

# How to use Intel® DAAL Decision Forest via SageMaker web interface

## Description of algorithm:

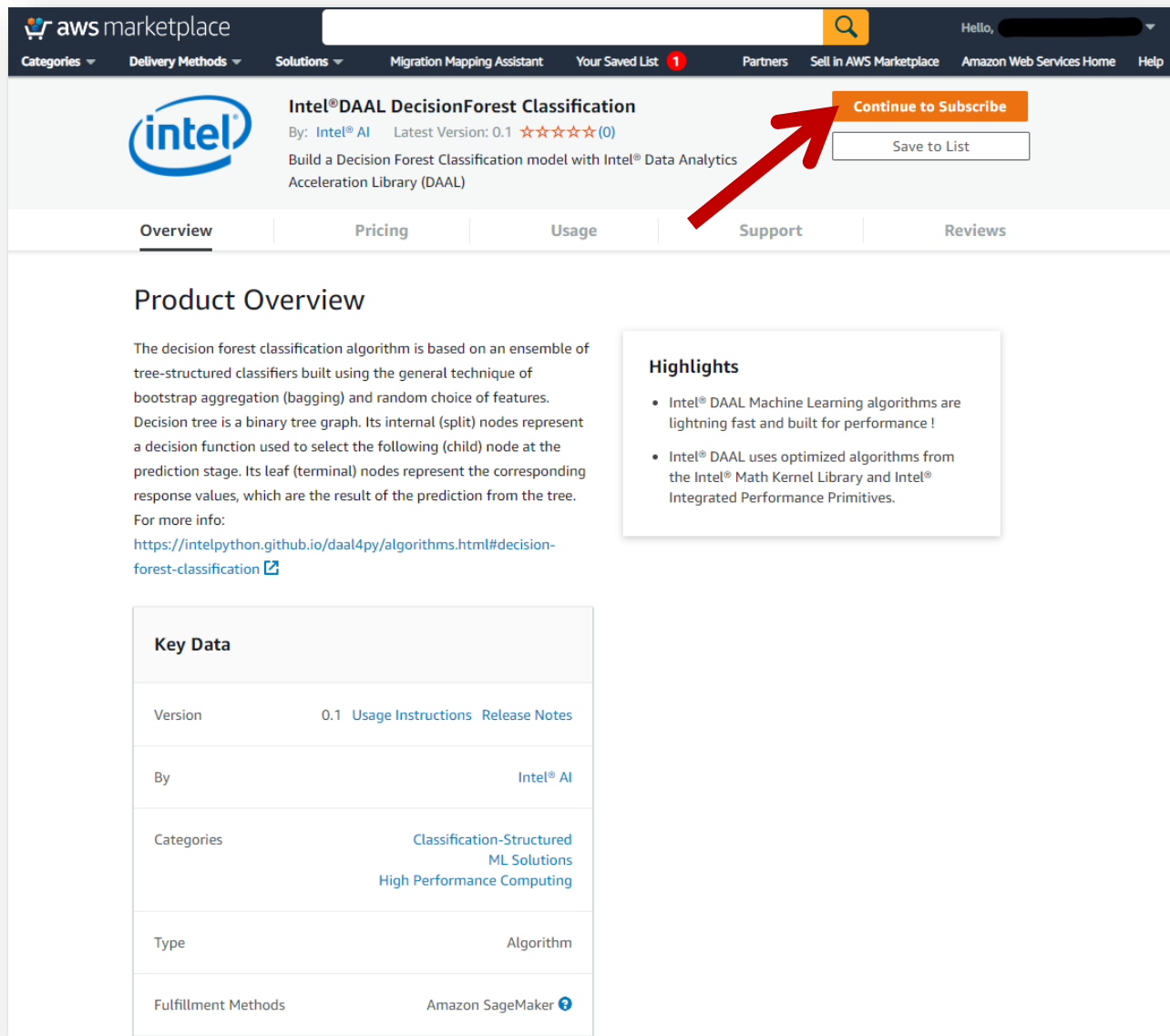
Decision forest classification and regression algorithms based on an ensemble of tree-structured classifiers (decision trees) built using the general technique of bootstrap aggregation (bagging) and random choice of features. Decision tree is a binary tree graph. Its internal (split) nodes represent a decision function used to select the following (child) node at the prediction stage. Its leaf (terminal) nodes represent the corresponding response values, which are the result of the prediction from the tree.

[Intel® DAAL developer guide](#)

[Intel® DAAL documentation for Decision Forest](#)

## Instruction:

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



The screenshot shows the AWS Marketplace page for the Intel® DAAL DecisionForest Classification algorithm. The page header includes the AWS Marketplace logo, a search bar, and navigation links. The main content area features the Intel logo, the product name, and a 'Continue to Subscribe' button highlighted by a red arrow. Below the product name, there is a 'Product Overview' section with a description of the algorithm and a 'Key Data' table.

**Intel® DAAL DecisionForest Classification**  
By: Intel® AI Latest Version: 0.1 ★★★★★ (0)  
Build a Decision Forest Classification model with Intel® Data Analytics Acceleration Library (DAAL)

**Continue to Subscribe**  
Save to List

**Product Overview**

The decision forest classification algorithm is based on an ensemble of tree-structured classifiers built using the general technique of bootstrap aggregation (bagging) and random choice of features. Decision tree is a binary tree graph. Its internal (split) nodes represent a decision function used to select the following (child) node at the prediction stage. Its leaf (terminal) nodes represent the corresponding response values, which are the result of the prediction from the tree.

For more info:  
<https://intelpython.github.io/daal4py/algorithms.html#decision-forest-classification>

**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 <a href="#">Usage Instructions</a> <a href="#">Release Notes</a>
By	Intel® AI
Categories	<a href="#">Classification-Structured ML Solutions</a> <a href="#">High Performance Computing</a>
Type	Algorithm
Fulfillment Methods	Amazon SageMaker <a href="#">?</a>

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 DecisionForest Classification algorithm. The top navigation bar includes the AWS Marketplace logo, a search bar, and user account information. The main header features the Intel logo and the product name, with a 'Continue to configuration' button. Below the header, the page is titled 'Subscribe to this software'. A paragraph explains the subscription process and mentions the EULA. A red arrow points to the 'Accept Offer' button. Below this, the 'Pricing Terms' section is visible, showing a version dropdown set to '0.1' and a table of algorithm usage pricing.

**Intel® AI Offer**  
By: Intel® AI

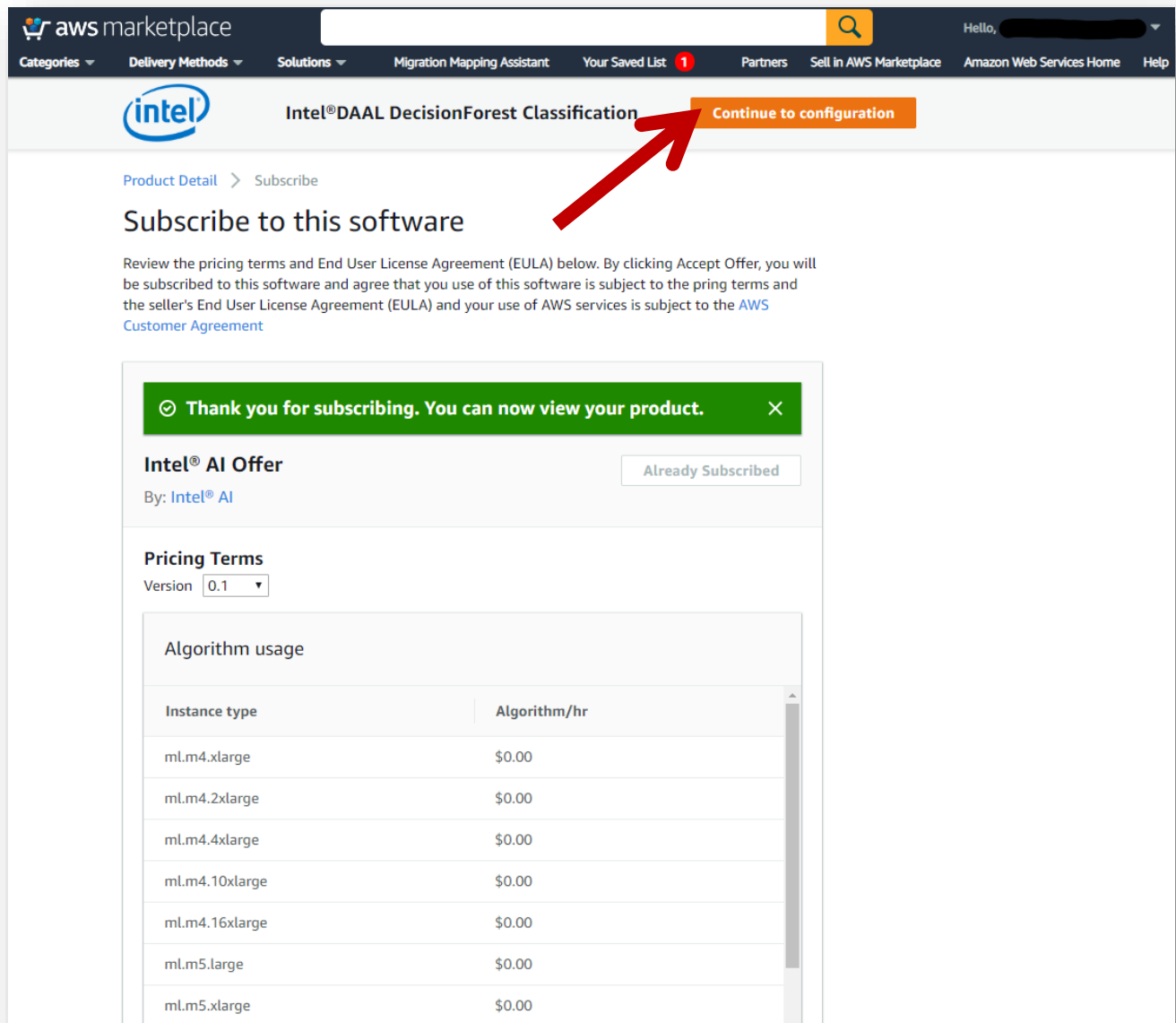
**Accept Offer**

**Pricing Terms**  
Version: 0.1

**Algorithm usage**

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
ml.m5.xlarge	\$0.00
ml.m5.2xlarge	\$0.00
ml.m5.4xlarge	\$0.00
ml.m5.12xlarge	\$0.00

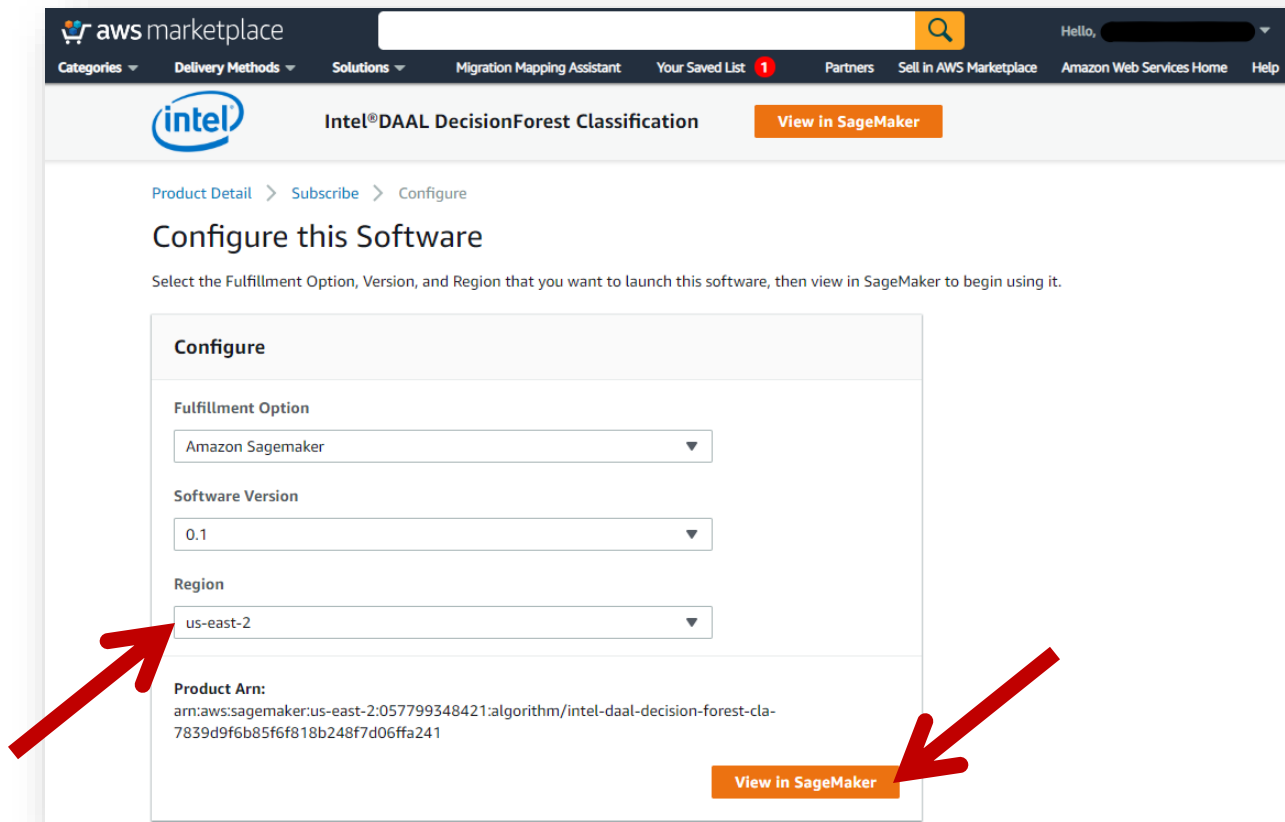
### 3. Click "Continue to configuration"



The screenshot shows the AWS Marketplace interface for the Intel® DAAL DecisionForest Classification software. The top navigation bar includes links for Categories, Delivery Methods, Solutions, Migration Mapping Assistant, Your Saved List (with a red notification icon), Partners, Sell in AWS Marketplace, Amazon Web Services Home, and Help. The main header features the Intel logo and the product name "Intel® DAAL DecisionForest Classification". A red arrow points to the "Continue to configuration" button, which is highlighted in orange. Below the header, there is a "Product Detail" link and a "Subscribe" button. The main content area is titled "Subscribe to this software" and includes a paragraph about reviewing pricing terms and the End User License Agreement (EULA). A green notification banner at the top of the main content area states: "Thank you for subscribing. You can now view your product." Below this, the "Intel® AI Offer" section shows the product is by Intel® AI and includes an "Already Subscribed" button. The "Pricing Terms" section shows the version as 0.1. A table titled "Algorithm usage" lists various instance types and their corresponding algorithm usage rates.

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
ml.m5.xlarge	\$0.00

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



Decision forest 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. On the left is a navigation sidebar with options like Dashboard, Search, Ground Truth, Notebook, and Inference. The main content area is titled 'Intel®DAAL DecisionForest Classification'. Below the title is a table of 'Algorithm versions'. The first row is selected, showing 'Intel®DAAL DecisionForest Classification' with version '0.1' and a specific ARN. To the right of the table are two buttons: 'Create hyperparameter tuning job' and 'Create training job'. A red arrow points from the 'Create training job' button towards the 'Algorithm versions' table.

Title	Version	Algorithm ARN
Intel®DAAL DecisionForest Classification	0.1	arn:aws:sagemaker:us-east-2:057799348421:algorithm/intel-daal-decision-forest-cla-7839d9f6b85f6f818b248f7d06ffa241

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

The screenshot shows the 'Job settings' configuration page in the Amazon SageMaker console. The page has several sections: 'Job name' with a text input field, 'IAM role' with a dropdown menu showing 'AmazonSageMaker-ExecutionRole', 'Algorithm options' with expandable sections for 'Algorithm source' and 'Choose an algorithm subscription', and 'Resource configuration' with fields for 'Instance type' (set to 'ml.m4.xlarge'), 'Instance count' (set to '1'), and 'Additional storage volume per instance (GB)' (set to '1'). There is also an 'Encryption key' section. Three red arrows point to the 'Job name' input field, the 'IAM role' dropdown, and the 'Instance type' dropdown.

## 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
nTrees	100
observationsPerTreeFraction	1
featuresPerNode	0
maxTreeDepth	5
minObservationsInLeafNode	1
seed	777
impurityThreshold	0
varImportance	MDA_Raw
resultsToCompute	computeOutOfBagError computeOutOfE
memorySavingMode	False
bootstrap	True

Parameter name	Type	Default value	Description
nClasses	int	2	Number of classes in data (only for classification)
fptype	str	"double"	The floating-point type that the algorithm uses for intermediate computations. Can be "float" or "double"
method	str	"defaultDense"	The only training method supported so far is the default dense method
nTrees	int	100	The number of trees in the forest
observationsPerTree Fraction	int	1	Fraction of the training set S used to form the bootstrap set for a single tree training, observationsPerTreeFraction in (0, 1]. The observations are sampled randomly with replacement
featuresPerNode	int	0	The number of features tried as possible splits per node. If the parameter is set to 0, the library uses the square root of the number of features for classification and (the number of features)/3 for regression
maxTreeDepth	int	0	Maximal tree depth. Default is 0 (unlimited).
minObservationsInLeafNode	int	1 for classification, 5 for regression	The number of neighbors
seed	int	777	The seed for random number generator, which is used to choose the bootstrap set, split features in every split node in a tree, and generate permutation required in computations of MDA variable importance
impurityThreshold	float	0	The threshold value used as stopping criteria: if the impurity value in the node is smaller than the threshold, the node is not split anymore
varImportance	str	"none"	The variable importance computation mode. Possible values: none – variable importance is not calculated MDI - Mean Decrease of Impurity, also known as the Gini importance or Mean Decrease Gini MDA_Raw - Mean Decrease of Accuracy (permutation importance) MDA_Scaled - the MDA_Raw value scaled by its standard deviation
resultsToCompute	str	"none"	Provide one of the following values to request a single characteristic or use bitwise OR to request a combination of the characteristics: computeOutOfBagError, computeOutOfBagErrorPerObservation
memorySavingMode	bool	False	If True, memory saving mode is enabled
bootstrap	bool	False for classification, True for regression	If True, bootstrap is enabled

## 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

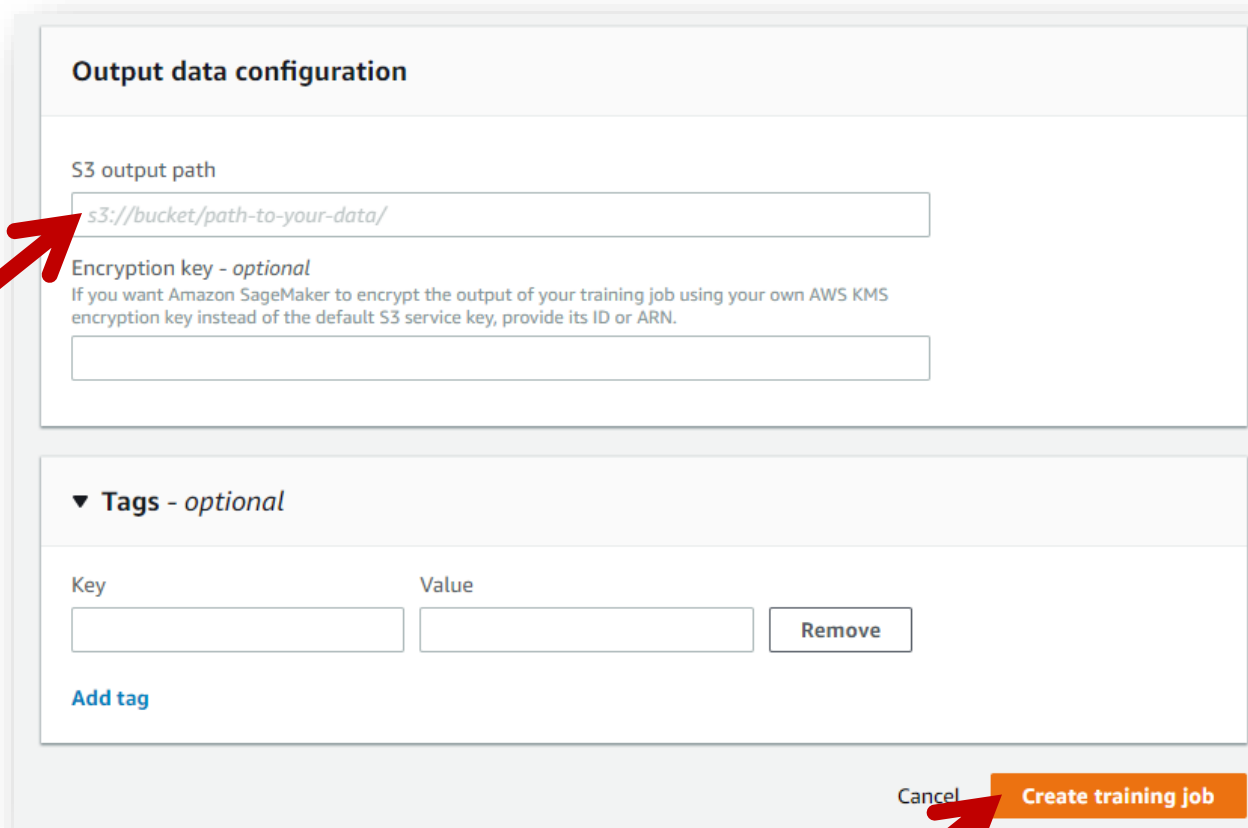
S3 location

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

Add channel



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



**Output data configuration**

S3 output path

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.

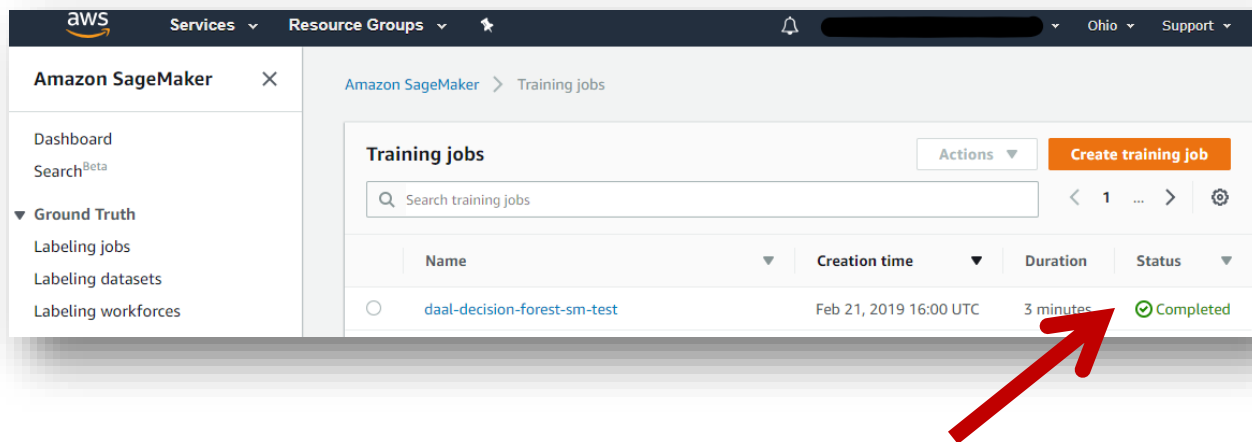
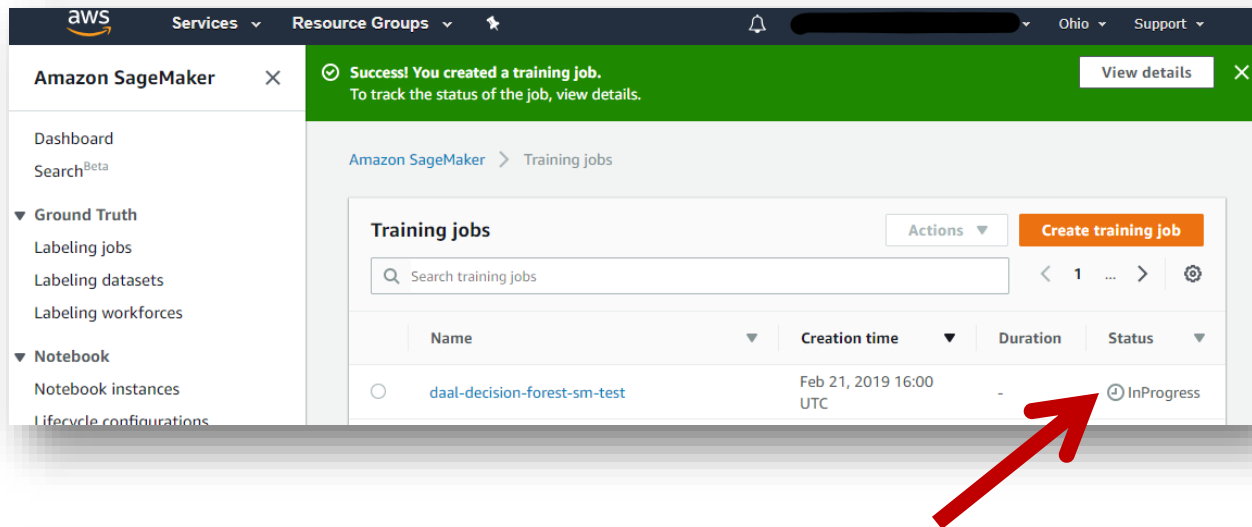
▼ **Tags - optional**

Key	Value	
<input type="text"/>	<input type="text"/>	<button>Remove</button>

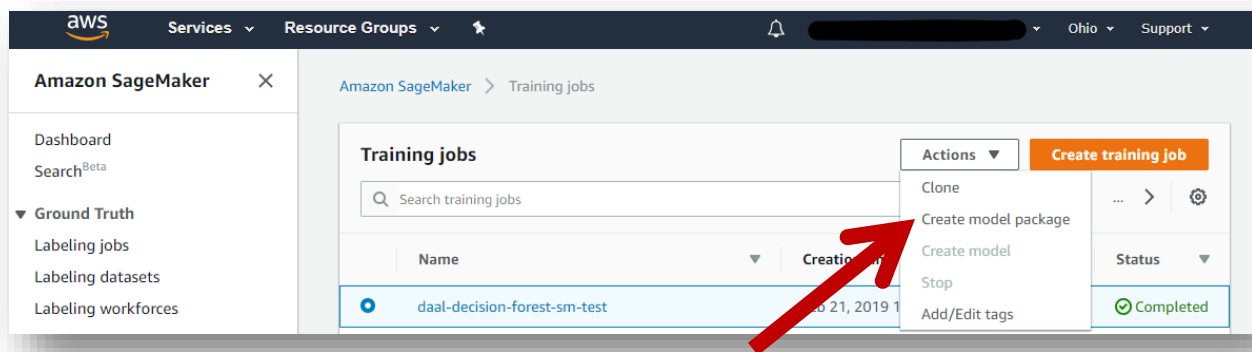
[Add tag](#)

[Cancel](#) [Create training job](#)

## 10. Wait until finish of training job



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



12. Type model package name and click “Next”

## Inference specifications

### Model package name and description

Model package name

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.

arn:aws:sagemaker:us-east-2:057799348421:algorithm/intel-daal-decision-forest-cla-

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.

s3://daal-validation/decision-forest/outputs/daal-decision-forest-sm-test/output/moc

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

### 14. Wait until package is created

Amazon SageMaker

Dashboard  
Search<sup>Beta</sup>

- Ground Truth
  - Labeling jobs
  - Labeling datasets
  - Labeling workforces
- Notebook
  - Notebook instances
  - Lifecycle configurations
  - Git repositories
- Training
  - Algorithms

New model package created successfully.

Amazon SageMaker > My model packages

My model packages | AWS Marketplace subscriptions

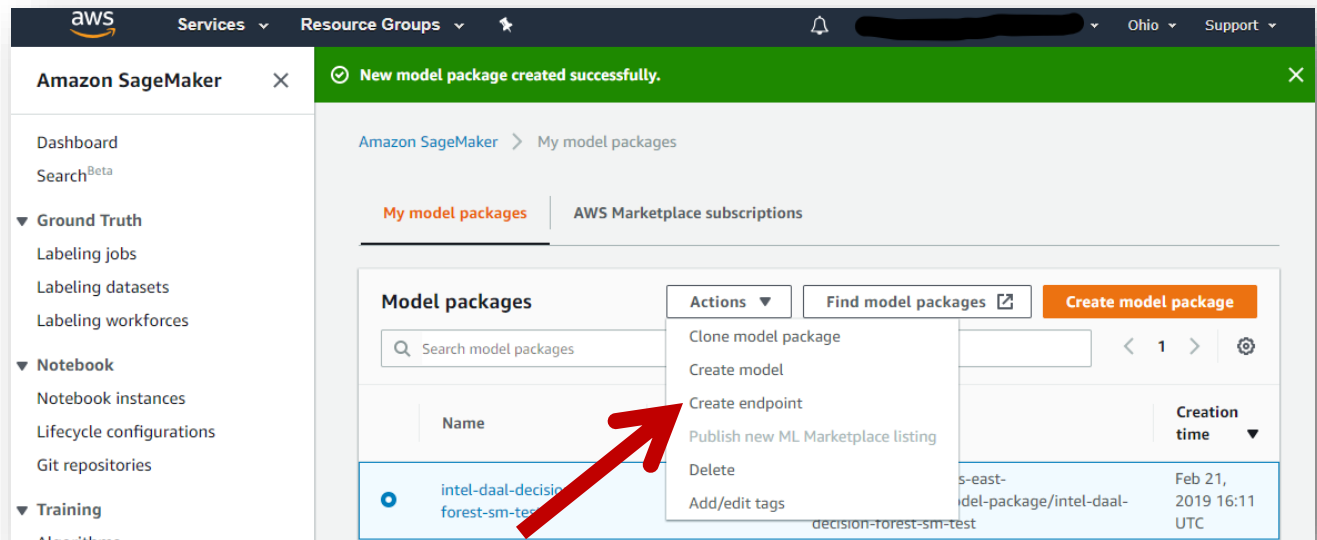
**Model packages**

	Name	Status	ARN	Creation time
<input type="radio"/>	intel-daal-decision-forest-sm-test	InProgress	arn:aws:sagemaker:us-east-2:[redacted]:model-package/intel-daal-decision-forest-sm-test	Feb 21, 2019 16:11 UTC

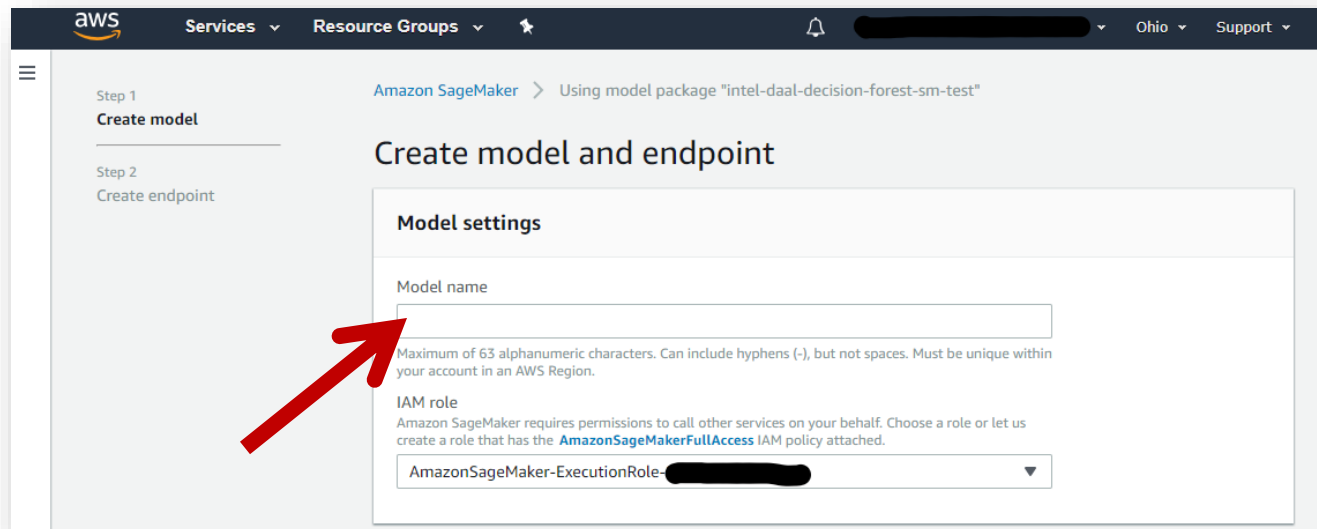
  

<input type="radio"/>	intel-daal-decision-forest-sm-test	Completed	arn:aws:sagemaker:us-east-2:[redacted]:model-package/intel-daal-decision-forest-sm-test	Feb 21, 2019 16:11 UTC
-----------------------	------------------------------------	-----------	---	------------------------

## 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”

The screenshot shows the AWS SageMaker console with a green notification bar at the top stating "Model was successfully updated. Continue to create an endpoint." The left sidebar shows "Step 1 Create model" and "Step 2 Create endpoint". The main heading is "Create model and endpoint" with a sub-header "Using model package 'intel-daal-decision-forest-sm-test'".

**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.

The screenshot shows the "New endpoint configuration" page in the AWS SageMaker console. It includes instructions on how to deploy models, an input field for the "Endpoint configuration name" (pre-filled with "endpointConfig-"), and a section for "Production variants".

**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-decision-forest-sm-test		variant-name-1	ml.m4.xlarge	none	1	1	Edit   Remove

[Add model](#)

**Create endpoint configuration**

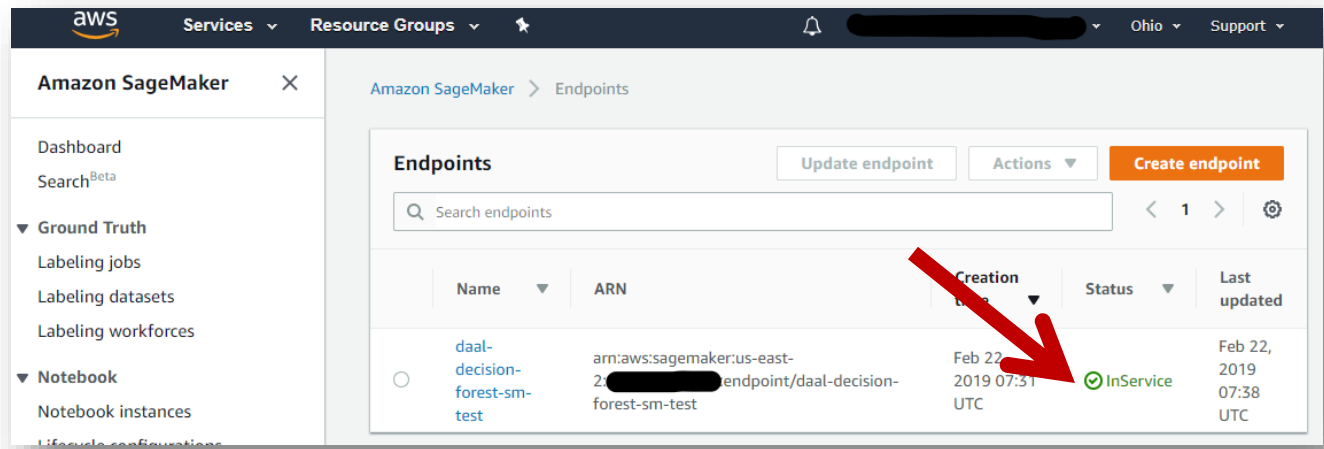
**Tags - optional**

Key	Value	Remove

[Add tag](#)

Cancel Previous **Submit**

18. Wait until endpoint is ready



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-172-31-11-84:~$ aws sagemaker-runtime invoke-endpoint --endpoint-name daal-pca-sm-test --body "$(cat probe_data.csv)" --content-type text/csv --accept text/csv output.txt
```

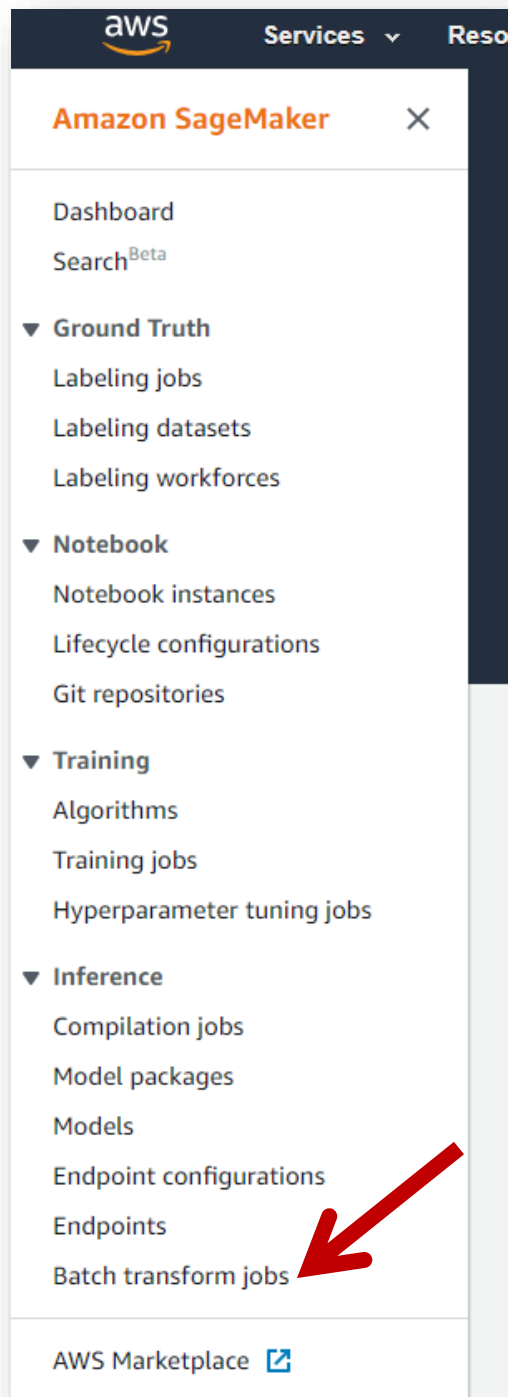
Then, see content of output file:

```
(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

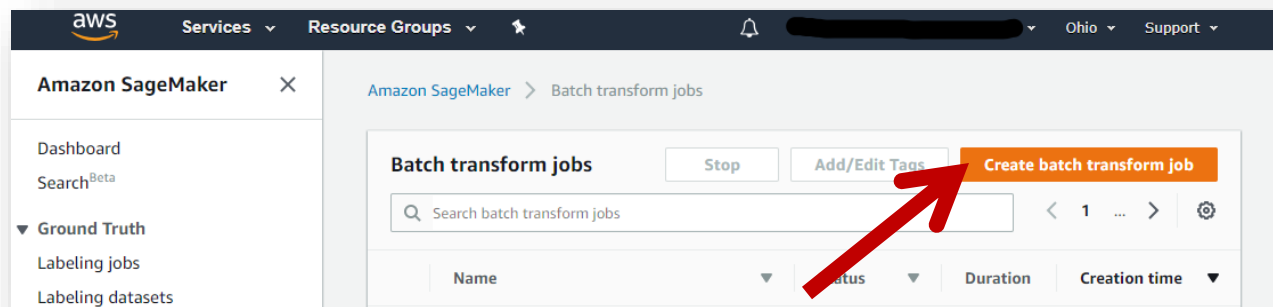
You can use batch transform job if you need compute predictions once.

1. Go to “Batch transform job” page

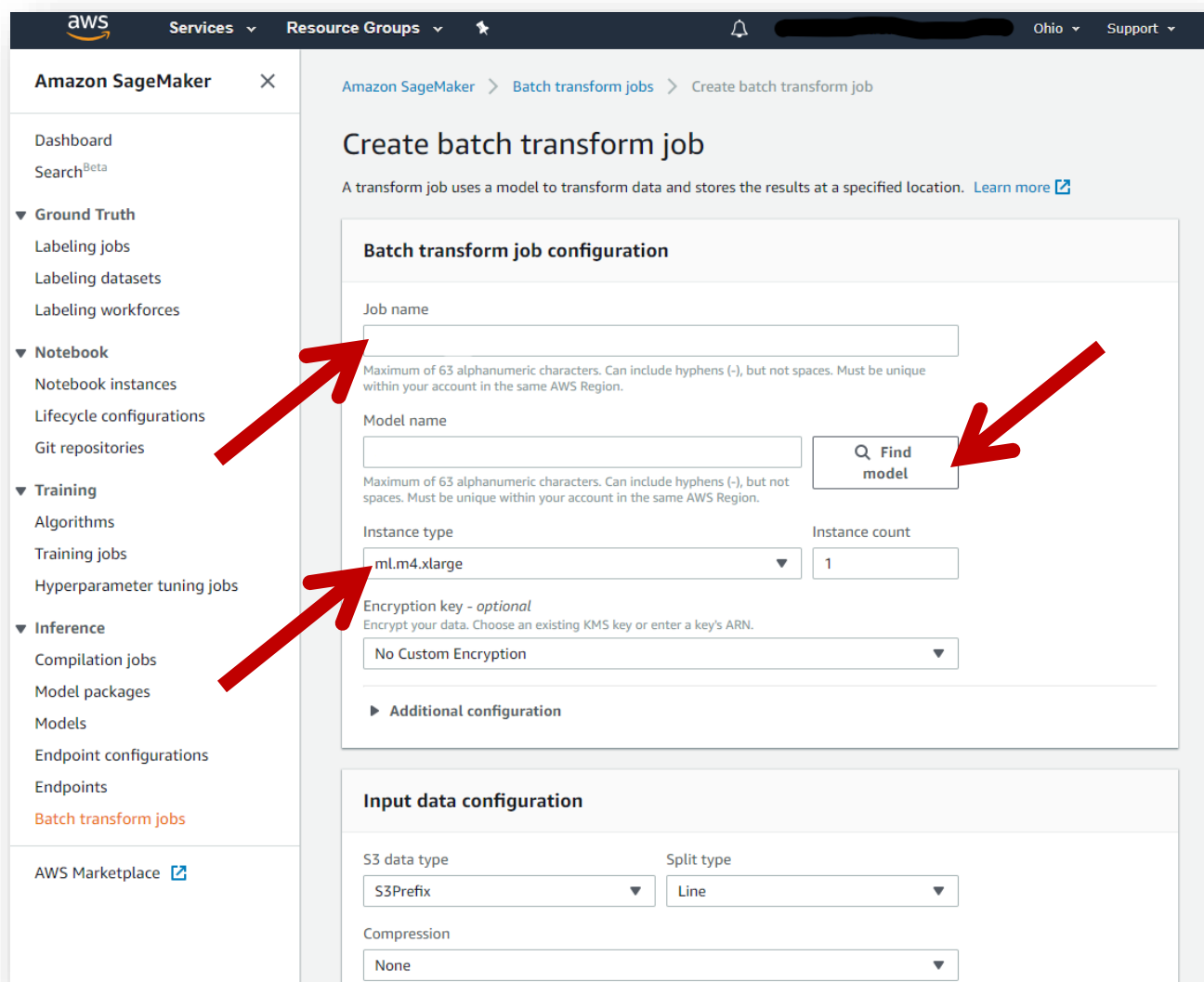




2. Click "Create batch transform job"



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



- Specify S3 location of input 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 'Output data configuration'. In the 'Input data configuration' section, the 'S3 data type' is set to 'S3Prefix' and 'Split type' is 'Line'. The 'Compression' is set to 'None' and 'Content type - optional' is 'text/csv'. The 'S3 location' field is empty, with a red arrow pointing to it and a link to 'go to Amazon S3'. The 'Output data configuration' section has an empty 'S3 output path' field, also with a red arrow pointing to it and a link to 'go to Amazon S3'. Below this, the 'Encryption key - optional' is set to 'No Custom Encryption'. The 'Accept - optional' is 'text/csv' and 'Assemble with' is 'Line'. At the bottom, there is a 'Tags - optional' section. The 'Create job' button is highlighted in orange at the bottom right, with a red arrow pointing to it.

Amazon SageMaker

Services Resource Groups

Dashboard  
Search<sup>Beta</sup>

▼ Ground Truth  
Labeling jobs  
Labeling datasets  
Labeling workforces

▼ Notebook  
Notebook instances  
Lifecycle configurations  
Git repositories

▼ Training  
Algorithms  
Training jobs  
Hyperparameter tuning jobs

▼ Inference  
Compilation jobs  
Model packages  
Models  
Endpoint configurations  
Endpoints  
Batch transform jobs

AWS Marketplace

### Input data configuration

S3 data type: S3Prefix Split type: Line

Compression: None

Content type - optional: text/csv

For content types that are available in built-in algorithms, [view our documentation](#)

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

An encryption key protects your data. Type the key ID or key ARN that you want to use.

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, Ground Truth, Labeling jobs, Labeling datasets, Labeling workforces, Notebook, and Notebook instances. The main content area displays the "Batch transform jobs" 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, but the batch transform job "daal-sm-test" is now in a "Completed" status. The status column shows a green checkmark and the word "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