

How to use Intel® DAAL kNN via SageMaker web interface

Description of algorithm:

k-Nearest Neighbors (kNN) classification is a non-parametric classification algorithm. The model of the kNN classifier is based on feature vectors and class labels from the training data set. This classifier induces the class of the query vector from the labels of the feature vectors in the training data set to which the query vector is similar. A similarity between feature vectors is determined by the type of distance (for example, Euclidian) in a multidimensional feature space.

[Intel® DAAL developer guide](#)

[Intel® DAAL documentation for kNN](#)

Instruction:

1. Visit page on SageMaker Marketplace and click "Continue to Subscribe"

aws marketplace

Categories ▾ Delivery Methods ▾ Solutions ▾ Migration Mapping Assistant Your Saved List Partners Sell in AWS Marketplace Amazon Web Services Home Help

intel **Intel® DAAL k-Nearest Neighbors (kNN)**
By: Intel® AI Latest Version: 0.1 ★★★★★ (0)
Train and Deploy k-Nearest Neighbors (kNN) Classifier model with Intel® Data Analytics Acceleration Library (DAAL)

Continue to Subscribe
Save to List

Overview Pricing Usage Support Reviews

Product Overview

k-Nearest Neighbors (kNN) classification is a non-parametric classification algorithm. The model of the kNN classifier is based on feature vectors and class labels from the training data set. This classifier induces the class of the query vector from the labels of the feature vectors in the training data set to which the query vector is similar. A similarity between feature vectors is determined by the type of distance (for example, Euclidian) in a multidimensional feature space. For more info:
<https://intelpython.github.io/daal4py/algorithms.html#k-nearest-neighbors-knn>

Highlights

- Intel® DAAL Machine Learning algorithms are lightning fast and built for performance !
- Intel® DAAL uses optimized algorithms from the Intel® Math Kernel Library and Intel® Integrated Performance Primitives.

Key Data	
Version	0.1 Usage Instructions Release Notes

2. Click "Accept Offer" if you agree with EULA at end of page. If you already subscribed on algorithm on Marketplace this step will be skipped.

The screenshot shows the AWS Marketplace interface for the Intel DAAL k-Nearest Neighbors (kNN) software. The top navigation bar includes the AWS Marketplace logo, a search bar, and links to Categories, Delivery Methods, Solutions, Migration Mapping Assistant, Your Saved List, Partners, Sell in AWS Marketplace, Amazon Web Services Home, and Help. The Intel logo is prominently displayed on the left, and the product name 'Intel® DAAL k-Nearest Neighbors (kNN)' is centered. A 'Continue to configuration' button is on the right. Below the navigation bar, the page title 'Subscribe to this software' is shown. A paragraph of text explains the subscription process and mentions the End User License Agreement (EULA). The 'Intel® AI Offer' section is highlighted, and a red arrow points to the 'Accept Offer' button. Below this, the 'Pricing Terms' section is visible, containing a table of algorithm usage prices.

Intel® AI Offer
By: Intel® AI

Accept Offer

Pricing Terms

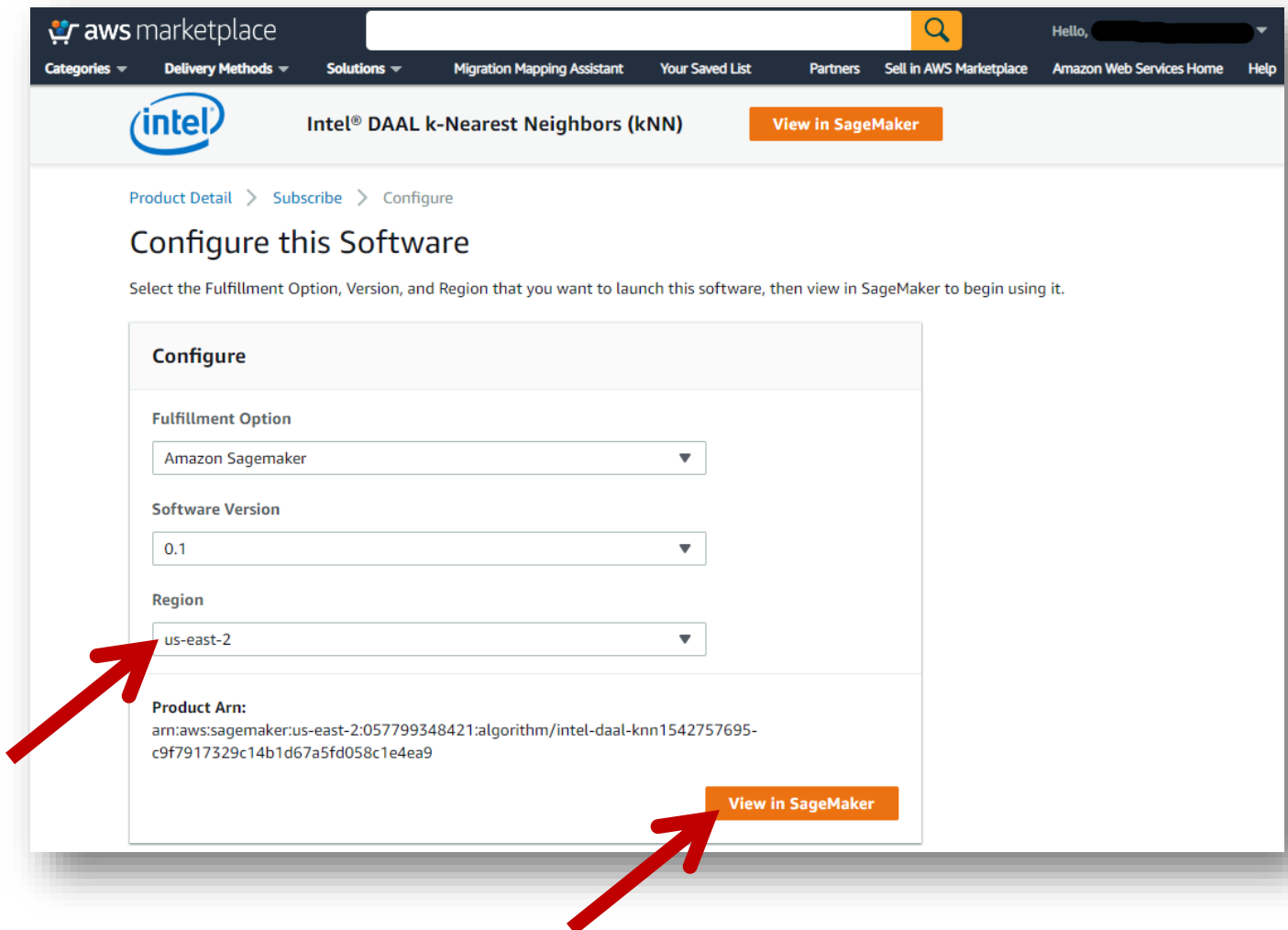
Instance type	Algorithm/hr
mL.m4.xlarge	\$0.00
mL.m4.2xlarge	\$0.00
mL.m4.4xlarge	\$0.00
mL.m4.10xlarge	\$0.00
mL.m4.16xlarge	\$0.00
mL.m5.large	\$0.00

3. Click "Continue to configuration"

The screenshot shows the AWS Marketplace interface for the Intel® DAAL k-Nearest Neighbors (kNN) software. The top navigation bar includes the AWS Marketplace logo, a search bar, and links for Categories, Delivery Methods, Solutions, Migration Mapping Assistant, Your Saved List, Partners, Sell in AWS Marketplace, Amazon Web Services Home, and Help. The main header features the Intel logo and the product name "Intel® DAAL k-Nearest Neighbors (kNN)". A red arrow points to the "Continue to configuration" button. Below the header, the page title is "Subscribe to this software". A paragraph of text explains the subscription process and mentions the End User License Agreement (EULA). Below this, there is a section titled "Intel® AI Offer" with a sub-header "By: Intel® AI". To the right of this section is a button labeled "Already Subscribed". The "Pricing Terms" section contains a table titled "Algorithm usage" with two columns: "Instance type" and "Algorithm/hr". The table lists several instance types and their corresponding prices, all of which are \$0.00.

Instance type	Algorithm/hr
ml.m4.xlarge	\$0.00
ml.m4.2xlarge	\$0.00
ml.m4.4xlarge	\$0.00
ml.m4.10xlarge	\$0.00
ml.m4.16xlarge	\$0.00
ml.m5.large	\$0.00

4. Choose the Region and click “View in SageMaker”



kNN algorithm on AWS SageMaker is divided into two stages: training job and getting inference from endpoint.

Training job is computing tree and other values from provided training data.

After that, you should create model with computed values and endpoint based on it.

Sending data to endpoint gives you predictions in response

5. Select needed algorithm version and click “Create training job”

The screenshot shows the Amazon SageMaker console interface. The left sidebar contains navigation links: Dashboard, Search^{Beta}, Ground Truth (Labeling jobs, Labeling datasets, Labeling workforces), Notebook (Notebook instances), Training (Algorithms, Training jobs, Hyperparameter tuning jobs), and Inference (Compilation jobs, Model packages, Models). The main content area is titled 'Intel® DAAL k-Nearest Neighbors (kNN)' and includes a breadcrumb trail: Amazon SageMaker > Algorithms: My Subscriptions > View subscription details. Two buttons are visible: 'Create hyperparameter tuning job' and 'Create training job'. A red arrow points to the 'Create training job' button. Below the buttons is a table titled 'Algorithm versions' with columns: Title, Version, and Algorithm ARN. The table contains one entry: Intel® DAAL k-Nearest Neighbors (kNN) with version 0.1 and ARN arn:aws:sagemaker:us-east-2:057799348421:algorithm/intel-daal-knn1542757695-c9f7917329c14b1d67a5fd058c1e4ea9.

Title	Version	Algorithm ARN
Intel® DAAL k-Nearest Neighbors (kNN)	0.1	arn:aws:sagemaker:us-east-2:057799348421:algorithm/intel-daal-knn1542757695-c9f7917329c14b1d67a5fd058c1e4ea9

6. Type job name, select IAM role and instance type

The screenshot shows the 'Job settings' form in the Amazon SageMaker console. The left sidebar is the same as in the previous screenshot. The main content area is titled 'Job settings' and contains several sections: 'Job name' with a text input field containing 'daal-knn-sm-test' and a description; 'IAM role' with a dropdown menu showing 'AmazonSageMaker-ExecutionRole-'; 'Algorithm options' with a section for 'Algorithm source' and a 'Choose an algorithm subscription' button; and 'Resource configuration' with fields for 'Instance type' (ml.m4.xlarge), 'Instance count' (1), and 'Additional storage volume per instance (GB)' (1). Three red arrows point to the 'Job name' input field, the 'IAM role' dropdown, and the 'Instance type' dropdown respectively.

7. Choose hyperparameters

Hyperparameters

You can use hyperparameters to finely control training. We've set default hyperparameters for the algorithm you've chosen.

Key	Value
nClasses	5
fptype	double
method	defaultDense
k	1
dataUseInModel	doNotUse
seed	777
distributed	False

Parameter name	Type	Default value	Description
nClasses	int	2	Number of classes in data
fptype	str	"double"	The floating-point type that the algorithm uses for intermediate computations. Can be "float" or "double"
method	str	"defaultDense"	The computation method used by the K-D tree based kNN classification. The only training method supported so far is the default dense method.
k	int	1	The number of neighbors
dataUseInModel	str	"doNotUse"	A parameter to enable/disable use of the input data set in the kNN model. Possible values: "doNotUse" - the algorithm does not include the input data and labels in the trained kNN model but creates a copy of the input data set "doUse" - the algorithm includes the input data and labels in the trained kNN model
seed	int	777	Seed for random number generator engine that is used internally to perform sampling needed to choose dimensions and cut-points for the K-D tree.

8. Specify S3 location of input data for training

Input data configuration

Create up to 8 channels of input sources. If the algorithm you chose supports multiple input channels, you can specify those here. See [Algorithms Provided by Amazon SageMaker: Common Parameters](#)

Channels

▼ training

Remove

Channel name

training

Input mode - optional

File

Content type - optional

text/csv

Choose one of the formats below

- text/csv

Compression type

None

Record wrapper

None

S3 data type

S3Prefix

S3 data distribution type

FullyReplicated

S3 location

s3://daal-knn-test/inputdata

Add channel

9. Specify S3 output path (model will be stored here) and click “Create training job”

Output data configuration

S3 output path

s3://bucket/path-to-your-data/

Encryption key - *optional*

If you want Amazon SageMaker to encrypt the output of your training job using your own AWS KMS encryption key instead of the default S3 service key, provide its ID or ARN.

▼ Tags - *optional*

Key

Value

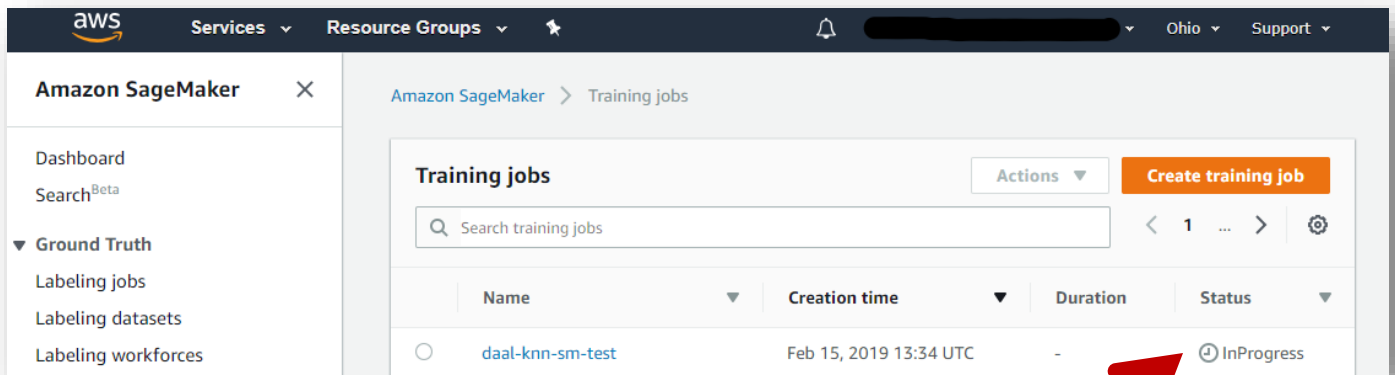
Remove

[Add tag](#)

Cancel

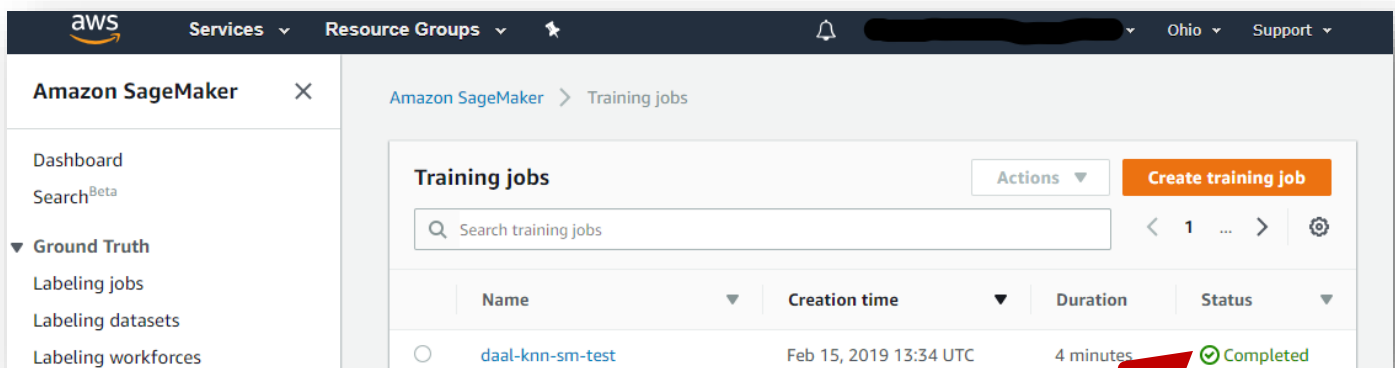
Create training job

10. Wait until finish of training job



The screenshot shows the Amazon SageMaker console with the 'Training jobs' page. A table lists the training job 'daal-knn-sm-test' with a status of 'InProgress'. A red arrow points to the 'InProgress' status.

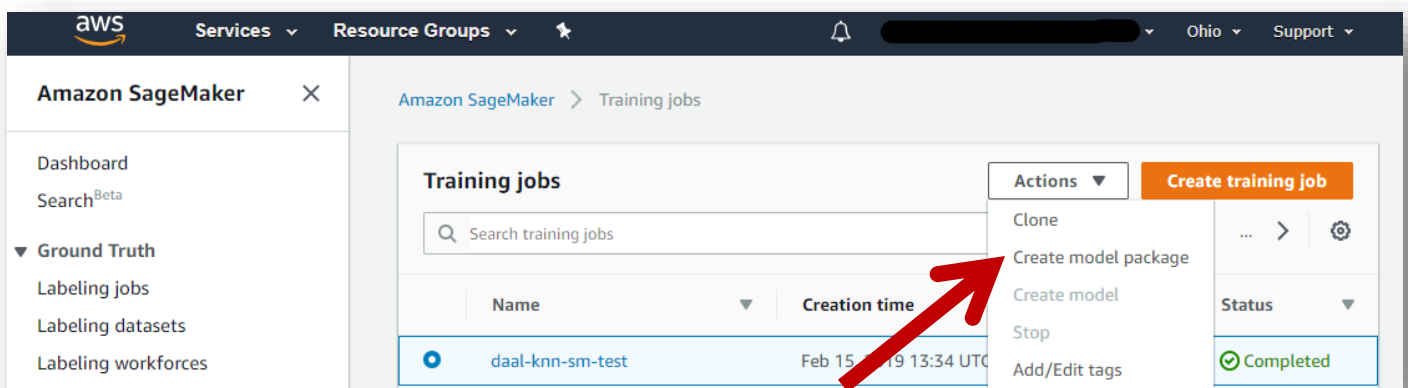
Name	Creation time	Duration	Status
daal-knn-sm-test	Feb 15, 2019 13:34 UTC	-	InProgress



The screenshot shows the Amazon SageMaker console with the 'Training jobs' page. The training job 'daal-knn-sm-test' now has a status of 'Completed'. A red arrow points to the 'Completed' status.

Name	Creation time	Duration	Status
daal-knn-sm-test	Feb 15, 2019 13:34 UTC	4 minutes	Completed

11. Select training job and take action "Create model package"



The screenshot shows the Amazon SageMaker console with the 'Training jobs' page. The training job 'daal-knn-sm-test' is selected, and the 'Actions' dropdown menu is open, showing the option 'Create model package'. A red arrow points to the 'Create model package' option.

Name	Creation time	Status
daal-knn-sm-test	Feb 15, 2019 13:34 UTC	Completed

12. Type model package name and click "Next"

Create model package

Inference specifications

Model package name and description

Model package name

The model package name must be unique in your account and in the AWS Region and can have up to 63 characters. Valid characters: a-z, A-Z, 0-9, and - (hyphen)

Description - *optional*

The description can be up to 1024 characters.

Inference specification options

- ☐ Provide the location of the inference image and model artifacts
Choose this option if your model was trained using an algorithm stored in ECR.
- ☒ Provide the algorithm used for training and its model artifacts
Choose this option if you are using a model trained by an algorithm resource or subscription algorithm from AWS Marketplace.

Algorithm and model artifacts

Algorithm ARN

Enter the Amazon Resource Name (ARN) used to create the training job and model artifacts.

Location of model artifacts - *optional*

If you want buyers to use the model artifacts from a specific model, enter the path to the S3 bucket where they are stored.

To find a path, [go to Amazon S3](#)

Cancel **Next**

13. Click "Create model package"

Step 1
Inference specifications

Step 2
Validation specifications

Create model package

Validation specifications

To list your model package on AWS Marketplace, you must have it validated by Amazon SageMaker. Provide the information that Amazon SageMaker needs to run transform jobs to validate your product.

Validation and scanning

Publish this model package on AWS Marketplace
AWS SageMaker requires successful validation before you can publish to AWS Marketplace.

☐ Yes ☐ No

Validate this resource
AWS SageMaker will create a training job and/or transform job based on your validation profiles below.

☐ Yes ☒ No

Cancel Previous **Create model package**

14. Wait until package is created

Amazon SageMaker

Dashboard
Search^{Beta}

▼ Ground Truth
Labeling jobs
Labeling datasets
Labeling workforces

▼ Notebook
Notebook instances
Lifecycle configurations
Git repositories

▼ Training

New model package created successfully.

Amazon SageMaker > My model packages

My model packages | AWS Marketplace subscriptions

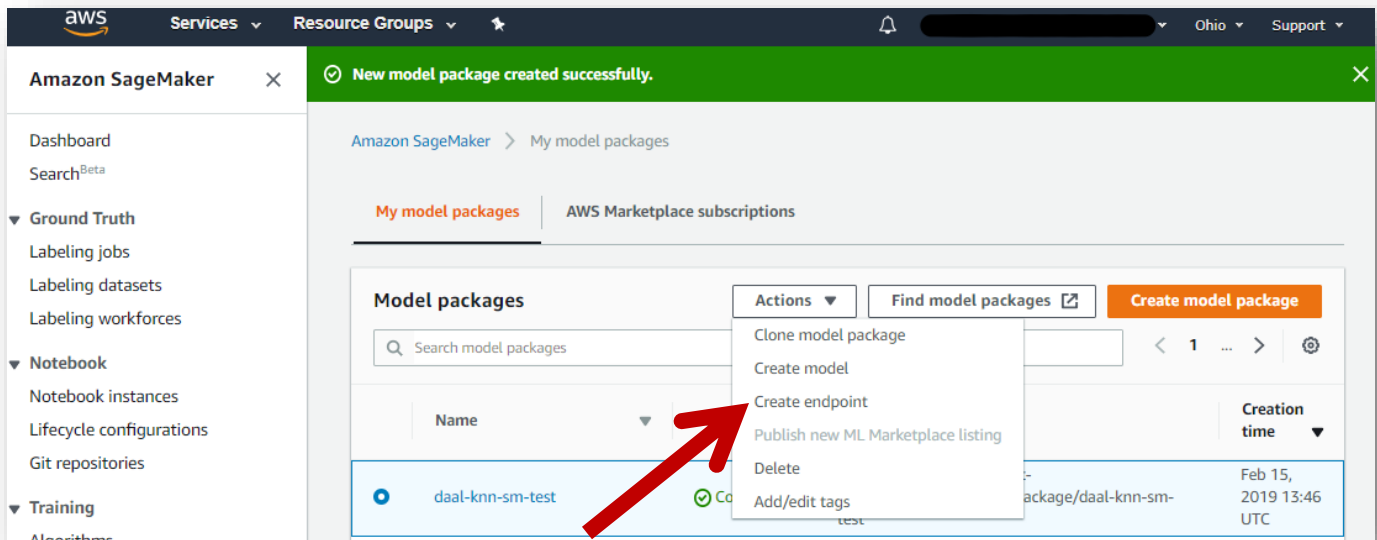
Model packages

Actions Find model packages Create model package

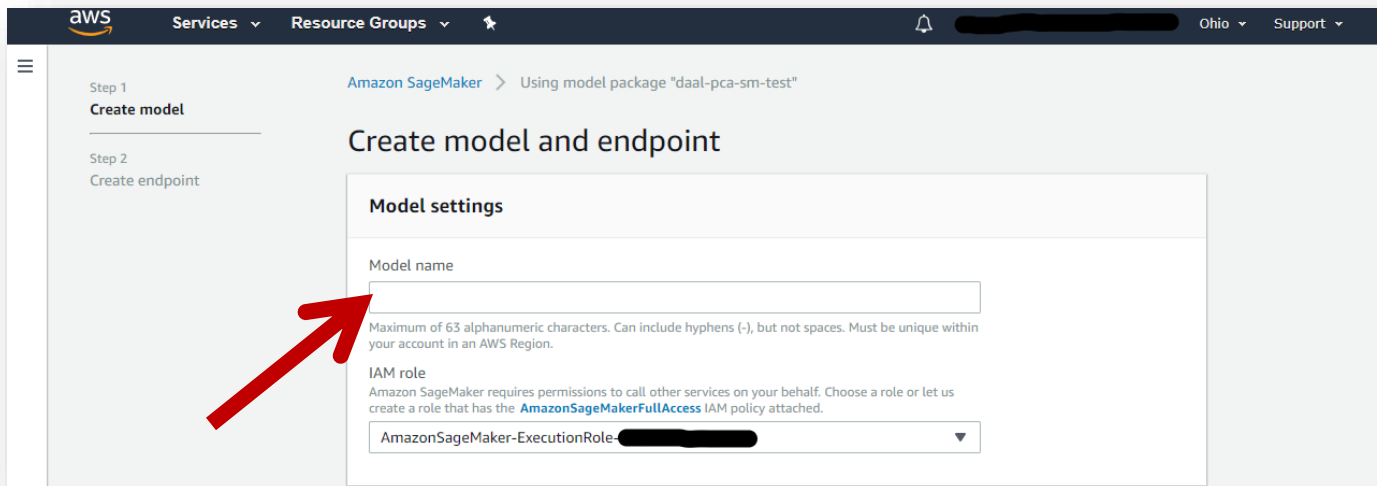
Search model packages

	Name	Status	ARN	Creation time
<input type="radio"/>	daal-knn-sm-test	Completed	arn:aws:sagemaker:us-east-2:[redacted]:model-package/daal-knn-sm-test	Feb 15, 2019 13:46 UTC

15. Select package and take action "Create endpoint"



16. Type model name and click "Next"



17. Type endpoint name, edit and create endpoint configuration and click "Submit"

aws Services Resource Groups Ohio Support

Model was successfully updated. Continue to create an endpoint.

Step 1 Create model

Step 2 Create endpoint

Amazon SageMaker > Using model package "daal-pca-sm-test"

Create model and endpoint

Endpoint

Endpoint name
Your application uses this name to access this endpoint.

Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.

Attach endpoint configuration

☐ Use an existing endpoint configuration
Use an existing endpoint configuration or clone an endpoint configuration.

☒ Create a new endpoint configuration
Add models and configure the instance and initial weight for each model.

New endpoint configuration

To deploy models to Amazon SageMaker, first create an endpoint configuration. In the configuration, specify which models to deploy, and the relative traffic weighting and hardware requirements for each.

Endpoint configuration name

endpointConfig-██████████

Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.

Encryption key - optional
Encrypt your data. Choose an existing KMS key or enter a key's ARN.

No Custom Encryption

Production variants

Model name	Training job	Variant name	Instance type	Elastic Inference	Initial instance count	Initial weight	Actions
daal-knn-sm-test	U	variant-name-1	mL.m4.xlarge	none	1	1	Edit Remove

Add model

Create endpoint configuration

Tags - optional

Key	Value	Remove
		Remove

Add tag

Cancel Previous Submit

18. Wait until endpoint is ready

Amazon SageMaker

Success! You created an endpoint.
To track the status of the endpoint, view details.

Amazon SageMaker > Endpoints

Endpoints

Update endpoint Actions Create endpoint

Search endpoints

	Name	ARN	Creation time	Status	Last updated
<input type="radio"/>	daal-knn-sm-test	arn:aws:sagemaker:us-east-2:██████████:endpoint/daal-knn-sm-test	Feb 15, 2019 14:05 UTC	Creating	Feb 15, 2019 14:05 UTC

Existing notebooks

Notebook

<input type="radio"/>	daal-knn-sm-test	arn:aws:sagemaker:us-east-2:██████████:endpoint/daal-knn-sm-test	Feb 15, 2019 14:05 UTC	InService	Feb 15, 2019 14:11 UTC
-----------------------	------------------	------------------------------------------------------------------	------------------------	-----------	------------------------

19. Use AWS CLI to get real-time prediction

Type command:

```
aws sagemaker-runtime invoke-endpoint --endpoint-name <endpoint-name> --body "$(cat <prediction_data_file_name>)" --content-type text/csv --accept text/csv <output_data_file_name>
```

```
(base) ubuntu@ip-172-31-22-46:~$ aws sagemaker-runtime invoke-endpoint --endpoint-name daal-knn-sm-test --body "$(cat probe_data.csv)" --content-type text/csv --accept text/csv output.txt
```

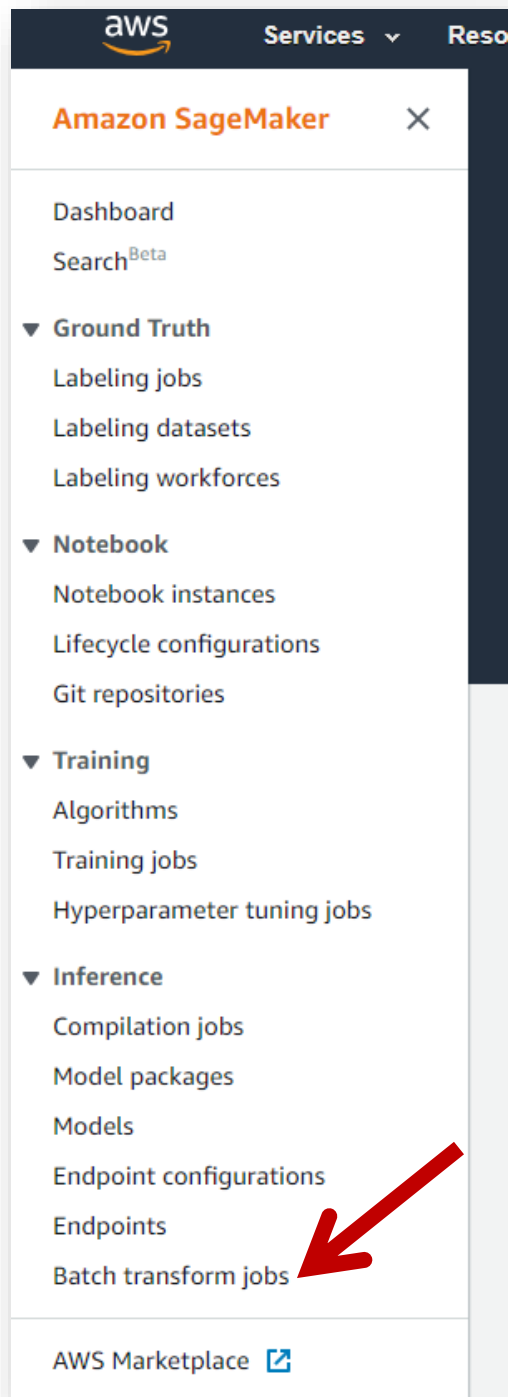
Then, see content of output file:

```
(base) ubuntu@ip-172-31-22-46:~$ cat output.txt
1
4
0
2
3
4
1
1
0
2
4
```

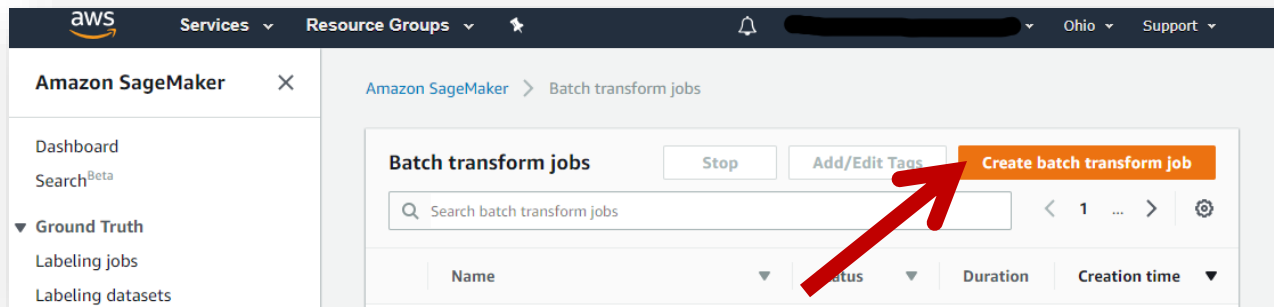
Batch transform job as alternative to endpoint

You can use batch transform job if you need compute predictions once.

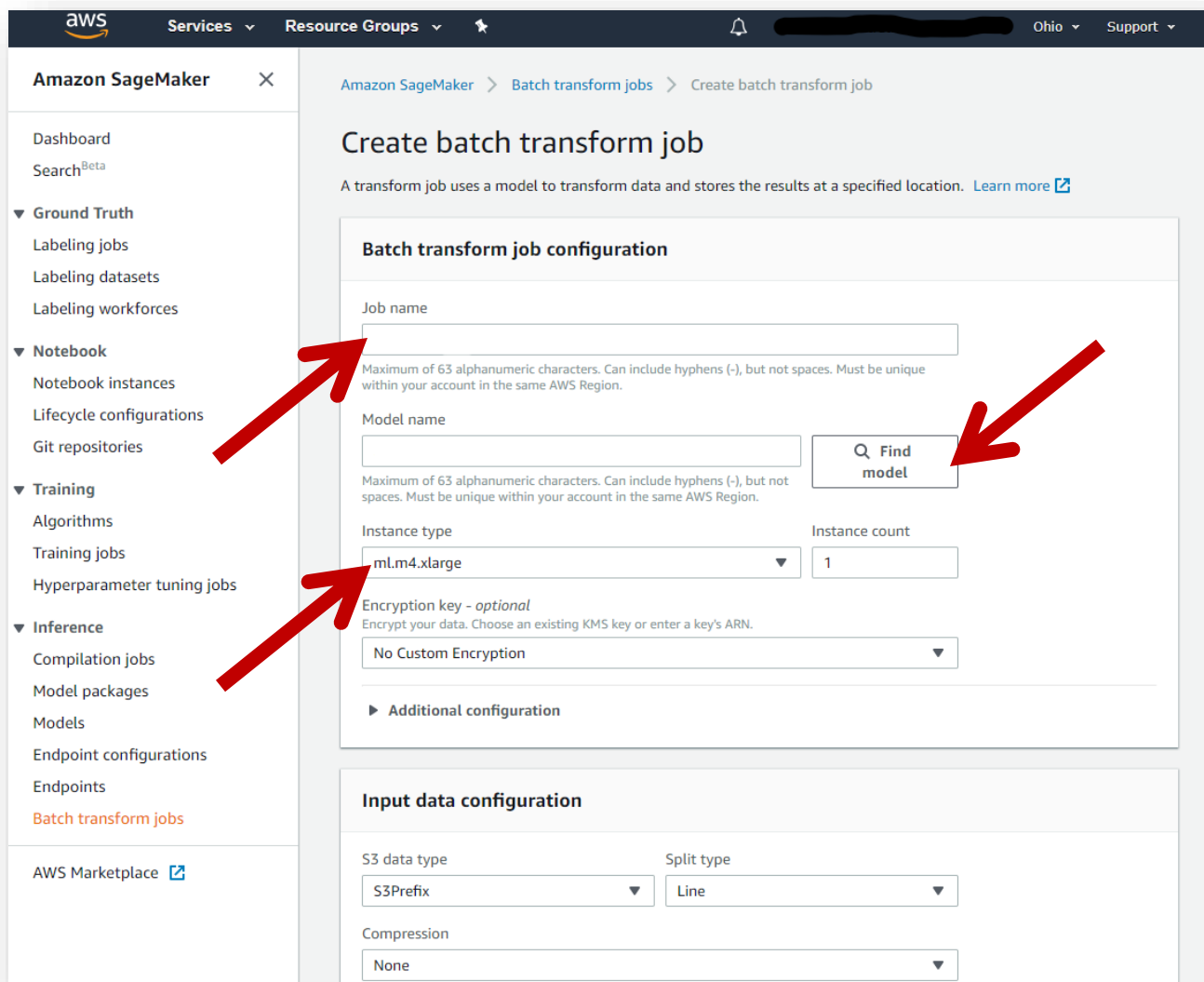
1. Go to “Batch transform job” page



2. Click "Create batch transform job"



3. Enter job name, select previously created model, instance and set instance count to 1



- Specify S3 location of data for prediction, S3 output path (predictions will be stored here) and click “Create job”

The screenshot displays the Amazon SageMaker console interface. On the left, a navigation sidebar lists various services under categories like Ground Truth, Notebook, Training, and Inference. The main panel is titled 'Input data configuration' and contains several dropdown menus: 'S3 data type' (set to 'S3Prefix'), 'Split type' (set to 'Line'), 'Compression' (set to 'None'), and 'Content type - optional' (set to 'text/csv'). Below these is a text input field for 'S3 location', which is highlighted by a red arrow. A link 'To find a path, go to Amazon S3' is provided. The 'Output data configuration' section follows, featuring an 'S3 output path' text input field, also highlighted by a red arrow, with a similar 'To find a path, go to Amazon S3' link. Other options include 'Encryption key - optional' (set to 'No Custom Encryption'), 'Accept - optional' (set to 'text/csv'), and 'Assemble with' (set to 'Line'). At the bottom, there is a 'Tags - optional' section. The bottom right corner contains 'Cancel' and 'Create job' buttons, with the 'Create job' button highlighted by a red arrow.

Amazon SageMaker

Input data configuration

S3 data type: S3Prefix

Split type: Line

Compression: None

Content type - optional: text/csv

S3 location:

To find a path, [go to Amazon S3](#)

Output data configuration

S3 output path:

To find a path, [go to Amazon S3](#)

Encryption key - optional: No Custom Encryption

Accept - optional: text/csv

Assemble with: Line

Tags - optional

Cancel Create job

5. Wait until job is completed and find predictions in previously specified S3 output path

The screenshot shows the Amazon SageMaker console with a green notification banner at the top stating "Batch transform job daal-sm-test was successfully created". The left sidebar contains navigation links for Dashboard, Search^{Beta}, Ground Truth, Labeling jobs, Labeling datasets, Labeling workforces, Notebook, and Notebook instances. The main content area displays "Batch transform jobs" with a search bar and a table. The table has columns for Name, Status, Duration, and Creation time. A single job, "daal-sm-test", is listed with a status of "InProgress", a duration of "a few seconds", and a creation time of "Feb 17, 2019 14:32 UTC". A red arrow points to the "InProgress" status.

Name	Status	Duration	Creation time
daal-sm-test	InProgress	a few seconds	Feb 17, 2019 14:32 UTC

The screenshot shows the same Amazon SageMaker console interface, but the status of the "daal-sm-test" job has changed to "Completed". The duration is now "3 minutes" and the creation time remains "Feb 17, 2019 14:32 UTC". A red arrow points to the "Completed" status.

Name	Status	Duration	Creation time
daal-sm-test	Completed	3 minutes	Feb 17, 2019 14:32 UTC