Capitolo 1

Presentazione di Amazon SageMaker

I professionisti del *machine learning* utilizzano un'ampia raccolta di strumenti per sviluppare i loro progetti: librerie open source, framework di deep learning e molto altro ancora. Inoltre, spesso devono scrivere i propri strumenti di automazione e controllo. La gestione di questi strumenti e della relativa infrastruttura è dispendiosa in termini di tempo e soggetta a errori.

Questo è esattamente il problema affrontato da Amazon SageMaker (`https://aws.amazon.com/sagemaker`). Amazon SageMaker è un servizio completamente gestito che può aiutarvi a creare e distribuire rapidamente i vostri modelli di machine learning. Che siate solo alle prime armi con il machine learning o che siate professionisti esperti, troverete che le funzionalità di SageMaker rendono più agili i flussi di lavoro e migliorano le prestazioni dei modelli. Sarete in grado di concentrarvi al 100% sul problema in questione, senza perdere tempo a installare, gestire e ridimensionare gli strumenti e l'infrastruttura di machine learning.

In questo primo capitolo scopriremo le principali funzionalità di SageMaker, come esse aiutano a risolvere gli ostacoli incontrati dai professionisti del machine learning e come configurare SageMaker.

In questo capitolo

Requisiti tecnici

Per seguire gli esempi presentati in questo capitolo avrete bisogno di un account AWS. Se non lo avete, puntate il vostro browser su `https://aws.amazon.com/getting-started` per conoscere AWS e i suoi concetti

fondamentali e per creare un account AWS. Dovreste anche familiarizzare con il piano gratuito AWS Free Tier (`https://aws.amazon.com/free`), che permette di impiegare gratuitamente molti servizi di AWS entro determinati limiti di utilizzo.
Dovrete installare e configurare l'interfaccia a riga di comando (CLI) di AWS per il vostro account (`https://aws.amazon.com/cli`).
Avrete bisogno di un ambiente Python 3.x perfettamente operativo. L'installazione della distribuzione Anaconda (`https://www.anaconda.com`) non è obbligatoria ma è fortemente consigliata, in quanto include molti progetti di cui avremo bisogno (Jupyter, `pandas`, `numpy` e altro).
Gli esempi di codice presentati nel libro sono disponibili su GitHub all'indirizzo `https://github.com/PacktPublishing/Learn-Amazon-SageMaker-second-edition`. Per accedervi dovrete installare un client Git (`https://git-scm.com`).

Le funzionalità di Amazon SageMaker

Amazon SageMaker è stato lanciato in occasione dell'evento *AWS re:Invent 2017*. Da allora, gli sono state aggiunte molte nuove funzionalità: potete vedere l'elenco completo (in continuo sviluppo) su `https://aws.amazon.com/about-aws/whats-new/machine-learning`.
In questa parte del capitolo esamineremo le funzionalità principali di Amazon SageMaker e scopriremo quali sono le sue finalità. Non preoccupatevi di afferrare tutto subito: ci immergeremo in ognuno di essi nei capitoli successivi. Parleremo anche delle API (*Application Programming Interface*) di SageMaker e degli SDK (*Software Development Kit*) che le implementano.

Le principali funzionalità di Amazon SageMaker

Amazon SageMaker nasce per aiutare a preparare, costruire, sottoporre a training, ottimizzare e distribuire modelli di machine learning su un'infrastruttura completamente gestita e a qualsiasi scala. Ciò permette di concentrarsi sullo studio e sulla risoluzione del problema di machine learning, invece di dedicare tempo e risorse alla creazione e gestione dell'infrastruttura. In breve, potete passare dalla creazione al training fino all'implementazione molto più rapidamente. Esaminiamo ogni passaggio ed evidenziamo le relative funzionalità di SageMaker.

Preparazione

Amazon SageMaker offre alcuni potenti strumenti per etichettare e preparare i dataset.

- *Amazon SageMaker Ground Truth*: annota dataset di qualsiasi dimensione. Offre i flussi di lavoro per i casi d'uso più diffusi (rilevamento di immagini, estrazione di entità e molto altro) e potete implementarne di vostri. I lavori di annotazione possono essere distribuiti a operatori dell'ambito privato, di terzi o pubblico.
- *Amazon SageMaker Processing*: potete eseguire job batch di elaborazione dei dati (e altre attività come la valutazione del modello) utilizzando codice che potete scrivere con scikit-learn o Spark.

- *Amazon SageMaker Data Wrangler*: utilizzando un'interfaccia grafica, permette di applicare centinaia di trasformazioni integrate (o anche le vostre trasformazioni) a dataset tabulari e di esportare il risultato con un clic su un notebook Jupyter.
- *Amazon SageMaker Feature Store*: permette di memorizzare offline in Amazon S3 le funzionalità progettate, per creare dataset oppure online per utilizzarle in fase di predizione.
- *Amazon SageMaker Clarify*: utilizzando vari parametri statistici, analizza le potenziali distorsioni presenti nei vostri dataset e modelli, e spiega le predizioni dei modelli.

Realizzazione

Amazon SageMaker offre due ambienti di sviluppo.

- *Istanze notebook*: istanze Amazon EC2 completamente gestite, preinstallate con gli strumenti e le librerie più diffusi, come Jupyter, Anaconda e così via.
- *Amazon SageMaker Studio*: un ambiente di sviluppo end-to-end integrato per progetti di machine learning, che fornisce un'interfaccia grafica intuitiva per molte delle funzionalità di SageMaker. Studio è ora il modo preferenziale per eseguire i notebook e vi consigliamo di utilizzarlo al posto delle istanze notebook.

Nel campo degli algoritmi sperimentabili, potete scegliere tra i seguenti.

- Una raccolta di *17 algoritmi built-in* per il machine learning e il deep learning, già implementati e ottimizzati per essere eseguiti in modo efficiente su AWS. Non è più necessario scrivere il codice di machine learning.
- Un'intera raccolta di framework open source built-in (*TensorFlow, PyTorch, Apache MXNet, scikit-learn e altri*), in cui dovete semplicemente integrare il vostro codice.
- Il vostro codice, in esecuzione in un vostro container: codice Python, R, C++, Java e così via.
- Algoritmi e modelli con pre-training per il machine learning dall'AWS Marketplace (`https://aws.amazon.com/marketplace/solutions/machine-learning`).
- Soluzioni di machine learning e modelli all'avanguardia, a portata di clic in *Amazon SageMaker JumpStart*.

Inoltre, *Amazon SageMaker Autopilot* utilizza l'apprendimento AutoMachine per creare, sottoporre a training e ottimizzare automaticamente i modelli senza la necessità di scrivere una sola riga di codice di machine learning.

Training

Come abbiamo accennato in precedenza, Amazon SageMaker si occupa della fornitura e della gestione dell'infrastruttura di training. Non dovrete perdere tempo a gestire i server e potrete invece concentrarvi sul lavoro di machine learning. Inoltre, SageMaker offre funzionalità avanzate come le seguenti.

- *Storage gestito*, utilizzando Amazon S3, Amazon EFS o Amazon FSx for Lustre a seconda dei vostri requisiti prestazionali.
- *Training spot gestito*, utilizzando per il training istanze spot Amazon EC2, al fine di ridurre i costi anche dell'80%.

- Il *training distribuito* suddivide automaticamente i job di training su larga scala su un intero cluster di istanze gestite, utilizzando tecniche avanzate come il parallelismo dei dati e del modello.
- La *modalità pipe* invia da Amazon S3 alle istanze di training dataset infinitamente grandi, evitandovi la necessità di copiare i dati da un luogo all'altro.
- L'*ottimizzazione automatica del modello* effettua l'ottimizzazione degli iperparametri per ottenere più rapidamente modelli ad alta accuratezza.
- *Amazon SageMaker Experiments* tiene traccia, organizza e confronta con facilità tutti i vostri job SageMaker.
- Durante il training, *Amazon SageMaker Debugger* cattura lo stato interno del modello, lo ispeziona per osservare in quale modo apprende il modello, rileva eventuali condizioni indesiderate che riducono l'accuratezza e profila le prestazioni del job di training.

Distribuzione

Proprio come con il training, Amazon SageMaker si occupa di tutta l'infrastruttura di distribuzione e offre tutta una serie di funzionalità aggiuntive.

- Gli *endpoint real-time* creano un'API HTTPS che fornisce predizioni dal vostro modello. Come è lecito aspettarsi, è disponibile l'autoscaling.
- La *trasformazione batch* utilizza un modello per eseguire le predizioni sui dati in modalità batch.
- *Amazon Elastic Inference* aggiunge l'accelerazione frazionata su GPU agli endpoint basati su CPU, per trovare il miglior rapporto costo/prestazioni per la vostra infrastruttura di predizione.
- *Amazon SageMaker Model Monitor* cattura i dati inviati a un endpoint e li confronta con un valore base per identificare e segnalare problemi di qualità dei dati (valori mancanti, derive dei dati e altro).
- *Amazon SageMaker Neo* compila modelli per una specifica architettura hardware, comprese le piattaforme embedded, e distribuisce una versione ottimizzata utilizzando un runtime leggero.
- *Amazon SageMaker Edge Manager* vi aiuta a distribuire e gestire i vostri modelli sui dispositivi edge.
- Ultimo, ma non meno importante, *Amazon SageMaker Pipelines* vi permette di creare pipeline end-to-end automatizzate per eseguire e gestire i carichi di lavoro di preparazione dei dati, training e distribuzione.

L'API Amazon SageMaker

Proprio come tutti gli altri servizi di AWS, Amazon SageMaker è governato da API implementate negli SDK per i linguaggi supportati da AWS (`https://aws.amazon.com/tools`). Inoltre, è disponibile anche un SDK dedicato per Python, noto anche come SageMaker SDK. Esaminiamoli entrambi e vediamo quali vantaggi offrono.

Gli SDK per i linguaggi di AWS

Gli SDK per i linguaggi implementano API specifiche per tutti i servizi di AWS: S3, EC2 e così via. Naturalmente, includono anche le API SageMaker, che sono documentate qui: `https://docs.aws.amazon.com/sagemaker/latest/dg/api-and-sdk-reference.html`.
Parlando di data science e machine learning, Python è il linguaggio più utilizzato, quindi vediamo le API SageMaker disponibili in `boto3`, l'SDK AWS per il linguaggio Python (`https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/sagemaker.html`). Si tratta di API di basso livello e dettagliate: per esempio, `create_training_job()` offre molti parametri JSON che non sono particolarmente chiari (ne vedete alcuni nella Figura 1.1). Potreste pensare che la cosa non sia molto comoda per la sperimentazione quotidiana del machine learning... Siamo totalmente d'accordo.

```
response = client.create_training_job(
    TrainingJobName='string',
    HyperParameters={
        'string': 'string'
    },
    AlgorithmSpecification={
        'TrainingImage': 'string',
        'AlgorithmName': 'string',
        'TrainingInputMode': 'Pipe'|'File',
        'MetricDefinitions': [
            {
                'Name': 'string',
                'Regex': 'string'
            },
        ],
        'EnableSageMakerMetricsTimeSeries': True|False
    },
    RoleArn='string',
    InputDataConfig=[
        {
            'ChannelName': 'string',
            'DataSource': {
                'S3DataSource': {
                    'S3DataType': 'ManifestFile'|'S3Prefix'|'AugmentedManifestFil
                    'S3Uri': 'string',
                    'S3DataDistributionType': 'FullyReplicated'|'ShardedByS3Key',
                    'AttributeNames': [
                        'string',
                    ]
                },
                'FileSystemDataSource': {
                    'FileSystemId': 'string',
                    'FileSystemAccessMode': 'rw'|'ro',
                    'FileSystemType': 'EFS'|'FSxLustre',
                    'DirectoryPath': 'string'
                }
```

Figura 1.1 Una vista (parziale) dell'API create_training_job() in boto3.

In effetti, queste API a livello di servizio non sono concepite per essere utilizzate per la sperimentazione su notebook. Il loro scopo è l'automazione, tramite script personalizzati o strumenti Infrastructure as Code come AWS CloudFormation (`https://aws.amazon.com/cloudformation`) e Terraform (`https://terraform.io`). Il team di sviluppo le utilizzerà per gestire la produzione, dove avrà bisogno di avere il pieno controllo di ogni possibile parametro.
Quindi, che cosa dovreste usare per la sperimentazione? Amazon SageMaker SDK.

Amazon SageMaker SDK

Amazon SageMaker SDK (`https://github.com/aws/sagemaker-python-sdk`) è un SDK Python specifico per Amazon SageMaker. Potete trovare la sua documentazione su `https://sagemaker.readthedocs.io/en/stable`.

> **NOTA**
>
> È stato compiuto ogni sforzo per verificare il funzionamento degli esempi di codice di questo libro con l'ultimo SageMaker SDK (v2.58.0 al momento della scrittura di queste pagine).

Qui, il livello di astrazione è molto più elevato: l'SDK contiene oggetti per i modelli, gli estimator, i predittori e così via. Siamo decisamente tornati nel territorio del machine learning.
Per esempio, questo SDK semplifica enormemente l'avvio di un job di training (una riga di codice) e la distribuzione di un modello (una riga di codice). I problemi legati all'infrastruttura vengono completamente astratti, e così possiamo concentrarci sul machine learning. Ecco un esempio. Non preoccupatevi di cogliere i dettagli, per ora:

```
# Configurazione del job di training
my_estimator = TensorFlow(
    entry_point='my_script.py',
    role=my_sagemaker_role,
    train_instance_type='machine learning.p3.2xlarge',
    instance_count=1,
    framework_version='2.1.0'
)

# Training del modello
my_estimator.fit('s3://my_bucket/my_training_data/')
# Distribuzione del modello su un endpoint HTTPS
my_predictor = my_estimator.deploy(
    initial_instance_count=1,
    instance_type='machine learning.c5.2xlarge'
)
```

Ora che sappiamo qualcosa in più su Amazon SageMaker, vediamo come occorre configurarlo.

Configurazione di Amazon SageMaker sul computer locale

Un malinteso comune è che non potete utilizzare SageMaker al di fuori del cloud AWS. Ovviamente, si tratta di un servizio basato sul cloud e le sue funzionalità più interessanti richiedono l'esecuzione dell'infrastruttura cloud. Tuttavia, molti sviluppatori preferiscono configurare a modo loro il proprio ambiente di sviluppo, e SageMaker permette di farlo: in questo paragrafo imparerete a installare SageMaker SDK sul vostro computer locale o su un server locale. Nei capitoli successivi imparerete a sottoporre a training e distribuire i modelli in locale.
È buona norma isolare gli ambienti Python, per evitare che si sviluppi un vero "inferno" di dipendenze. Vediamo come potete raggiungere questo obiettivo utilizzando due noti progetti: `virtualenv` (`https://virtualenv.pypa.io`) e Anaconda (`https://www.anaconda.com`).

Installazione di SageMaker SDK con virtualenv

Se non avete mai utilizzato `virtualenv`, leggete questo tutorial prima di procedere: `https://packaging.python.org/en/latest/guides/installing-using-pip-and-virtual-environments`.

1. Per prima cosa, create un nuovo ambiente chiamato `sagemaker` e attivatelo:

```
$ mkdir workdir
$ cd workdir
$ python3 -m venv sagemaker
$ source sagemaker/bin/activate
```

2. Ora installate `boto3`, SageMaker SDK e la libreria `pandas` (`https://pandas.pydata.org`), anch'essa un requisito:

```
$ pip3 install boto3 sagemaker pandas
```

3. Ora controllate rapidamente di poter importare questi SDK in Python:

```
$ python3
Python 3.9.5 (default, May  4 2021, 03:29:30)
>>> import boto3
>>> import sagemaker
>>> print(boto3.__version__)
1.17.70
>>> print(sagemaker.__version__)
2.39.1
>>> exit()
```

L'installazione sembra a posto. Le versioni che vedrete saranno sicuramente più recenti e va bene così. Ora, eseguite un rapido test con un server Jupyter locale (`https://jupyter.org`). Se sul vostro computer non è installato Jupyter, potete trovare le istruzioni su `https://jupyter.org/install`.

1. Innanzitutto, create un kernel Jupyter basato sul vostro ambiente virtuale:

```
$ pip3 install jupyter ipykernel
$ python3 -m ipykernel install --user --name=sagemaker
```

2. Quindi, potete avviare Jupyter:

```
$ jupyter notebook
```

3. Creando un nuovo notebook, potete vedere che il kernel `sagemaker` è disponibile, quindi selezionatelo nel menu *New*, come vedete nella Figura 1.2:

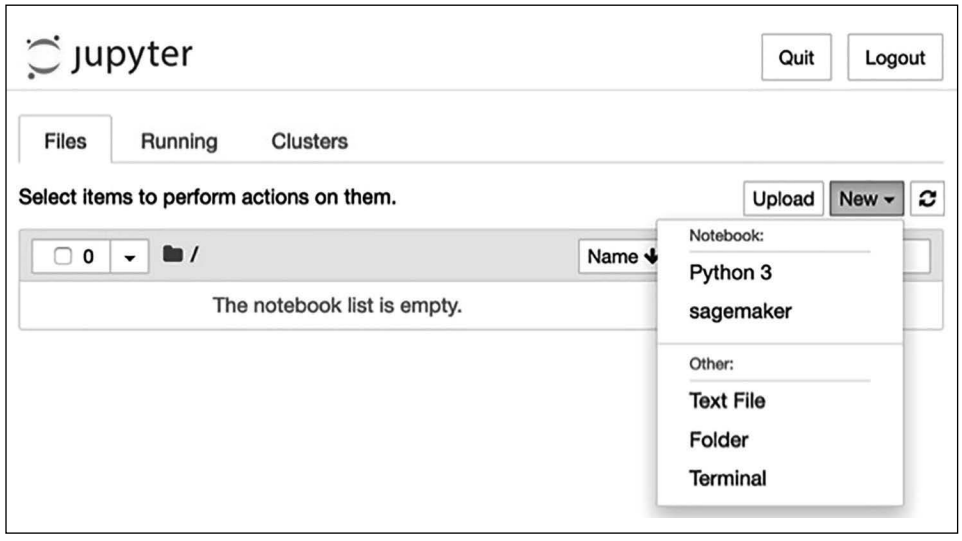

Figura 1.2 Creazione di un nuovo notebook.

4. Infine, potete verificare che gli SDK siano disponibili, importandoli e stampando la loro versione, come vedete nella Figura 1.3:

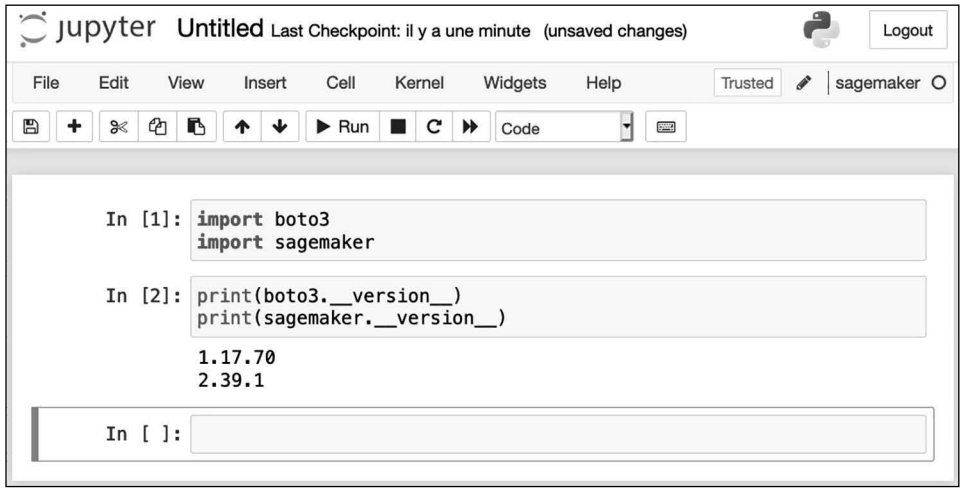

Figura 1.3 Verifica della versione dell'SDK.

Questo completa l'installazione con `virtualenv`. Non dimenticate di terminare Jupyter e di disattivare il vostro `virtualenv`:

```
$ deactivate
```

Potete installare l'SDK anche usando Anaconda.

Installazione di SageMaker SDK con Anaconda

Anaconda include un gestore di pacchetti, `conda`, che permette di creare e gestire ambienti isolati. Se non avete mai utilizzato *conda*, dovreste fare le seguenti cose.

- Installare Anaconda: `https://docs.anaconda.com/anaconda/install`.
- Leggere questo tutorial: `https://docs.conda.io/projects/conda/en/last/user-guide/getting-started.html`.

Inizieremo utilizzando i seguenti passi.

1. Create e attivate un nuovo ambiente `conda` chiamato `conda-sagemaker`:

   ```
   $ conda create -y -n conda-sagemaker
   $ conda activate conda-sagemaker
   ```

2. Quindi installate `pandas`, `boto3` e SageMaker SDK. Quest'ultimo deve essere installato con `pip`, in quanto non è disponibile come pacchetto `conda`:

   ```
   $ conda install -y boto3 pandas
   $ pip3 install sagemaker
   ```

3. Ora aggiungete all'ambiente Jupyter e le sue dipendenze, e create un nuovo kernel:

   ```
   $ conda install -y jupyter ipykernel
   $ python3 -m ipykernel install --user --name conda-sagemaker
   ```

4. Quindi, potete avviare Jupyter:

   ```
   $ jupyter notebook
   ```

   Verificate che il kernel `conda-sagemaker` sia presente nel menu `New`, come vedete nella Figura 1.4.

5. Proprio come nel paragrafo precedente, potete creare un notebook utilizzando questo kernel e verificare che gli SDK siano stati importati correttamente.

Questo completa l'installazione con `conda`. Il fatto di preferirlo a `virtualenv` è in gran parte una questione di preferenze personali. Potete sicuramente eseguire tutti i notebook di questo libro e realizzare i vostri progetti con l'uno o l'altro.

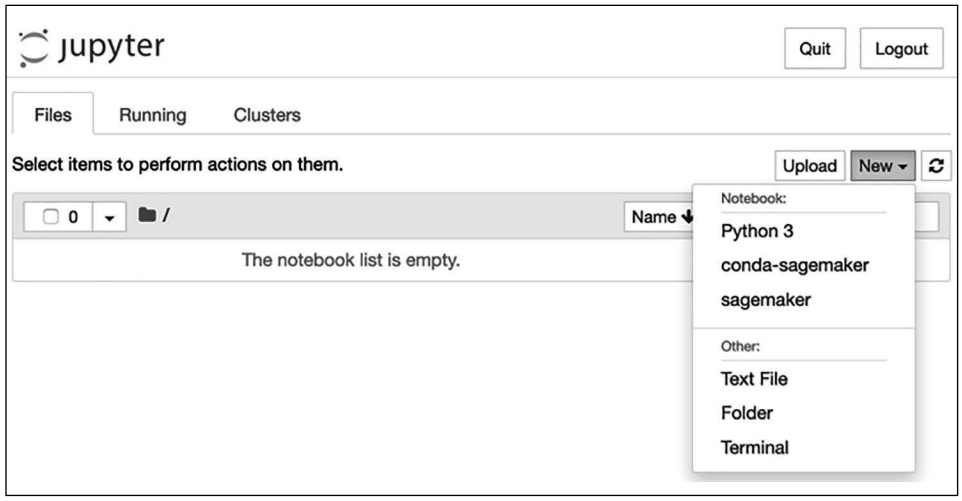

Figura 1.4 Creazione di un nuovo ambiente conda.

Una parola sulle autorizzazioni di AWS

Il sistema Amazon IAM (*Identity and Access Management*) vi permette di gestire in modo sicuro l'accesso ai servizi e alle risorse di AWS (`https://aws.amazon.com/iam`). Naturalmente, questo vale anche per Amazon SageMaker e dovete assicurarvi che il vostro utente AWS disponga di autorizzazioni sufficienti per richiamare l'API SageMaker.

Autorizzazioni IAM

Se non avete particolare dimestichezza con IAM, consultate la seguente documentazione: `https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html`

Potete eseguire un test rapido utilizzando l'interfaccia a riga di comando di AWS su una delle API SageMaker, per esempio `list-endpoints`. Qui usiamo la regione `eu-west-1` qui, ma potete impiegare la regione che preferite:

```
$ aws sagemaker list-endpoints --region eu-west-1
{
    "Endpoints": []
}
```

Se ottenete un messaggio d'errore per autorizzazioni insufficienti, dovrete aggiornare il ruolo IAM associato al vostro utente AWS.
Se possedete l'account AWS in questione, potete farlo facilmente nella console IAM, aggiungendo al vostro ruolo la policy gestita `AmazonSageMakerFullAccess`. Notate che questa policy è estremamente permissiva: va bene per un account di sviluppo, ma certamente non per un account di produzione.

Se invece lavorate con un account per il quale non disponete dei diritti amministrativi (come un account fornito dall'azienda per cui lavorate), contattate il vostro amministratore IT perché aggiunga al vostro utente AWS le autorizzazioni SageMaker.
Per ulteriori informazioni sulle autorizzazioni di SageMaker, fate riferimento alla documentazione: `https://docs.aws.amazon.com/sagemaker/latest/dg/security-iam.html`.

Configurazione di Amazon SageMaker Studio

La sperimentazione è una parte fondamentale del processo di machine learning. Sviluppatori e data scientist utilizzano tutta una serie di strumenti e librerie open source per l'esplorazione e l'elaborazione dei dati e, naturalmente, per valutare gli algoritmi candidati. L'installazione e la manutenzione di questi strumenti richiede una discreta quantità di tempo, che probabilmente sarebbe meglio speso per studiare il problema da risolvere.
Amazon SageMaker Studio vi offre tutti gli strumenti di machine learning di cui avete bisogno, dalla sperimentazione alla produzione. Alla base c'è un ambiente di sviluppo integrato basato su Jupyter che lo rende immediatamente familiare.
Inoltre, SageMaker Studio è dotato di altre funzionalità di SageMaker, come SageMaker Experiments, per monitorare e confrontare tutti i job, SageMaker Autopilot, per creare automaticamente modelli di machine learning, e molto altro ancora. Molte operazioni possono essere eseguite con pochi clic, senza nemmeno dover scrivere del codice.
SageMaker Studio semplifica ulteriormente la gestione dell'infrastruttura: non dovrete creare istanze notebook: SageMaker Studio vi offre ambienti di elaborazione prontamente disponibili per i vostri notebook.

NOTA

La lettura di questo paragrafo richiede una conoscenza di base di Amazon S3, Amazon VPC e Amazon IAM. Se non sapete di cosa si tratta, consultate la seguente documentazione:

```
https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html
https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html
https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html
```

Questo sarebbe probabilmente anche un buon momento per dare un'occhiata (e aggiungere un segnalibro) alla pagina dei prezzi per SageMaker: `https://aws.amazon.com/it/sagemaker/pricing`.

Accesso ad Amazon SageMaker Studio

Potete accedere a SageMaker Studio utilizzando una di queste tre opzioni.

- *La procedura di avvio rapido*: questa è l'opzione più semplice per i singoli account; la analizzeremo nei prossimi paragrafi.
- *AWS SSO (Single Sign-On)*: se la vostra azienda ha configurato un'applicazione SSO, questa è probabilmente l'opzione migliore. Per informazioni su SSO, consultate: `https://docs.aws.amazon.com/sagemaker/latest/dg/onboard-sso-users.html`. Contattate l'amministratore IT per i dettagli.

- *Amazon IAM*: se la vostra azienda non utilizza SSO, questa è probabilmente l'opzione migliore. Per informazioni su IAM, consultate: `https://docs.aws.amazon.com/sagemaker/latest/dg/onboard-iam.html`. Anche in questo caso, contattate il vostro amministratore IT per i dettagli.

Accesso con la procedura di avvio rapido

Ci sono diversi passaggi per svolgere la procedura di avvio rapido.

1. Innanzitutto, aprite la Console AWS in una delle regioni in cui è disponibile Amazon SageMaker Studio, nel nostro specifico caso `https://us-east-2.console.aws.amazon.com/sagemaker`.
2. Come vedete nella Figura 1.5, il riquadro verticale a sinistra ha un link ad *Amazon SageMaker Studio*.

Figura 1.5 Apertura di SageMaker Studio.

3. Facendo clic su questo link si apre la schermata di accesso; potete vedere la sua prima sezione nella Figura 1.6.

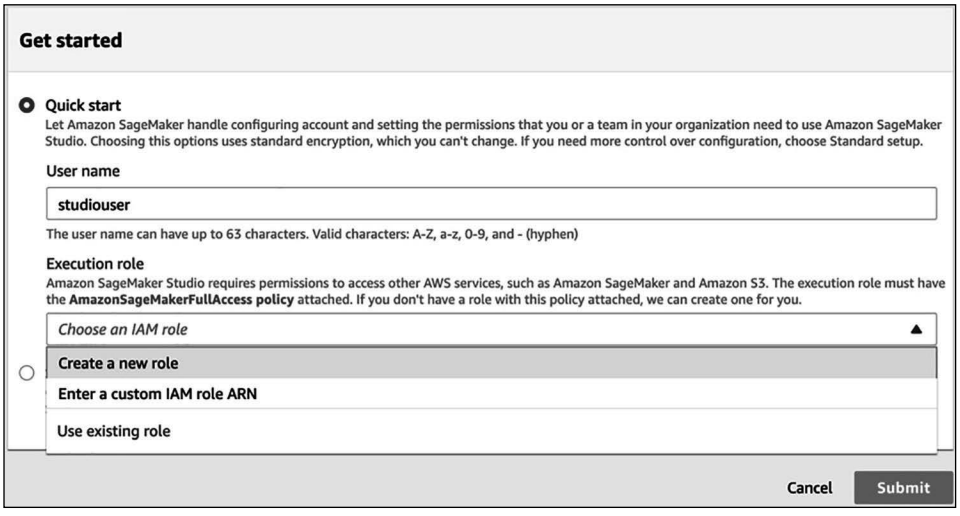

Figura 1.6 Avvio rapido con Quick start.

4. Selezionate *Quick start*. Quindi, inserite il nome utente da utilizzare per accedere a SageMaker Studio e create un nuovo ruolo IAM, come mostrato nella Figura 1.6. Si aprirà la schermata rappresentata nella Figura 1.7.

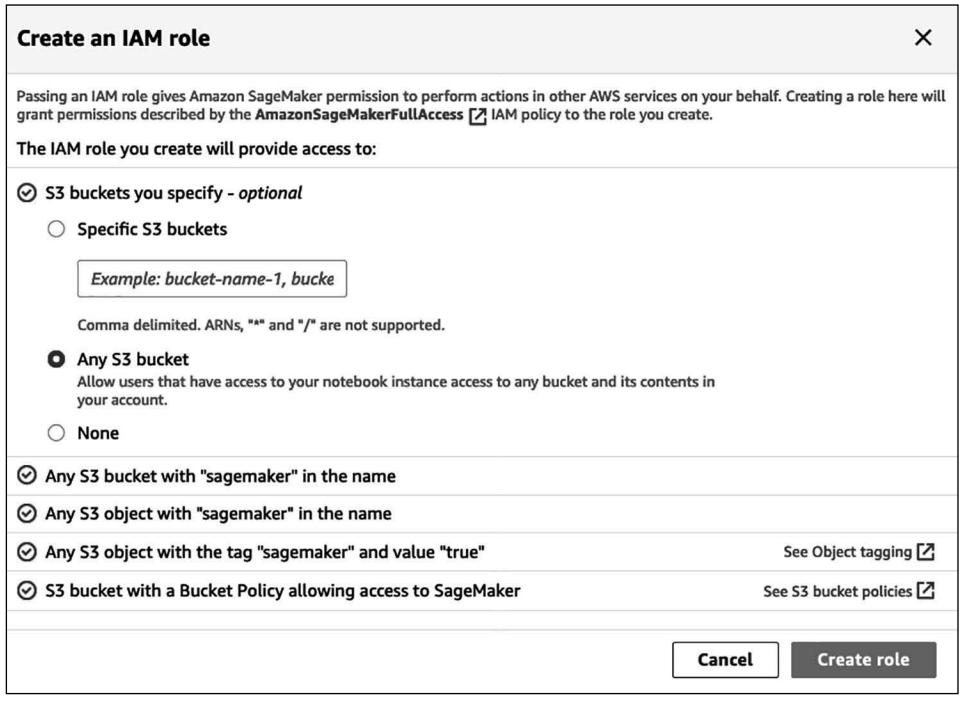

Figura 1.7 Creazione di un ruolo IAM.

   L'unica decisione che dovete prendere qui è se volete consentire alla vostra istanza notebook di accedere a specifici bucket Amazon S3. Selezionate *Any S3 bucket* e fate clic su *Create role*. Questa è l'impostazione più flessibile per lo sviluppo e il test, ma dovrete applicare impostazioni molto più rigide in produzione. Naturalmente, potete modificare questo ruolo in un secondo momento nella console IAM o crearne uno nuovo.
5. Dopo aver fatto clic su *Create role*, tornate alla schermata precedente. Assicuratevi che i template dei progetti e JumpStart siano abilitati per questo account (questa dovrebbe essere l'impostazione di default).
6. Non vi resta che fare clic su *Submit* per avviare la procedura di accesso. A seconda della configurazione dell'account, potreste ottenere una schermata aggiuntiva che vi chiede di selezionare un VPC e una sottorete. Vi consiglio di selezionare qualsiasi sottorete nel VPC di default.
7. Pochi minuti dopo, SageMaker Studio sarà attivo, come vedete nella Figura 1.8. Potete aggiungere anche altri utenti, se necessario, ma per ora fate clic su *Open Studio*.

Non preoccupatevi se l'operazione richiede qualche minuto, poiché SageMaker Studio deve completare la configurazione della prima esecuzione del vostro ambiente. Come vedete nella Figura 1.9, una volta aperto SageMaker Studio, vedrete il familiare layout di JupyterLab.

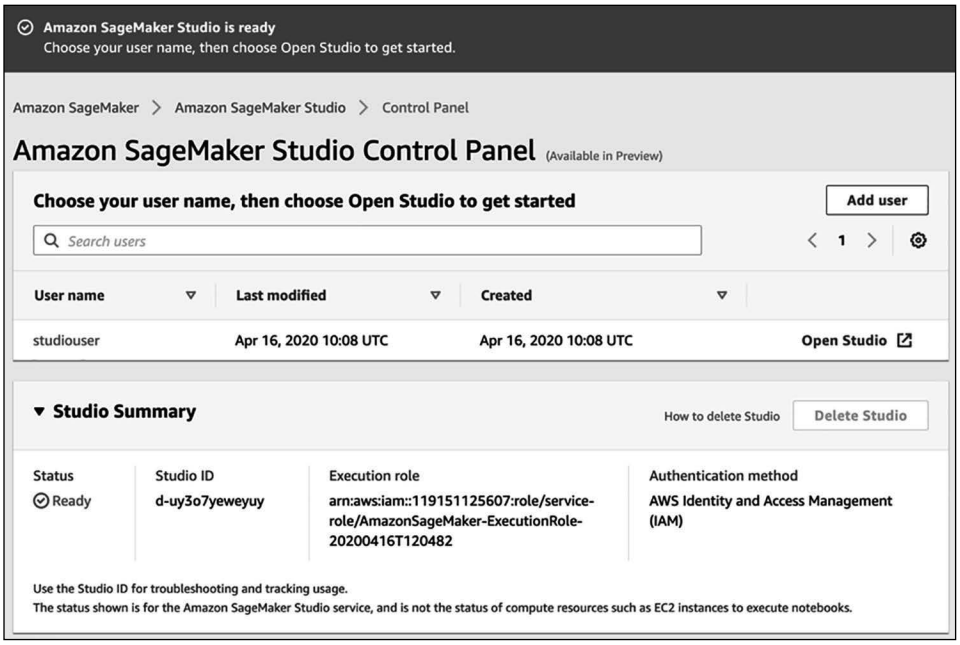

Figura 1.8 Avvio di SageMaker Studio.

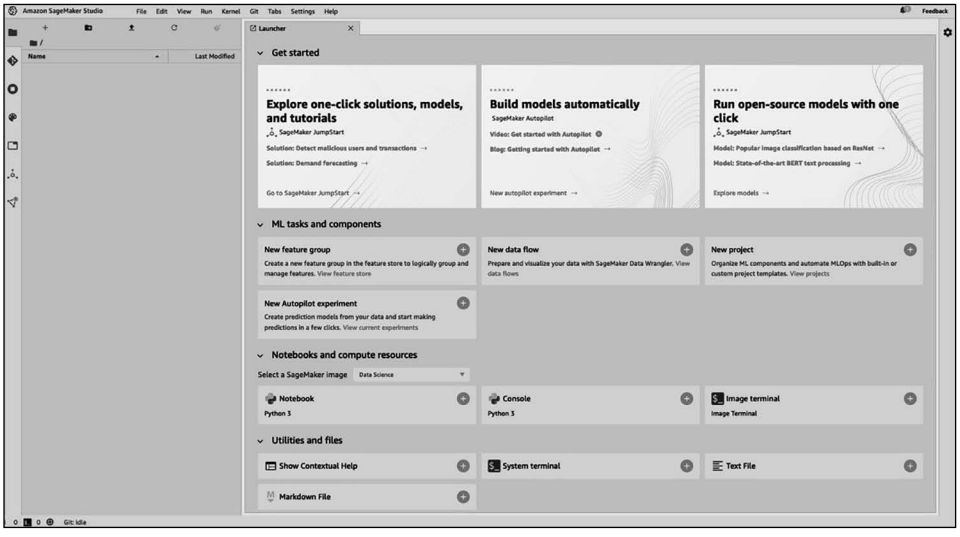

Figura 1.9 La schermata di benvenuto di SageMaker Studio.

NOTA

SageMaker Studio è un ambiente in divenire. Nel momento in cui starete leggendo queste pagine, alcune schermate potrebbero essere state aggiornate. Inoltre, potreste notare piccole differenze da una regione all'altra, poiché alcune funzionalità o tipi di istanze non sono disponibili.

8. Potete creare subito il vostro primo notebook. Nella scheda *Launcher*, nella sezione *Notebooks and compute resources*, selezionate *Data Science* e fate clic su *Notebook – Python 3*.
9. Questo apre un notebook, come vedete nella Figura 1.10. Per prima cosa controllate che gli SDK siano disponibili. Poiché questa è la prima volta che avviate il kernel *Data Science*, dovrete attendere un paio di minuti.

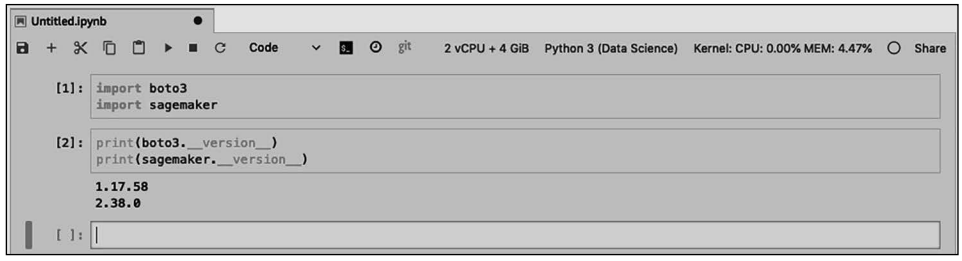

Figura 1.10 Verifica della versione dell'SDK.

10. Come vedete nella Figura 1.11, potete facilmente elencare le risorse attualmente in esecuzione nella vostra istanza di Studio: un'istanza *machine learning.t3.medium*, l'immagine data science che supporta il kernel utilizzato nel notebook e il notebook stesso.

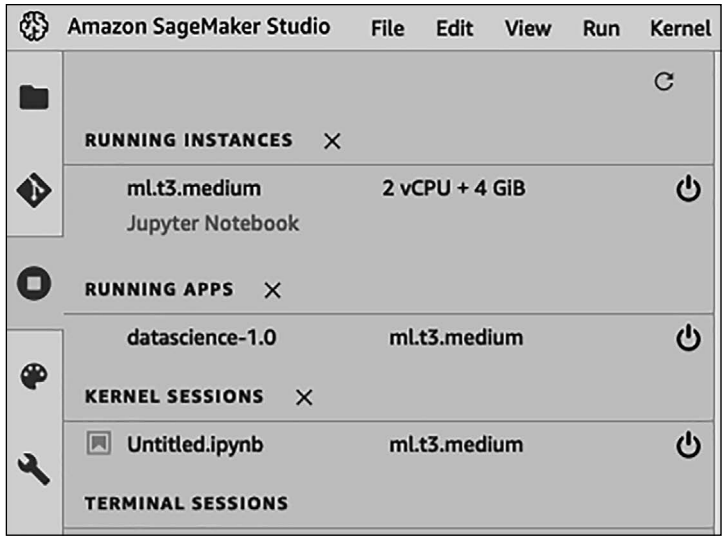

Figura 1.11 Visualizzazione delle risorse di Studio.

11. Per evitare costi inutili, dovreste chiudere queste risorse quando avrete finito di utilizzarle. Per esempio, potete chiudere l'istanza e tutte le risorse in esecuzione su di essa, come potete vedere nella Figura 1.12. Ma non fatelo ora, avrete bisogno dell'istanza per eseguire i prossimi esempi.

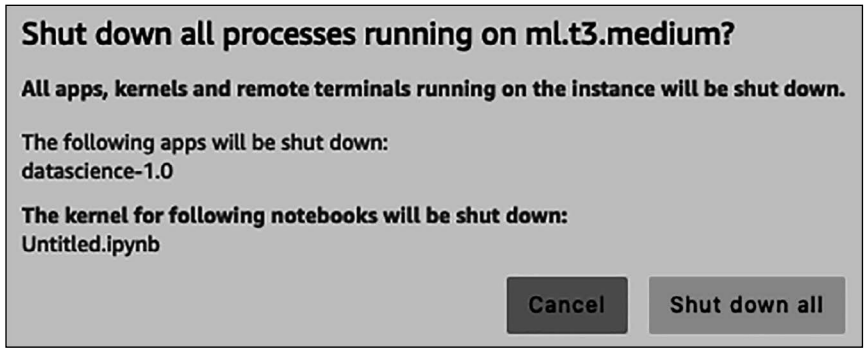

Figura 1.12 Chiusura di un'istanza.

12. *Machine learning.t3.medium* è la dimensione di default dell'istanza utilizzata da Studio. Potete passare ad altri tipi di istanza facendo clic su *2 vCPU + 4 GiB* nella parte superiore del vostro notebook. Ciò vi permette di selezionare una nuova dimensione per l'istanza e di lanciarla in Studio. Dopo alcuni minuti, l'istanza sarà attiva e il codice del vostro notebook sarà stato migrato automaticamente. Non dimenticate di chiudere l'istanza precedente, come descritto in precedenza.
13. Quando avrete finito di lavorare con SageMaker Studio, tutto ciò che dovete fare è chiudere la scheda del browser. Per riprendere a lavorare, vi basta tornare alla console di SageMaker e fare clic su *Open Studio*.
14. Per chiudere l'istanza di Studio, selezionate semplicemente *Shut Down* nel menu *File*. Tutti i file verrebbero comunque conservati fino all'eliminazione completa di Studio nella console di SageMaker.

Ora che abbiamo completato la configurazione, sono sicuro che sarete impazienti di iniziare a provare il machine learning. Iniziamo a distribuire alcuni modelli.

Distribuzione di soluzioni e modelli one-clic con Amazon SageMaker JumpStart

Se non sapete nulla del machine learning, potreste avere difficoltà a iniziare ad affrontare progetti reali. Avete eseguito tutti gli "esempi giocattolo" e avete letto vari post sul livello dei modelli di VISIONE AL COMPUTER O ELABORAZIONE DEL LINGUAGGIO NATURALE. E adesso? Come potete iniziare a utilizzare questi modelli sui vostri dati per risolvere i vostri problemi?

Anche se siete professionisti esperti, la creazione di soluzioni di machine learning end-to-end non è un compito facile. Il training e la distribuzione dei modelli sono solo una parte dell'equazione: che dire della preparazione dei dati, dell'automazione e così via?

Amazon SageMaker JumpStart è stato creato appositamente per aiutare a iniziare più rapidamente i progetti di machine learning. In, letteralmente, un clic, potete distribuire quanto segue.

- 16 soluzioni end-to-end per problemi operativi reali come il rilevamento delle frodi nelle transazioni finanziarie, la motivazione delle decisioni di credito, la manutenzione predittiva e molto altro ancora.
- Oltre 180 modelli TensorFlow e PyTorch con pre-training su una varietà di attività di visione artificiale ed elaborazione del linguaggio naturale.
- Altre risorse di apprendimento, come notebook di esempio, post di blog e tutorial video.

È ora di distribuire una soluzione.

Distribuzione di una soluzione

Cominciamo.

1. Partendo dalla barra delle icone a sinistra, aprite JumpStart. La Figura 1.13 mostra la schermata di apertura.

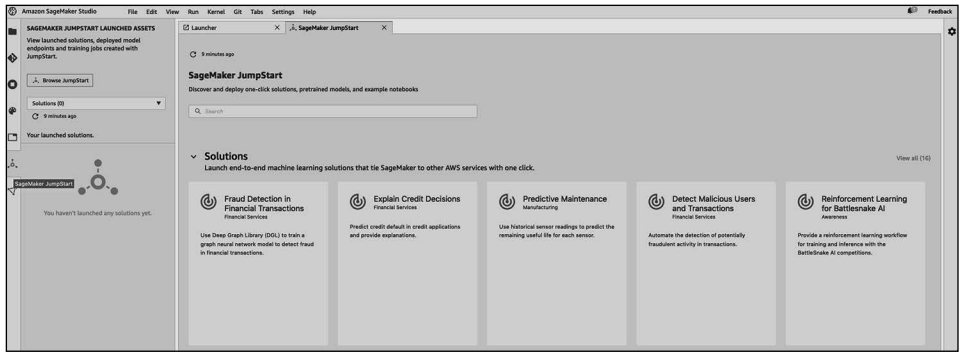

Figura 1.13 Visualizzazione delle soluzioni in JumpStart.

2. Selezionate *Fraud Detection in Financial Transactions*. Come potete vedere nella Figura 1.14, questo è un esempio affascinante che utilizza dati grafici e reti neurali a grafo per predire le attività fraudolente sulla base delle interazioni.
3. Dopo aver letto i dettagli della soluzione, tutto ciò che dovete fare è fare clic sul pulsante *Launch*. Lancerete un template AWS CloudFormation incaricato di creare tutte le risorse AWS richieste dalla soluzione.

CloudFormation

Se siete curiosi di sapere che cos'è CloudFormation, potreste trovare utile questa introduzione: `https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/Welcome.html`.

4. Pochi minuti dopo, la soluzione sarà pronta, come potete vedere nella Figura 1.15. Fate clic su *Open Notebook* per aprire il primo notebook.

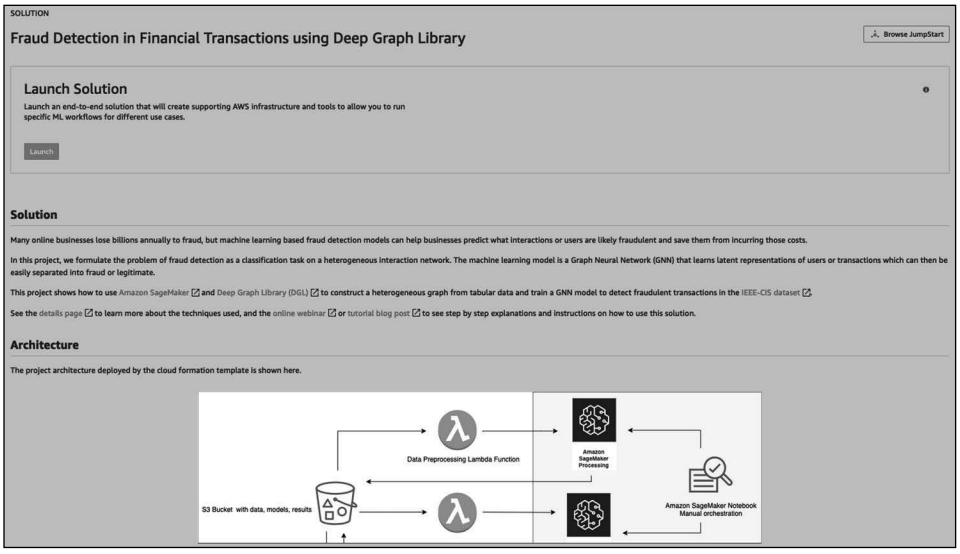

Figura 1.14 I dettagli della soluzione.

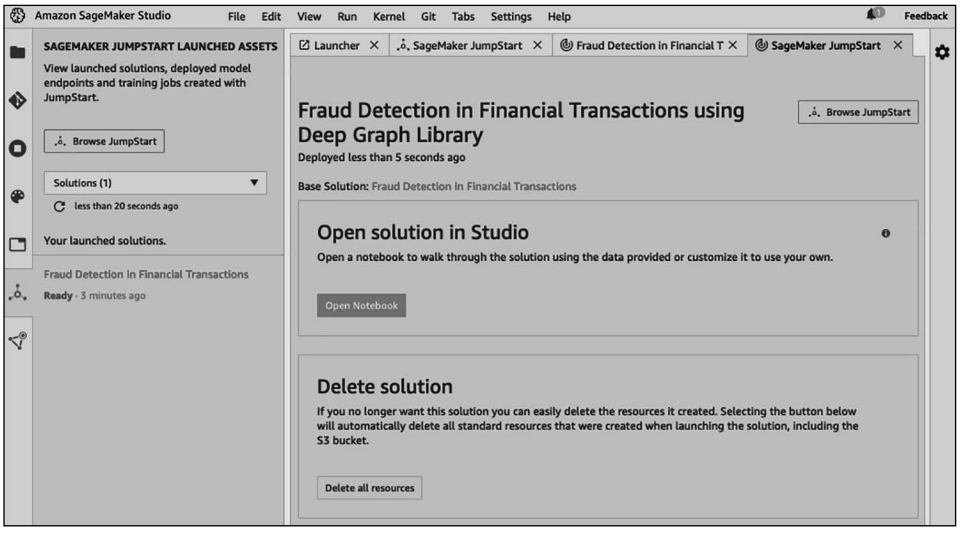

Figura 1.15 Apertura di una soluzione.

5. Come potete vedere nella Figura 1.16, potete sfogliare i file della soluzione nel riquadro a sinistra: notebook, codice di training e così via.
6. Da qui in poi, potete iniziare a eseguire e modificare il notebook. Se non avete ancora familiarità con SageMaker SDK, non preoccupatevi di seguire tutti i dettagli.
7. Al termine, tornate alla pagina della soluzione e fate clic su *Delete all resources* per ripulire ed evitare costi inutili, come vedete nella Figura 1.17.

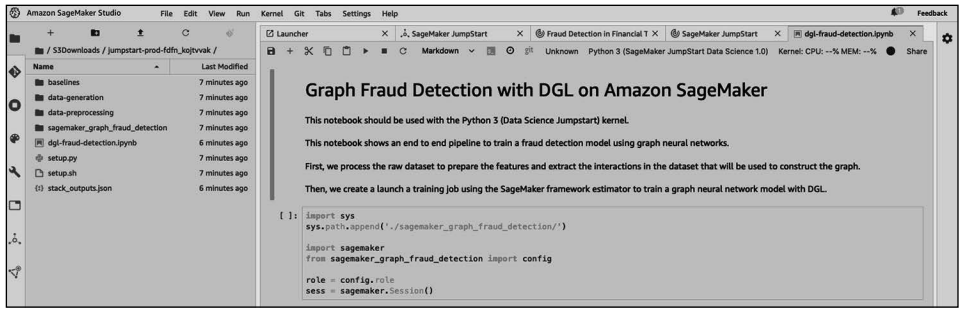

Figura 1.16 Visualizzazione dei file della soluzione.

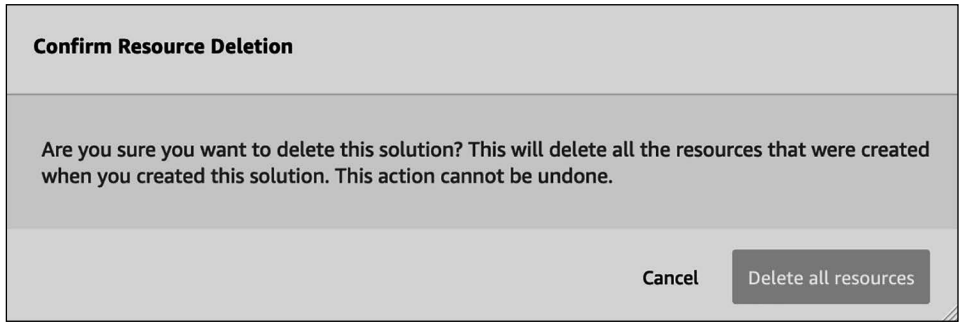

Figura 1.17 Eliminazione di una soluzione.

Come potete vedere, le soluzioni JumpStart sono un ottimo modo per esplorare come si possono risolvere i problemi operativi con il machine learning e per iniziare a pensare a come potreste fare lo stesso nel vostro ambiente aziendale.
Ora, vediamo come si distribuiscono i modelli con pre-training.

Distribuzione di un modello

JumpStart include oltre 180 modelli TensorFlow e PyTorch con pre-training su una grande varietà di attività di visione artificiale ed elaborazione del linguaggio naturale. Esaminiamo i modelli di computer vision.

1. Partendo dalla schermata principale di JumpStart, aprite *Vision models*, come potete vedere nella Figura 1.18.
2. Supponiamo che siate interessati a provare i modelli di rilevamento degli oggetti basati sull'architettura SSD (*Single Shot Detector*). Fate clic sul modello *SSD* dal riquadro *PyTorch* (il quarto da sinistra).
3. Si apre la pagina dei dettagli del modello, che ci dice da dove proviene il modello, su quale dataset è stato sottoposto a training e quali etichette è in grado di predire. Potete anche selezionare su quale tipo di istanza distribuire il modello. Rimanendo all'impostazione di default, fate clic su *Deploy* per distribuire il modello su un endpoint real-time, come vedete nella Figura 1.19.

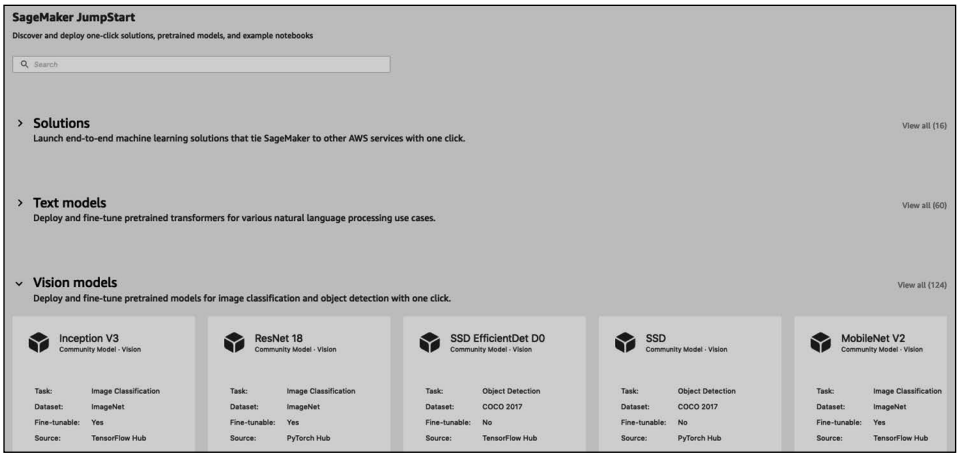

Figura 1.18 Visualizzazione dei modelli di computer vision.

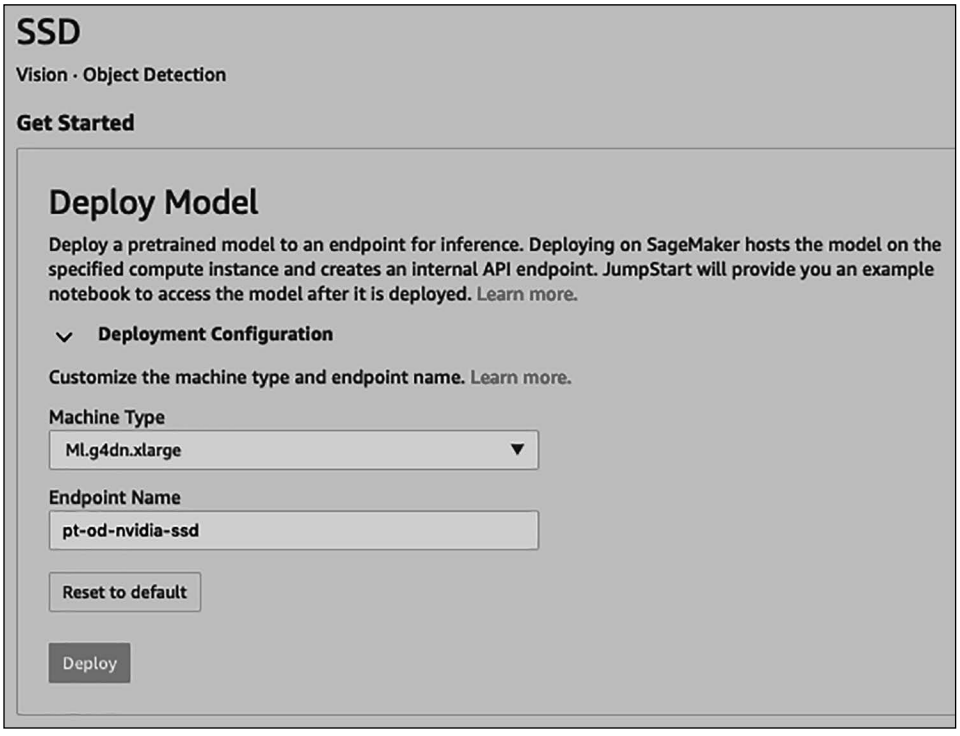

Figura 1.19 Distribuzione di un modello JumpStart.

4. Pochi minuti dopo, il modello sarà già stato distribuito. Come potete vedere nella Figura 1.20, nel riquadro a sinistra potete vedere lo stato dell'endpoint; fate semplicemente clic su *Open Notebook* per sottoporlo a test.

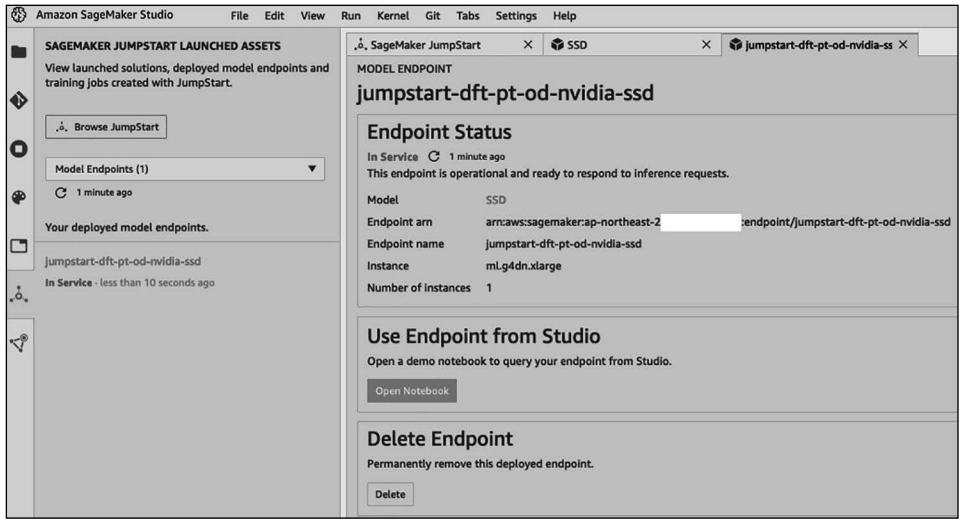

Figura 1.20 Apertura di un notebook JumpStart.

5. Facendo clic sulle celle del notebook, scaricate un'immagine di test e scopriamo quali oggetti contiene. Nella Figura 1.21 sono visibili i riquadri di delimitazione, le classi e le probabilità.

Figura 1.21 Rilevamento degli oggetti contenuti in un'immagine.

6. Quando avrete finito, assicuratevi di eliminare l'endpoint per evitare addebiti inutili: fate semplicemente clic su *Delete* nella schermata dei dettagli dell'endpoint, visibile nella Figura 1.20.

JumpStart non solo semplifica enormemente la sperimentazione di modelli all'avanguardia, ma fornisce anche codice che potete utilizzare facilmente nei vostri progetti: caricamento di un'immagine per la predizione, predizione con un endpoint, tracciamento dei risultati e così via.
Per quanto siano utili i modelli con pre-training, spesso abbiamo bisogno di metterli a punto sui nostri dataset. Vediamo come farlo con JumpStart.

Ottimizzazione di un modello

Useremo un modello per la classificazione delle immagini.

> **NOTA**
>
> Un avvertimento sull'ottimizzazione dei modelli per i testi: l'ottimizzazione di modelli complessi come BERT può richiedere molto tempo, a volte diverse ore per epoca su una singola GPU. Oltre al lungo tempo di attesa, il costo non sarà trascurabile, quindi vi consiglio di evitare questi esempi, a meno che non abbiate un progetto operativo reale su cui lavorare.

1. Selezionate il modello *Resnet 18* (il secondo da sinistra nella Figura 1.18).
2. Nella pagina dei dettagli del modello, potete vedere che questo modello può essere ottimizzato su un dataset di default disponibile per il test (un dataset TensorFlow con cinque classi di fiori) o sul vostro dataset archiviato in S3. Scorrendo verso il basso, potete vedere quale formato dovrebbe avere il vostro dataset.
3. Come potete vedere nella Figura 1.22, ci atterremo al dataset di default. Lasciate invariati anche la configurazione della distribuzione e i parametri di training. Quindi, fate clic su *Train* per avviare il job di ottimizzazione.
4. Dopo pochi minuti, l'ottimizzazione sarà terminata (ecco perché ho scelto proprio questo esempio). Potete vedere il percorso di output in S3 in cui è stato archiviato il modello ottimizzato. Prendete nota di quel percorso; ne avrete bisogno tra un minuto.
5. Quindi, fate clic su *Deploy*, come avete fatto nell'esempio precedente. Una volta che il modello è stato distribuito, aprite il notebook di esempio, che spiega come eseguire predizioni con questo primo modello con pre-training.
6. Questo notebook utilizza le immagini del dataset originale su cui il modello è stato sottoposto a pre-training. Nessun problema, adattiamolo! Anche se non avete ancora familiarità con SageMaker SDK, il notebook è abbastanza semplice da consentirvi di capire tutto quello che sta succedendo e aggiungere alcune celle per predire l'immagine di un fiore con il vostro modello ottimizzato.

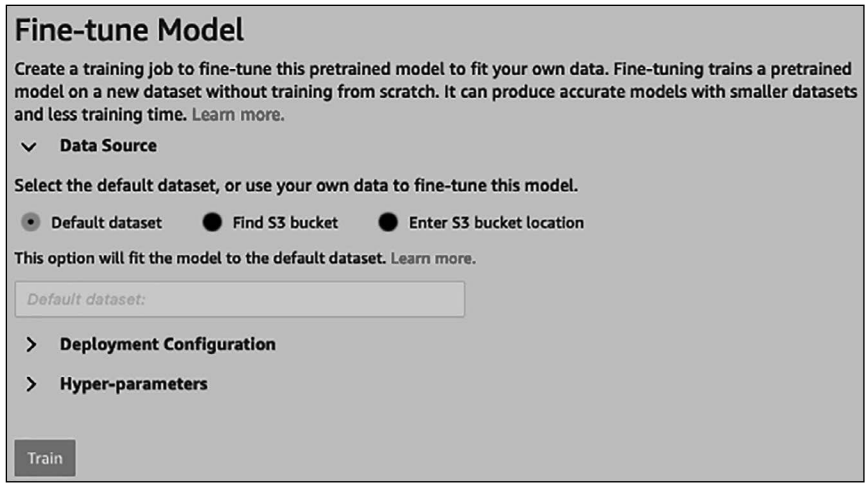

Figura 1.22 Ottimizzazione di un modello.

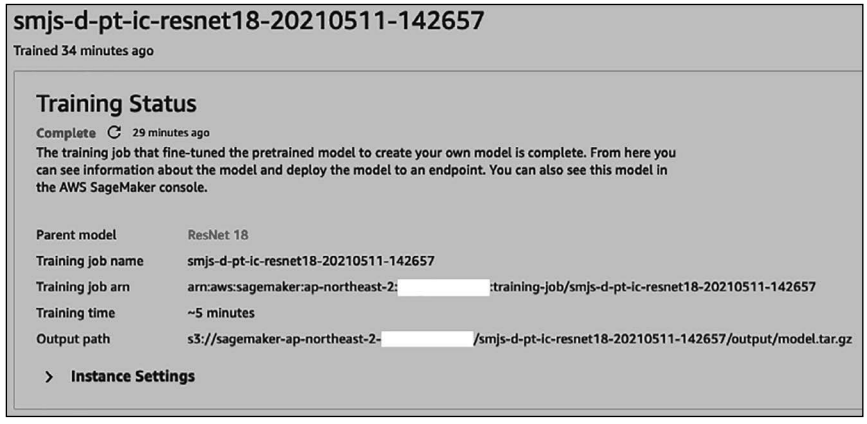

Figura 1.23 Visualizzazione dei risultati dell'ottimizzazione.

7. Innanzitutto, aggiungete una cella per copiare il contenuto del modello ottimizzato da S3 ed estraete l'elenco di classi e indici aggiunti da JumpStart:

```
%%sh
aws s3 cp s3://sagemaker-REGION_NAME-123456789012/
    smjs-d-pt-ic-resnet18-20210511-142657/output/model.tar.gz .
tar xfz model.tar.gz
cat class_label_to_prediction_index.json
{"daisy": 0, "dandelion": 1, "roses": 2, "sunflowers": 3, "tulips": 4}
```

8. Come previsto, il modello ottimizzato è in grado di predire cinque classi. Aggiungete una cella per scaricare un'immagine di girasole da Wikipedia:

```
%%sh
wget https://upload.wikimedia.org/wikipedia/commons/a/a9/A_sunflower.jpg
```

9. Ora caricate l'immagine e richiamate l'endpoint:

```
import boto3
endpoint_name = 'jumpstart-ftd-pt-ic-resnet18'
client = boto3.client('runtime.sagemaker')
with open('A_sunflower.jpg', 'rb') as file:
    image = file.read()
response = client.invoke_endpoint(
    EndpointName=endpoint_name,
    ContentType='application/x-image',
    Body=image
)
```

10. Infine, mostrate le predizioni. La probabilità più alta è relativa alla classe 3, al 60,67%, a conferma del fatto che l'immagine contiene un girasole.

```
import json
model_predictions = json.loads(response['Body'].read())
print(model_predictions)
[0.30362239480018616, 0.06462913751602173, 0.007234351709485054,
    0.6067869663238525, 0.017727158963680267]
```

11. Quando avrete terminato con le attività di test, assicuratevi di eliminare l'endpoint per evitare addebiti inutili.

Questo esempio illustra quanto sia facile ottimizzare modelli con pre-training sui propri dataset con SageMaker JumpStart e utilizzarli per eseguire predizioni sui dati. Questo è un ottimo modo per sperimentare l'impiego dei diversi modelli e per scoprire quale potrebbe funzionare meglio sullo specifico problema che state cercando di risolvere.

Riepilogo

In questo capitolo, avete esplorato le principali funzionalità di Amazon SageMaker e avete scoperto come possono aiutarvi a risolvere i vostri problemi di machine learning. Fornendovi un'infrastruttura gestita e strumenti preinstallati, SageMaker vi permette di concentrarvi solo sul problema stesso. Pertanto, potrete passare più rapidamente dalla sperimentazione dei modelli alla loro distribuzione in produzione.

Avete imparato a configurare Amazon SageMaker sul vostro computer locale e in Amazon SageMaker Studio. Quest'ultimo è un IDE di machine learning gestito, in cui molte altre funzionalità di SageMaker sono a portata di clic.

Infine, avete scoperto che cos'è Amazon SageMaker JumpStart, una raccolta di soluzioni e modelli di machine learning all'avanguardia che potete distribuire con un clic e iniziare a sottoporre a test in pochi minuti.

Nel prossimo capitolo impareremo a utilizzare Amazon SageMaker e altri servizi di AWS per preparare i vostri dataset per il training.