarrer.

Afficher les détails

Amazon SageMaker > Instances de bloc-notes

Instances de bloc-notes

Rechercher instances de bloc-notes

Actions

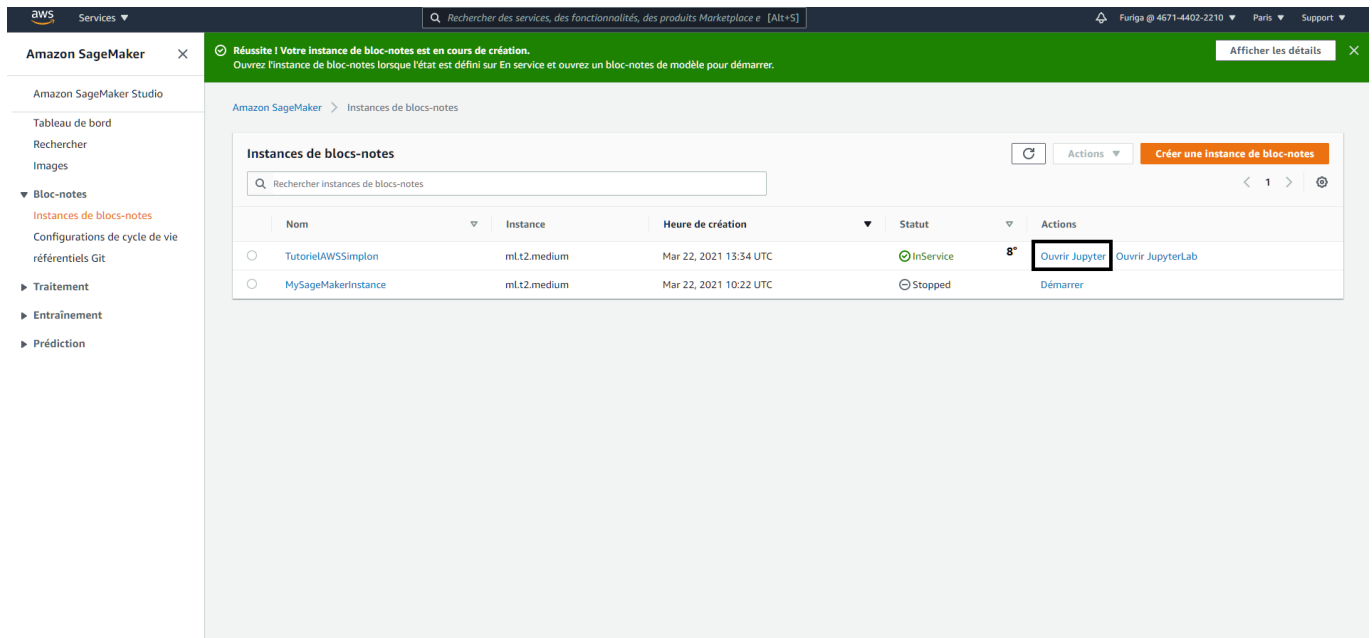
Créer une instance de bloc-notes

	Nom	Instance	Heure de création	Statut	Actions
<input type="radio"/>	TutorielAWSsimplon	ml.t2.medium	Mar 22, 2021 13:34 UTC	Pending	-
<input type="radio"/>	MySageMakerInstance	ml.t2.medium	Mar 22, 2021 10:22 UTC	Stopped	Démarrer

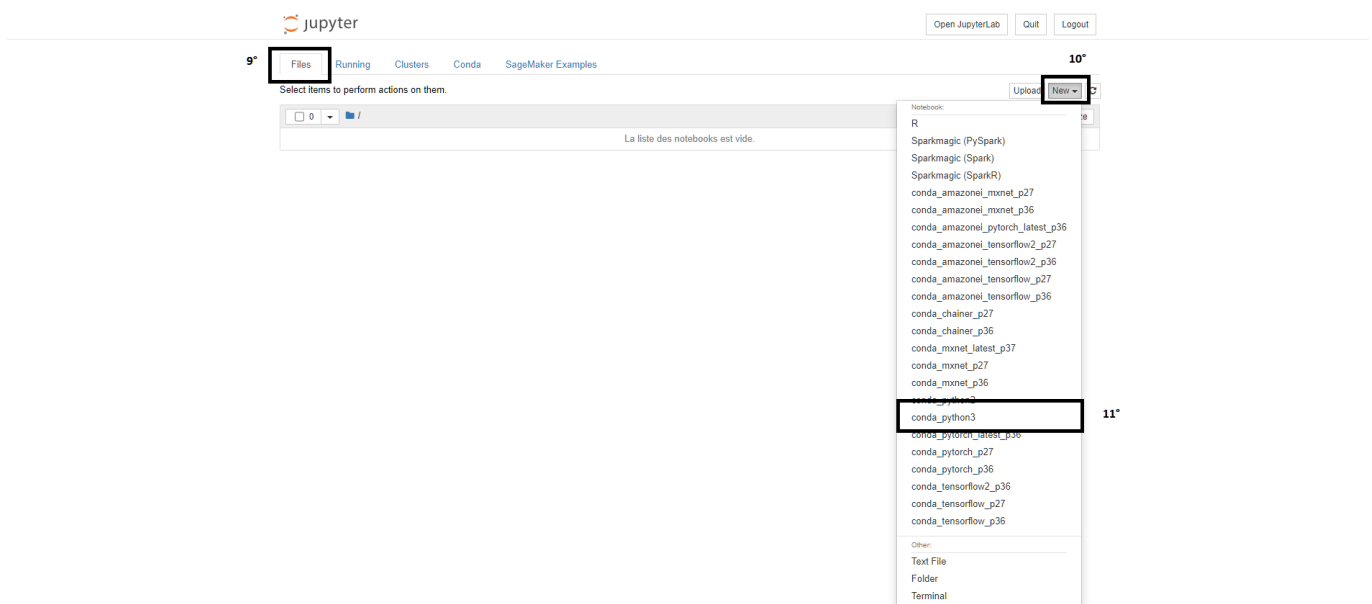
Commentaires Français

© 2008 - 2021, Amazon Web Services, Inc. ou ses sociétés affiliées. Tous droits réservés. Politique de confidentialité Conditions d'utilisation Préférences relatives aux cookies

Une fois celui-ci créé, le statut passe de "Pending" à "InService". On peut donc accéder à celui-ci en cliquant sur "Ouvrir Jupyter" (8)



Une fois Jupyter ouvert, allez dans l'onglet "Files" (9) et choisissez "New" (10), puis choisissez "conda_python3" (11).



3. Preparation des données

Pour préparer les données, puis entraîner et déployer le modèle de ML, vous devrez importer certaines bibliothèques et définir quelques variables d'environnement dans votre environnement Jupyter. Copiez le code suivant dans la cellule de code de votre instance et sélectionnez Exécuter.

```
# Import des librairies
import boto3, re, sys, math, json, os, sagemaker, urllib.request
from sagemaker import get_execution_role
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from IPython.display import Image
from IPython.display import display
from time import gmtime, strftime
from sagemaker.predictor import Predictor
from sagemaker.serializers import CSVSerializer

# Définir le rôle IAM
role = get_execution_role()
prefix = 'sagemaker/DEMO-xgboost-dm'
# Conteneurs disponibles pour XGB00ST
containers = {'us-west-2': '433757028032.dkr.ecr.us-west-2.amazonaws.com/xgboost:latest',
              'us-east-1': '811284229777.dkr.ecr.us-east-1.amazonaws.com/xgboost:latest',
              'us-east-2': '825641698319.dkr.ecr.us-east-2.amazonaws.com/xgboost:latest',
              'eu-west-1': '685385470294.dkr.ecr.eu-west-1.amazonaws.com/xgboost:latest'} # chaque région a son propre container
my_region = boto3.session.Session().region_name # selectionner la région de l'instance
print("Succès - l'instance de SageMaker est dans la region " + my_region + ".  
Utilisez le conteneur " + containers[my_region] + " pour le endpoint SageMaker.")
```

Dans cette étape, vous créez un compartiment S3 qui stockera vos données pour ce tutoriel.

Copiez le code suivant dans la prochaine cellule de code de votre bloc-notes et changez le nom du compartiment S3 pour le personnaliser. Les noms de compartiments S3 doivent être uniques au monde et comporter certaines autres restrictions et limitations.

Sélectionnez Exécuter. Si vous ne recevez pas de message de confirmation de réussite de l'opération, changez le nom du compartiment et réessayez.

```
# Création d'un compartiment S3 pour stocker les données de ce tutoriel
bucket_name = 'VARIABLE-A-MODIFIER' # <--- VARIABLE A CHANGER AVEC VOTRE PROPRE
NOM UNIQUE
s3 = boto3.resource('s3')
try:
    if my_region == 'us-east-1':
        s3.create_bucket(Bucket=bucket_name)
    else:
        s3.create_bucket(Bucket=bucket_name, CreateBucketConfiguration={
'LocationConstraint': my_region })
    print('S3 bucket, création réussie.')
except Exception as e:
    print('S3 erreur: ',e)
```

Ensuite, vous devez télécharger les données sur votre instance Amazon SageMaker et les charger dans une trame de données. Copiez et exécutez le code suivant :

```
# Téléchargement des données sur notre instance SageMaker
try:
    urllib.request.urlretrieve ("https://d1.awsstatic.com/tmt/build-train-deploy-
machine-learning-model-
sagemaker/bank_clean.27f01fbbdf43271788427f3682996ae29ceca05d.csv",
"bank_clean.csv")
    print('Succès: téléchargement de bank_clean.csv effectué.')
except Exception as e:
    print('Erreur chargement des données: ',e)

# Chargement des données pour utilisation par XGBOOST
try:
    model_data = pd.read_csv('./bank_clean.csv',index_col=0)
    print('Succès: données importées dans le dataframe.')
except Exception as e:
    print('Erreur chargement des données: ',e)
```

Nous allons à présent réorganiser les données et les diviser en données d'entraînement et en données de test.

Les données d'entraînement (70 % des clients) seront utilisées pendant la boucle d'entraînement du modèle. Nous utiliserons l'optimisation par gradient pour affiner itérativement les paramètres du modèle. L'optimisation par gradient est un moyen de trouver les valeurs des paramètres du modèle qui minimisent l'erreur du modèle, en utilisant le gradient de la fonction de perte du modèle.

Les données de test (30 % des clients restants) seront utilisées pour évaluer la performance du modèle et mesurer la mesure dans laquelle le modèle entraîné se généralise à des données invisibles.

Copiez le code suivant dans une nouvelle cellule de code et sélectionnez Exécuter pour réorganiser et diviser les données :

```
# Réorganisation des données
# On divise en jeu d'entraînement et jeu de test
train_data, test_data = np.split(model_data.sample(frac=1, random_state=1729),
[0.7 * len(model_data)])
print(train_data.shape, test_data.shape)
```

Un petit point sur XGBoost :

XGBoost est une bibliothèque distribuée et optimisée d'amplification de gradient conçue pour être très efficace, flexible et portable. Elle implémente des algorithmes d'apprentissage automatique dans le cadre de Gradient Boosting (technique de machine learning pour la régression et la classification de problèmes). XGBoost fournit un boost d'arborescence parallèle (également connu sous le nom de GBDT, GBM) qui résout de nombreux problèmes de science des données de manière rapide et précise. Le même code fonctionne sur les principaux environnements distribués (Hadoop, SGE, MPI) et peut résoudre des problèmes au-delà de milliards d'exemples.

Pour utiliser un modèle XGBoost précréé par Amazon SageMaker, vous devrez reformater l'en-tête et la première colonne des données d'entraînement et charger les données du compartiment S3.

Copiez le code suivant dans une nouvelle cellule de code et sélectionnez Exécuter pour reformater et charger les données :

```
# Reformatage de l'en-tête de la première colonne des données d'entraînement
pd.concat([train_data['y_yes'], train_data.drop(['y_no', 'y_yes'], axis=1)],
axis=1).to_csv('train.csv', index=False, header=False)
# Chargement des données du compartiment S3
boto3.Session().resource('s3').Bucket(bucket_name).Object(os.path.join(prefix,
'train/train.csv')).upload_file('train.csv')
s3_input_train =
sagemaker.TrainingInput(s3_data='s3://{}/{}/train'.format(bucket_name, prefix),
content_type='csv')
```

Vous devez ensuite configurer la séance Amazon SageMaker, créer une instance du modèle XGBoost (un estimateur), et définir les hyperparamètres du modèle. Copiez le code suivant dans une nouvelle cellule de code et sélectionnez Exécuter :

```
sess = sagemaker.Session()
xgb = sagemaker.estimator.Estimator(containers[my_region], role, instance_count=1,
instance_type='ml.m4.xlarge', output_path='s3://{}/{}/output'.format(bucket_name,
prefix), sagemaker_session=sess)
xgb.set_hyperparameters(max_depth=5, eta=0.2, gamma=4, min_child_weight=6, subsample=0.8,
silent=0, objective='binary:logistic', num_round=100)
```

Une fois les données chargées et l'estimateur XGBoost configuré, entraînez le modèle avec l'optimisation de gradient sur une instance ml.m4.xlarge en copiant le code suivant dans la prochaine cellule de code et en sélectionnant Exécuter.

Au bout de quelques minutes, vous devriez commencer à voir les journaux de l'entraînement qui sont générés.

```
xgb.fit({'train': s3_input_train})
```

4. Deployer le modèle

Pour déployer le modèle sur un serveur et créer un point de terminaison auquel vous pouvez accéder, copiez le code suivant dans la prochaine cellule de code et sélectionnez Exécuter :

```
xgb_predictor = xgb.deploy(initial_instance_count=1,instance_type='ml.m4.xlarge')
```

Pour prévoir si les clients dans les données de test se sont inscrits pour le produit bancaire ou non, copiez le code suivant dans la prochaine cellule de code et sélectionnez Exécuter :

```
test_data_array = test_data.drop(['y_no', 'y_yes'], axis=1).values #Charger les data dans un array
xgb_predictor.serializer = CSVSerializer() # selectionner le type de serialiser
predictions = xgb_predictor.predict(test_data_array).decode('utf-8') # predict!
predictions_array = np.fromstring(predictions[1:], sep=',') # transformer les predictions en array
print(predictions_array.shape)
```


5. Evaluer la performance du modèle

Dans cette étape, vous évaluez la performance et la précision du modèle de machine learning. Copiez et collez le code ci-dessous et sélectionnez Exécuter pour comparer les valeurs réelles et prédites dans un tableau appelé matrice de confusion.

Sur la base de cette prédiction, nous pouvons conclure que vous avez prédit qu'un client s'inscrira pour un certificat de dépôt avec une précision de 90 % dans les données de test, avec une précision de 65 % (278/429) pour les inscrits et 90 % (10 785/11 928) pour les non inscrits.

```
cm = pd.crosstab(index=test_data['y_yes'], columns=np.round(predictions_array),
rownames=['Observed'], colnames=['Predicted'])
tn = cm.iloc[0,0]; fn = cm.iloc[1,0]; tp = cm.iloc[1,1]; fp = cm.iloc[0,1]; p =
(tp+tn)/(tp+tn+fp+fn)*100
print("\n{0:<20}{1:<4.1f}%\n".format("Taux de classification: ", p))
print("{0:<15}{1:<15}{2:>8}".format("Prediction", "Non Acheteur", "Acheteur"))
print("Observé")
print("{0:<15}{1:<2.0f}% ({2:<}){3:>6.0f}% ({4:<})".format("Non Acheteur",
tn/(tn+fn)*100,tn, fp/(tp+fp)*100, fp))
print("{0:<16}{1:<1.0f}% ({2:<}){3:>7.0f}% ({4:<}) \n".format("Acheteur",
fn/(tn+fn)*100,fn, tp/(tp+fp)*100, tp))
```

Retour :

Taux de classification: 89.5%

Prediction	Non Acheteur	Acheteur
Observé		
Non Acheteur	90% (10769)	37% (167)
Acheteur	10% (1133)	63% (288)

6. Supprimer vos ressources

Dans cette étape, vous supprimerez vos ressources liées à Amazon SageMaker.

Important : il est conseillé de supprimer les ressources qui ne sont pas utilisées de façon active, afin de réduire les coûts. La non-suppression de ressources peut entraîner des frais.

Copiez, collez et exécutez le code suivant pour supprimer le point de terminaison Amazon SageMaker et les objets dans votre compartiment S3 :

```
sagemaker.Session().delete_endpoint(xgb_predictor.endpoint_name)
bucket_to_delete = boto3.resource('s3').Bucket(bucket_name)
bucket_to_delete.objects.all().delete()
```