How to use Intel® DAAL kNN via SageMaker web interface

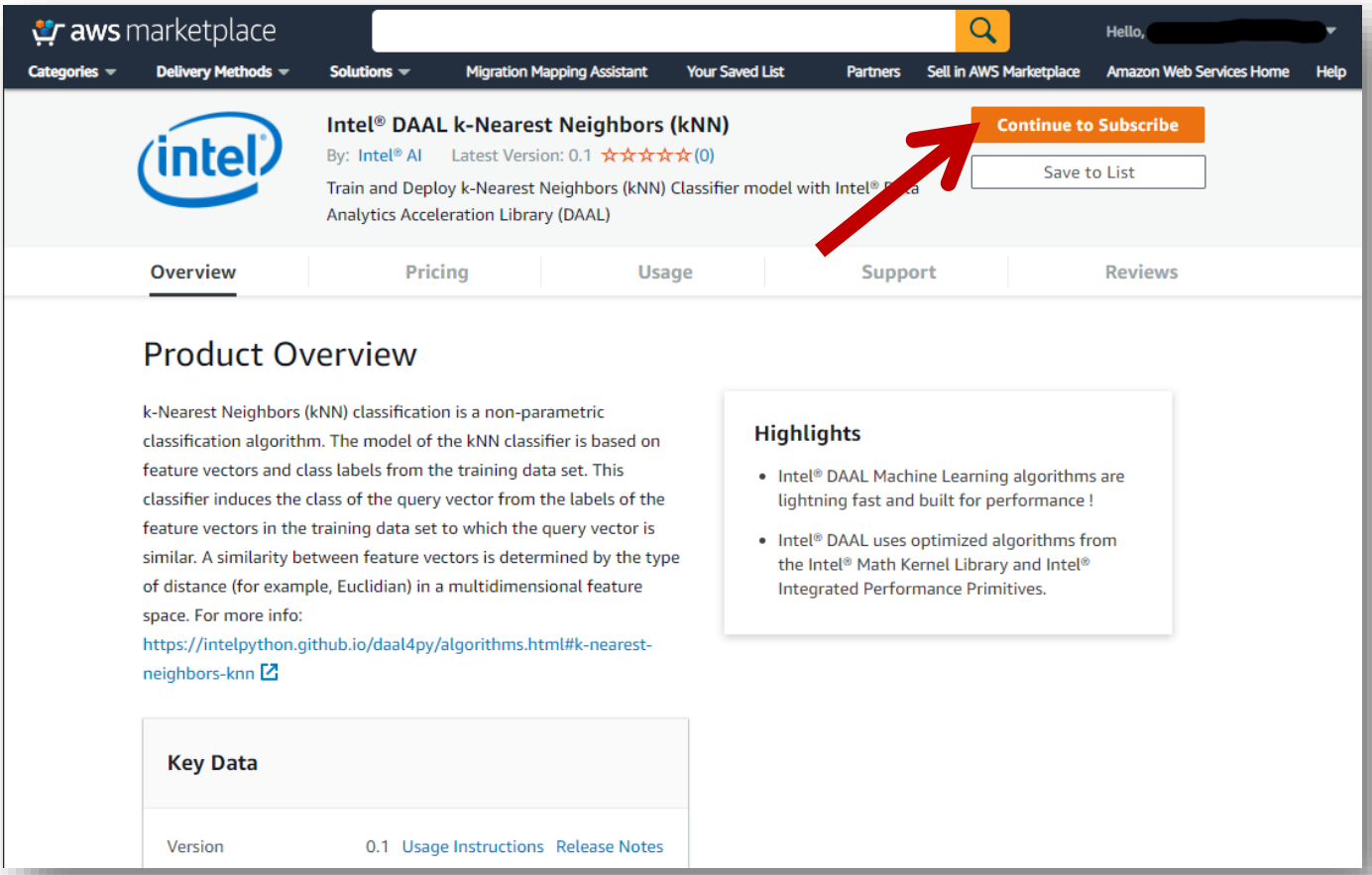
Description of algorithm:

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.

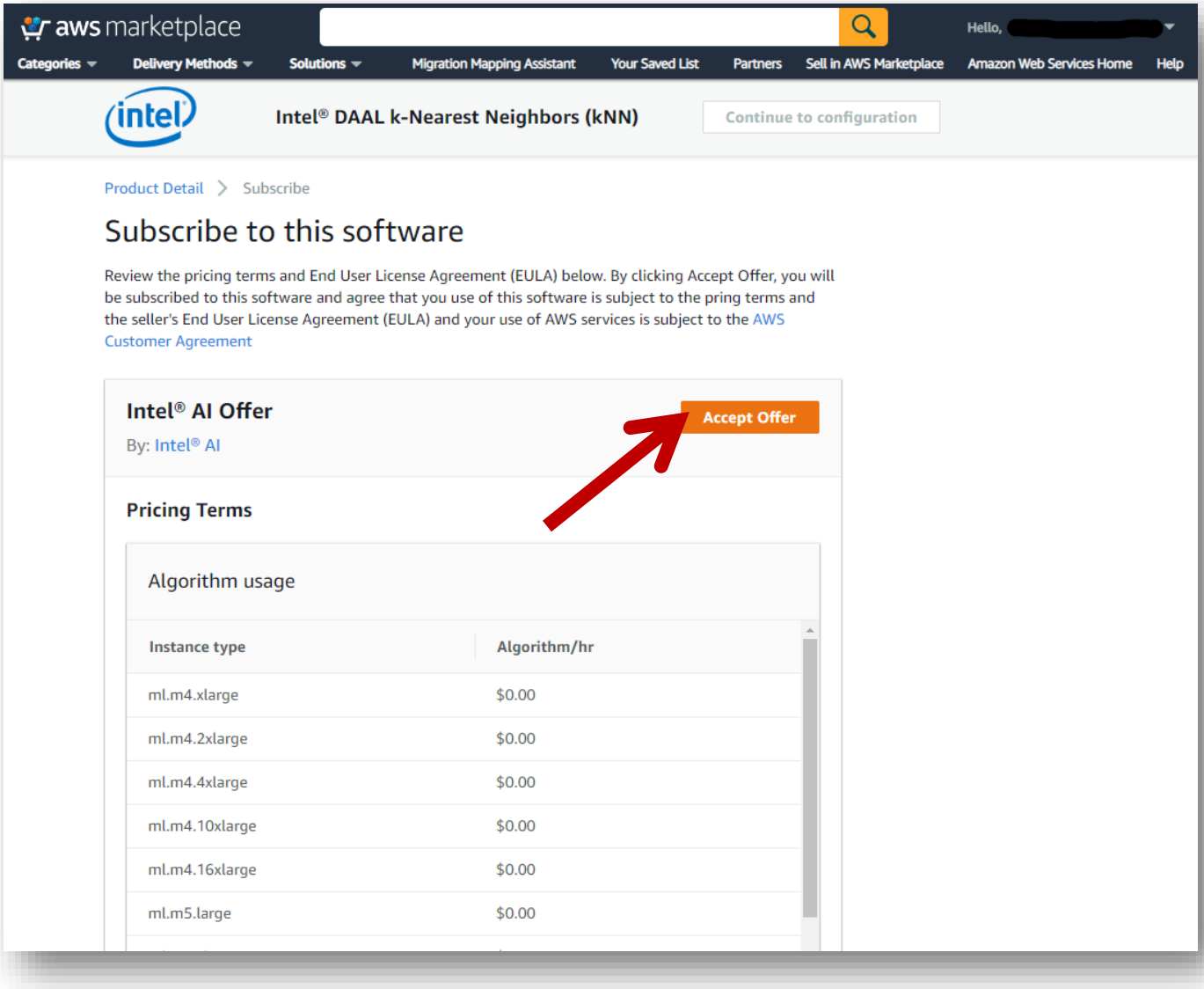
Intel® DAAL developer guide Intel® DAAL documentation for kNN

Instruction:

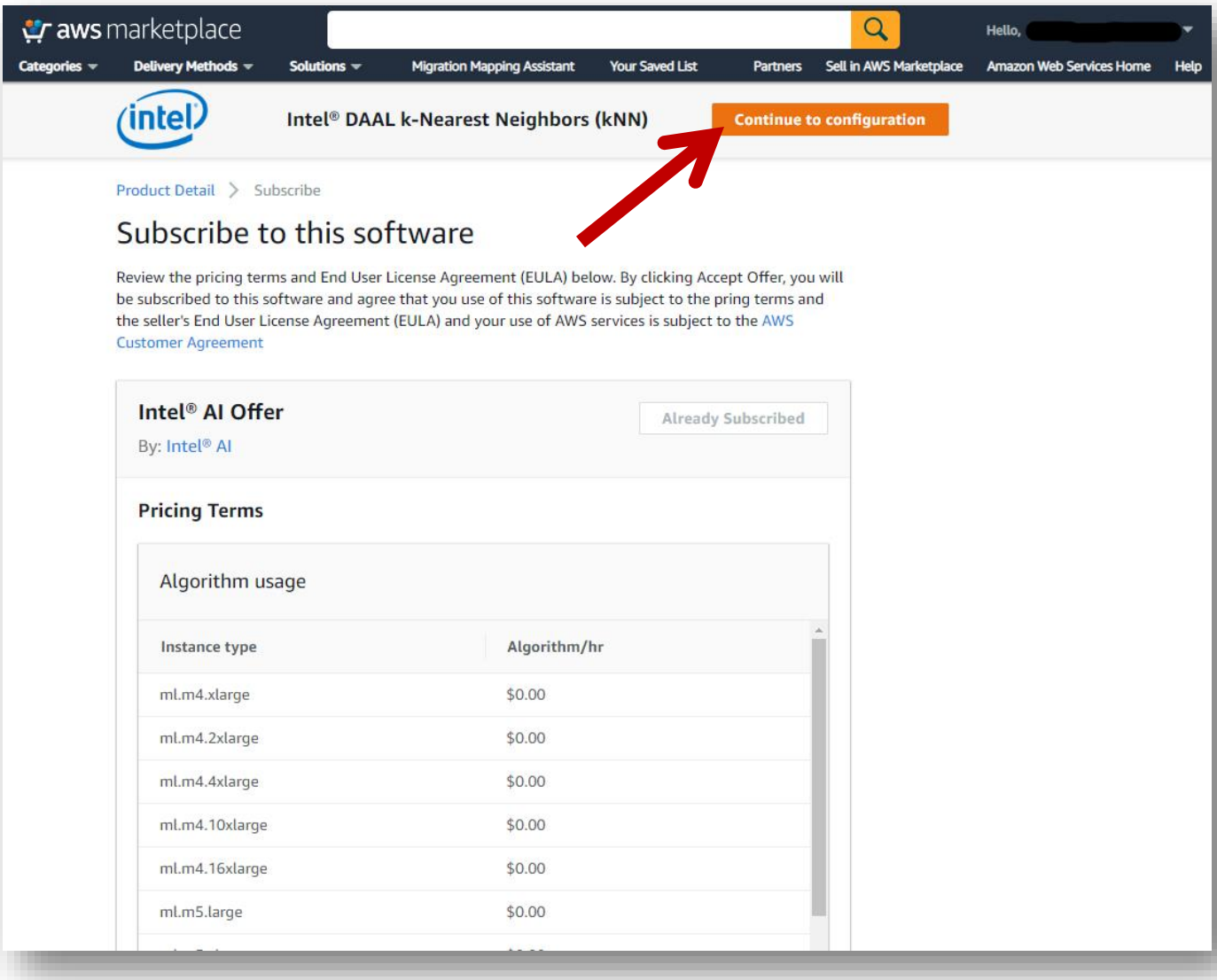
1. Visit page on SageMaker Marketplace and click “Continue to Subscribe”



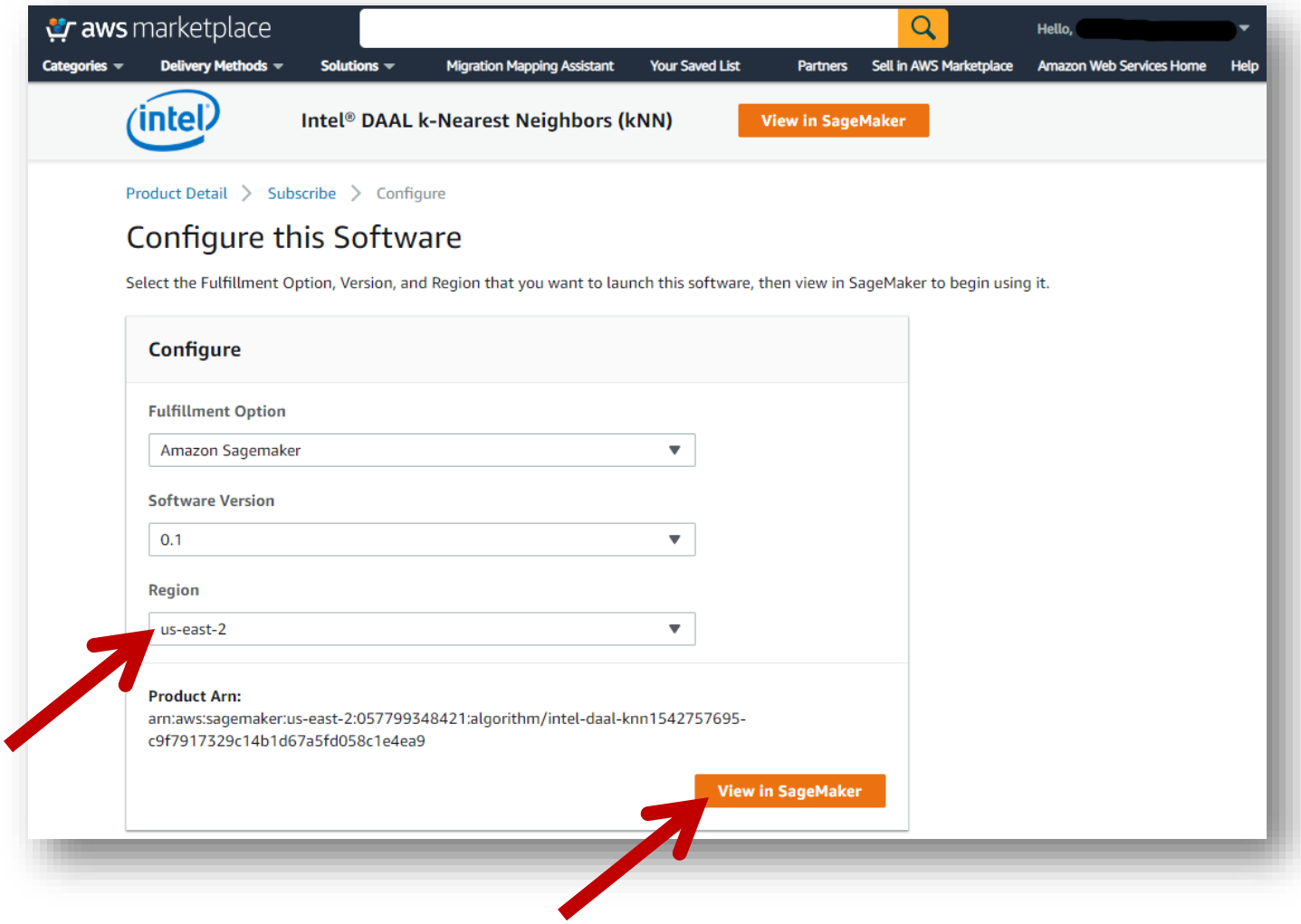
2. Click “Accept Offer” if you agree with EULA at end of page. If you already subscribed on algorithm on Marketplace this step will be skipped.



3. Click “Continue to configuration”



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



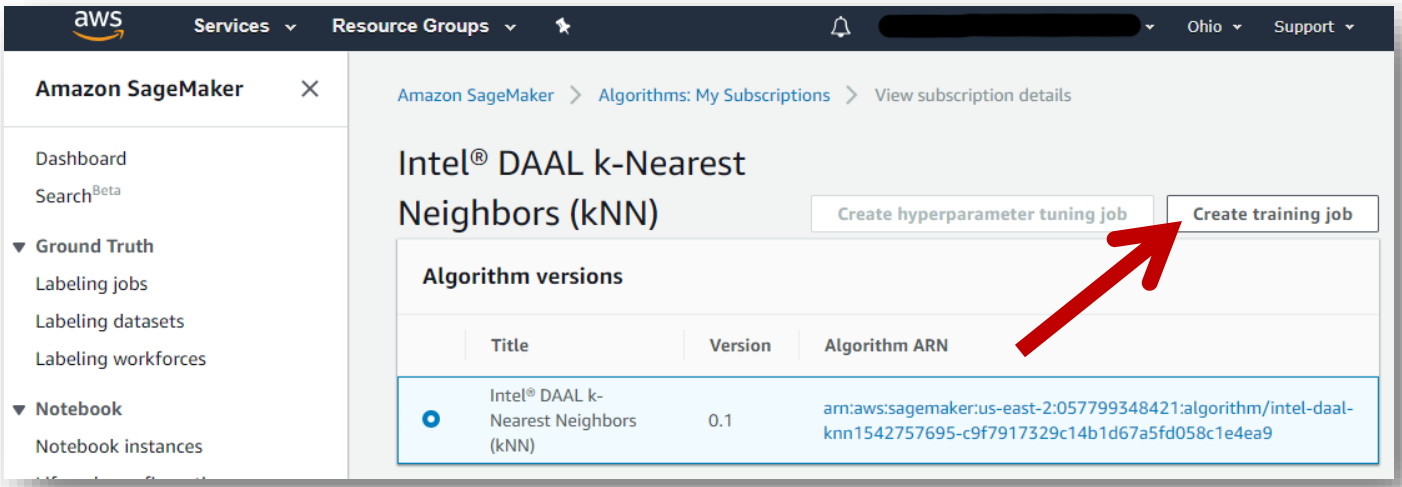
kNN algorithm on AWS SageMaker is divided into two stages: training job and getting inference from endpoint.

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

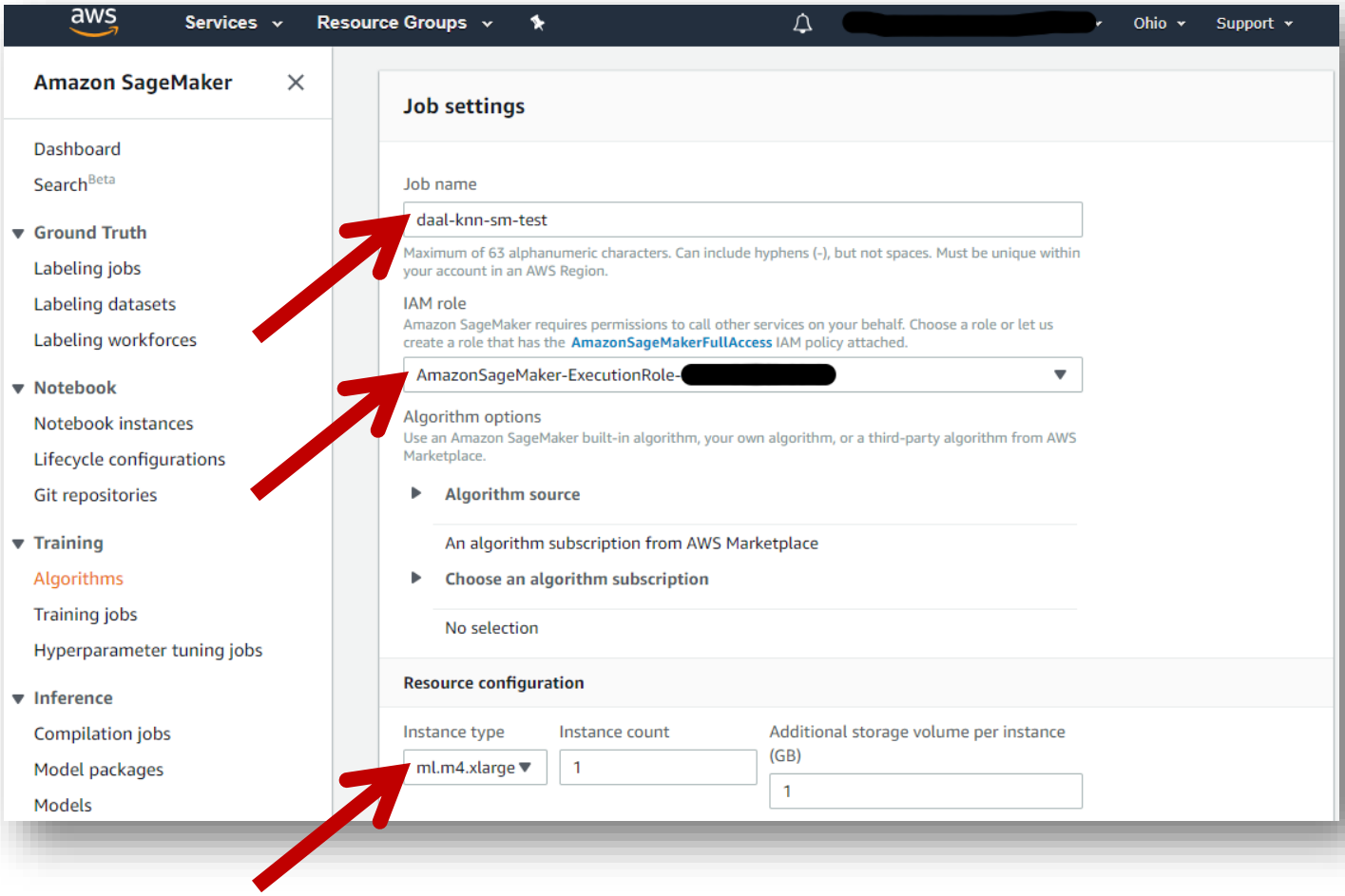
After that, you should create model with computed values and endpoint based on it.

Sending data to endpoint gives you predictions in response

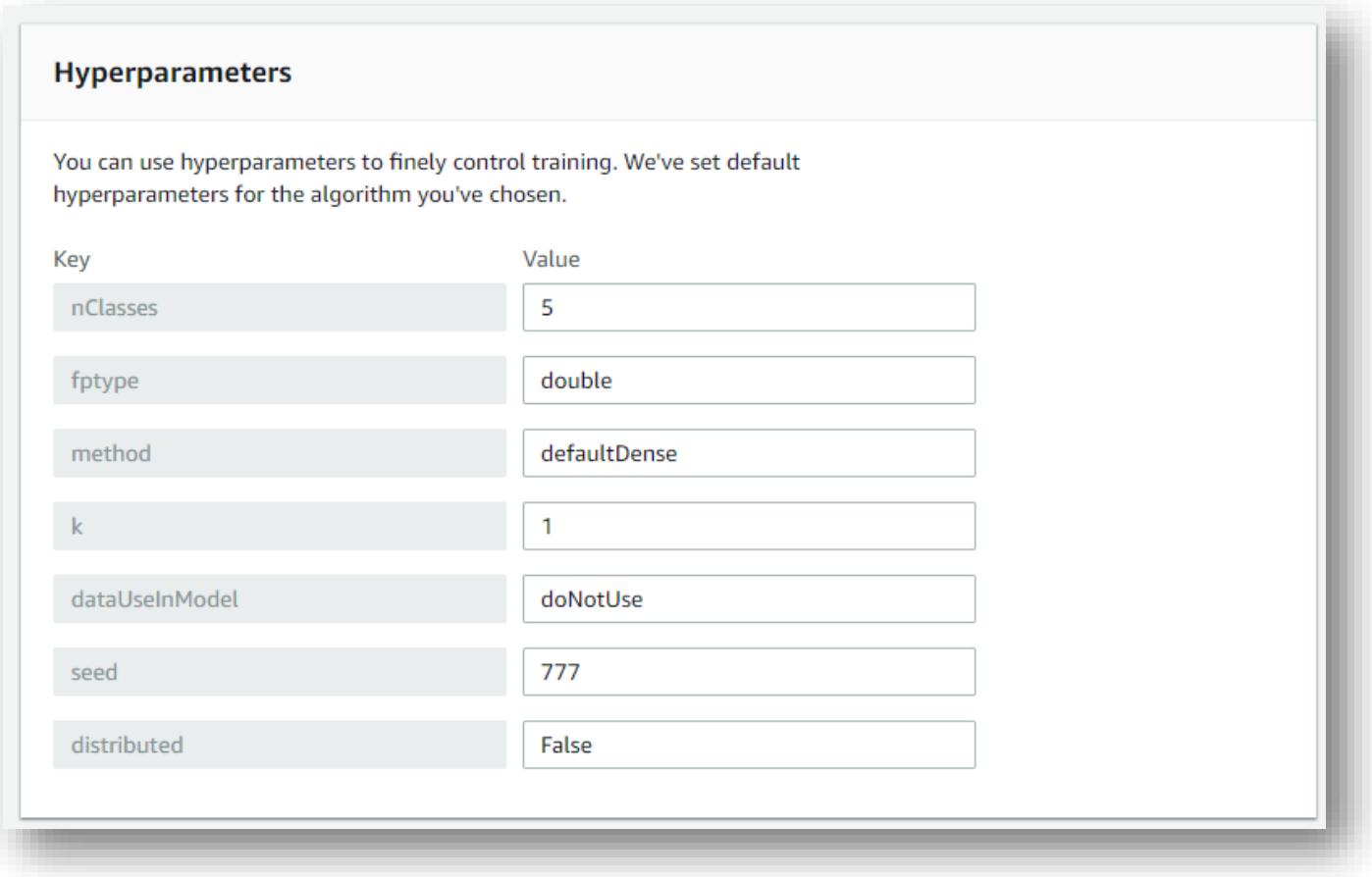
5. Select needed algorithm version and click “Create training job”



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

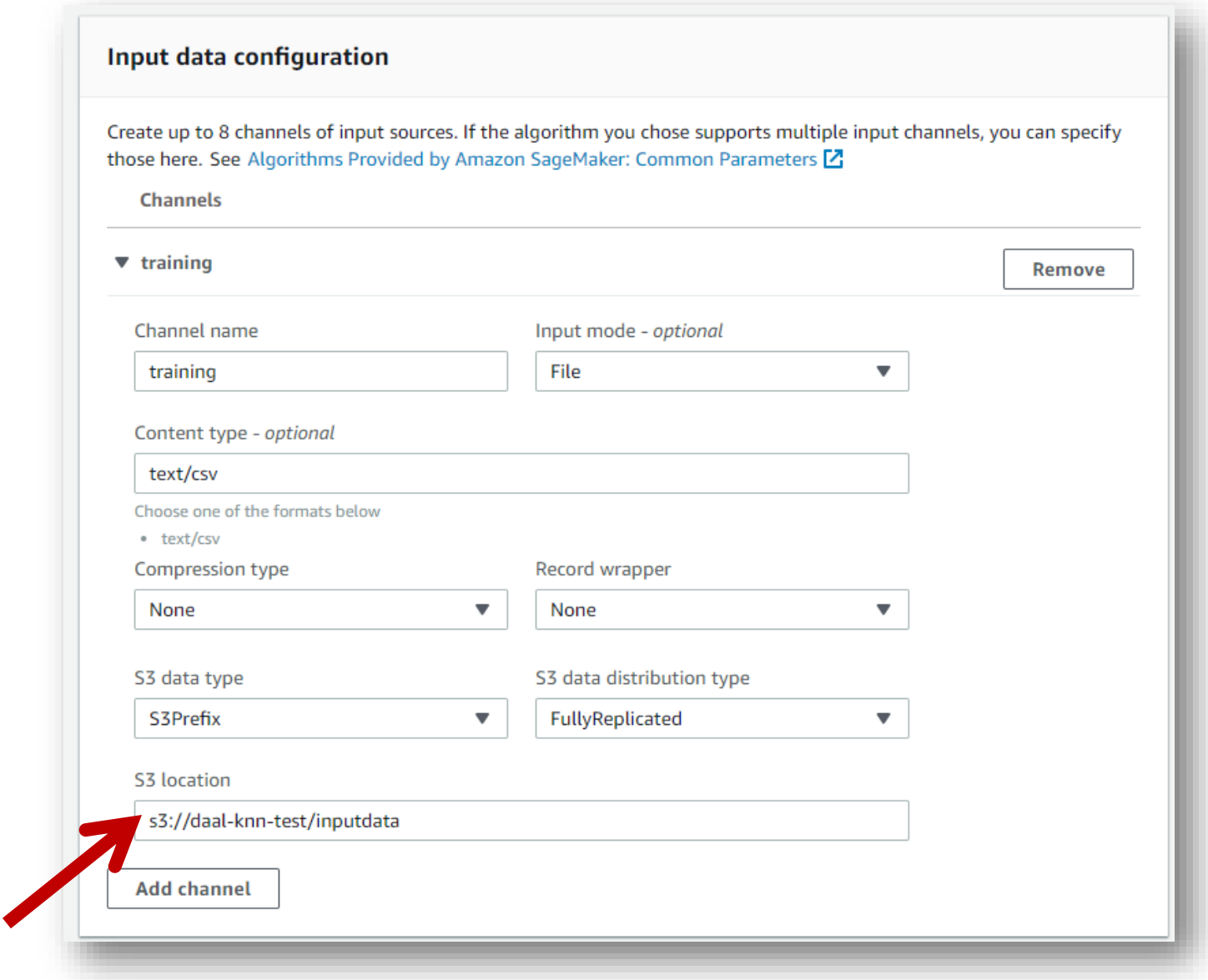


7. Choose hyperparameters

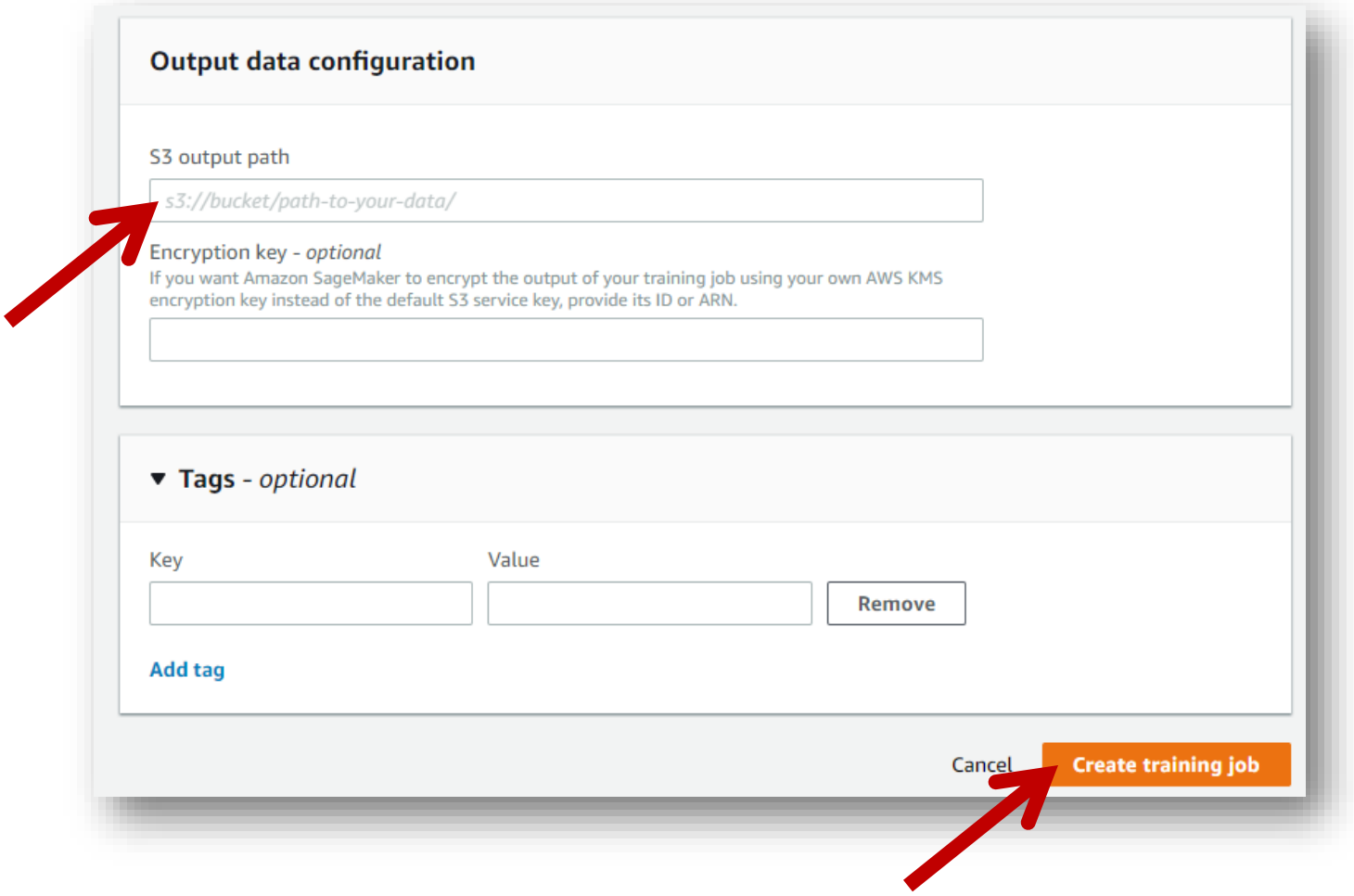


Parameter name	Type	Default value	Description
nClasses	int	2	Number of classes in data
fptype	str	"double"	The floating-point type that the algorithm uses for intermediate computations. Can be "float" or "double"
method	str	"defaultDense"	The computation method used by the K-D tree based kNN classification. 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.

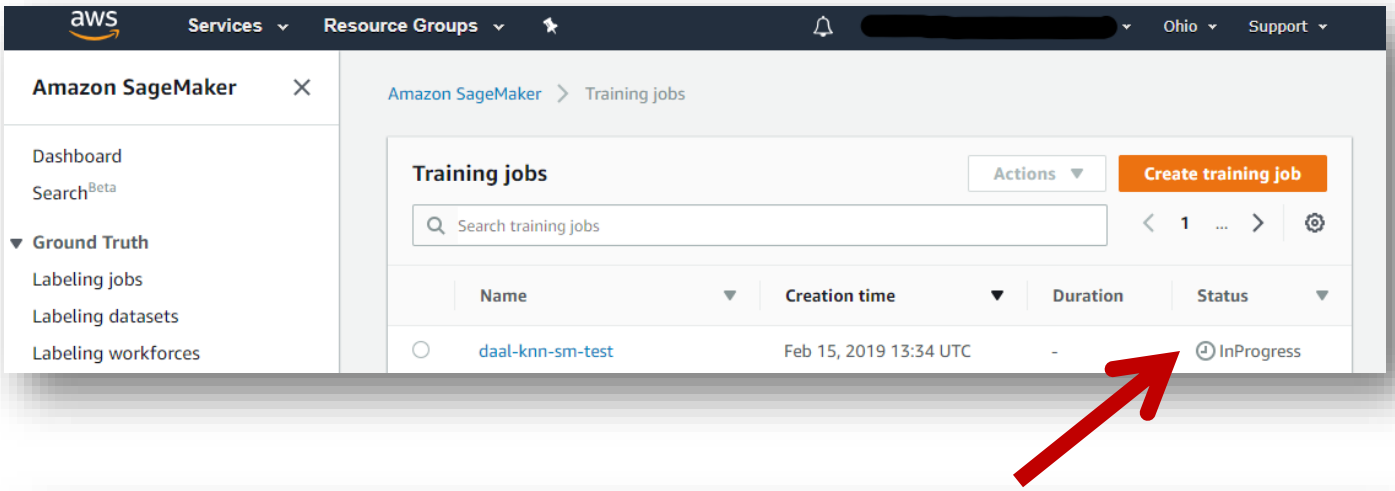
8. Specify S3 location of input data for training

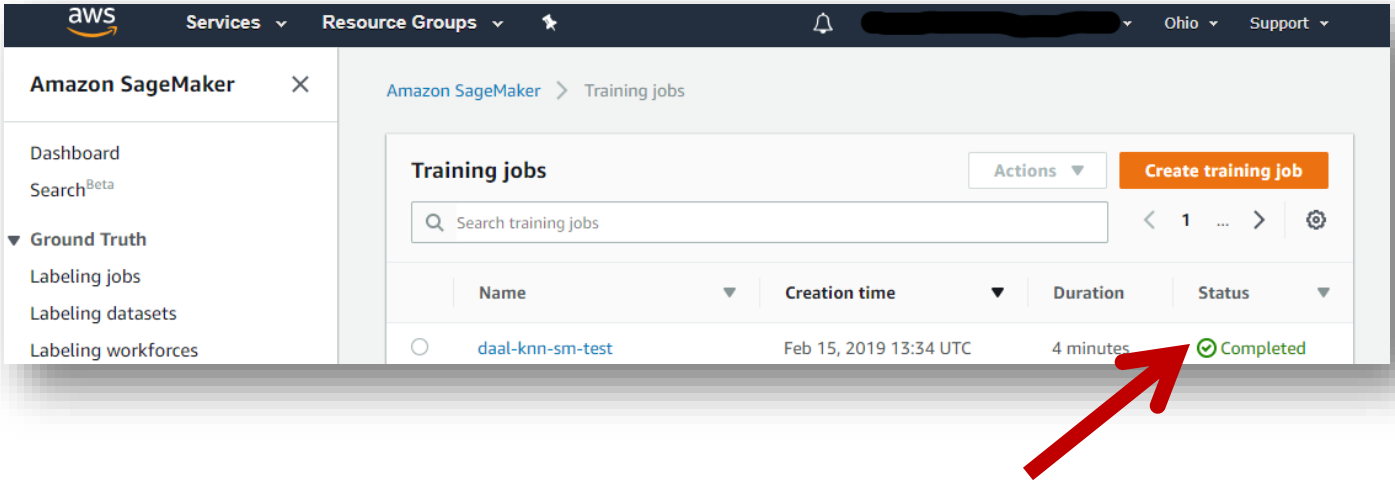


9. Specify S3 output path (model will be stored here) and click “Create training job”

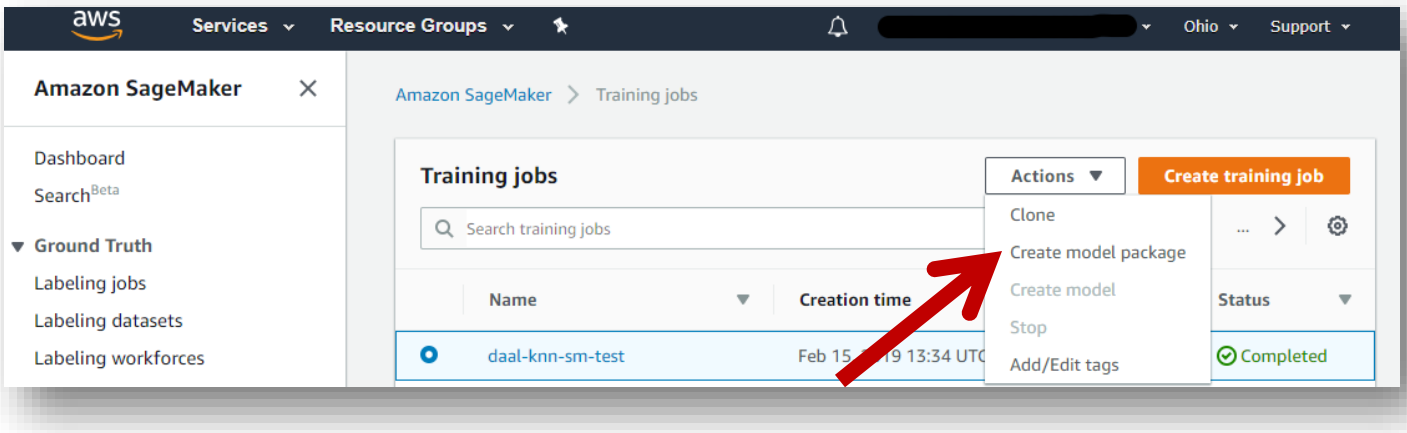


10. Wait until finish of training job

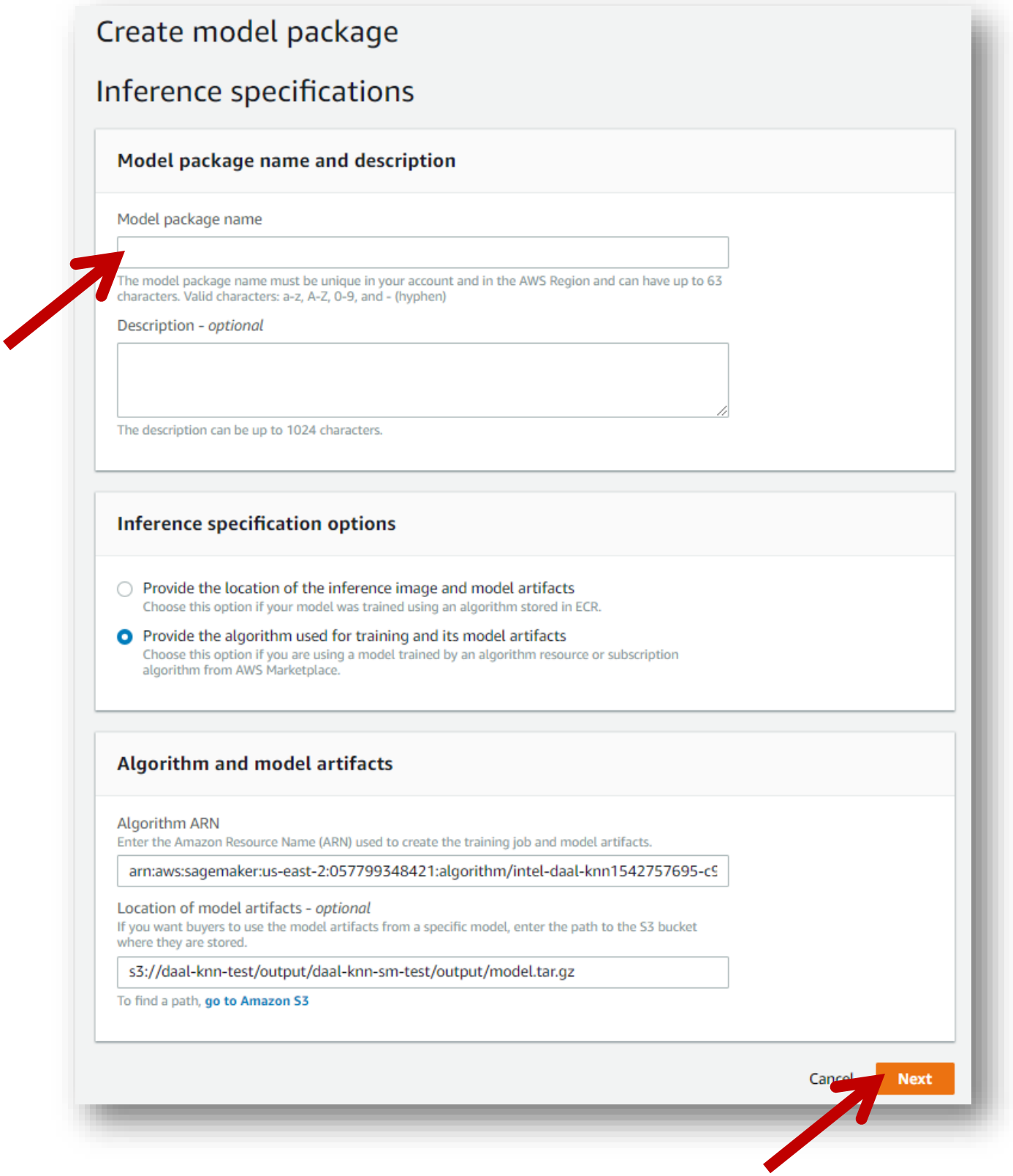




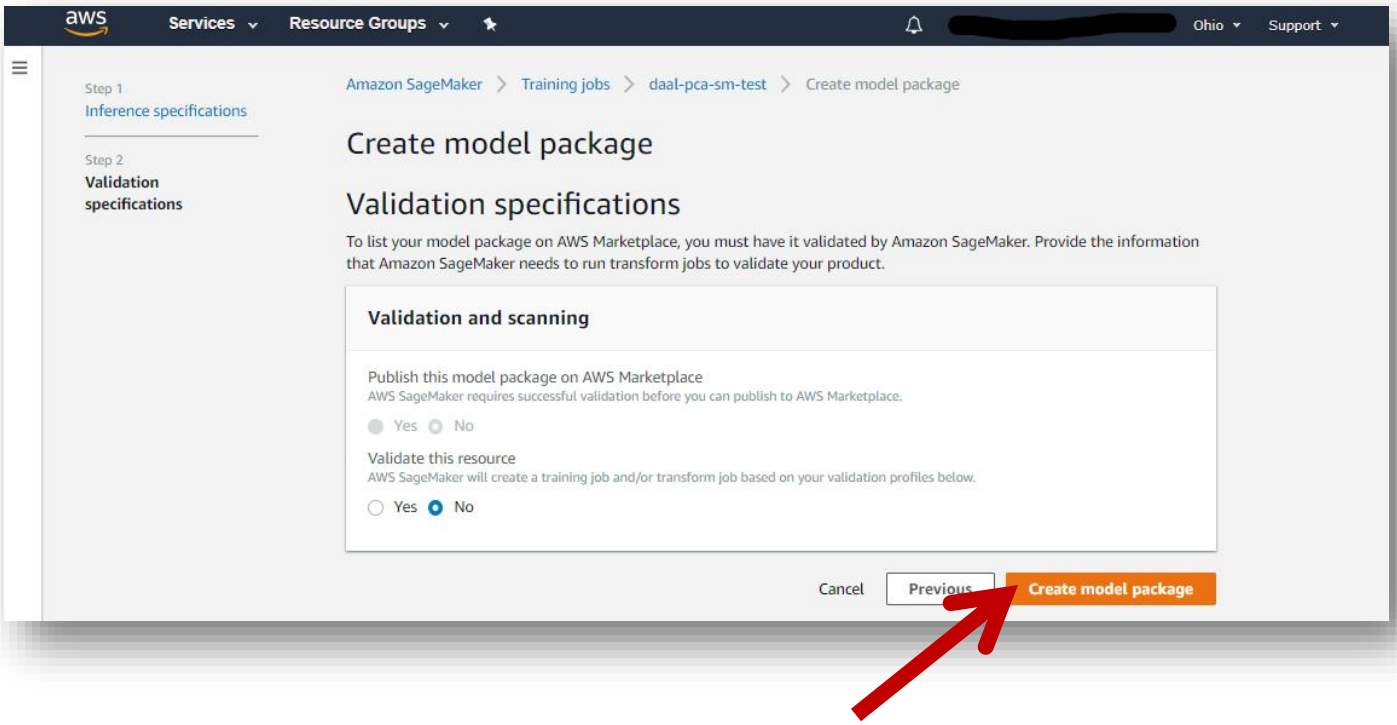
11. Select training job and take action “Create model package”



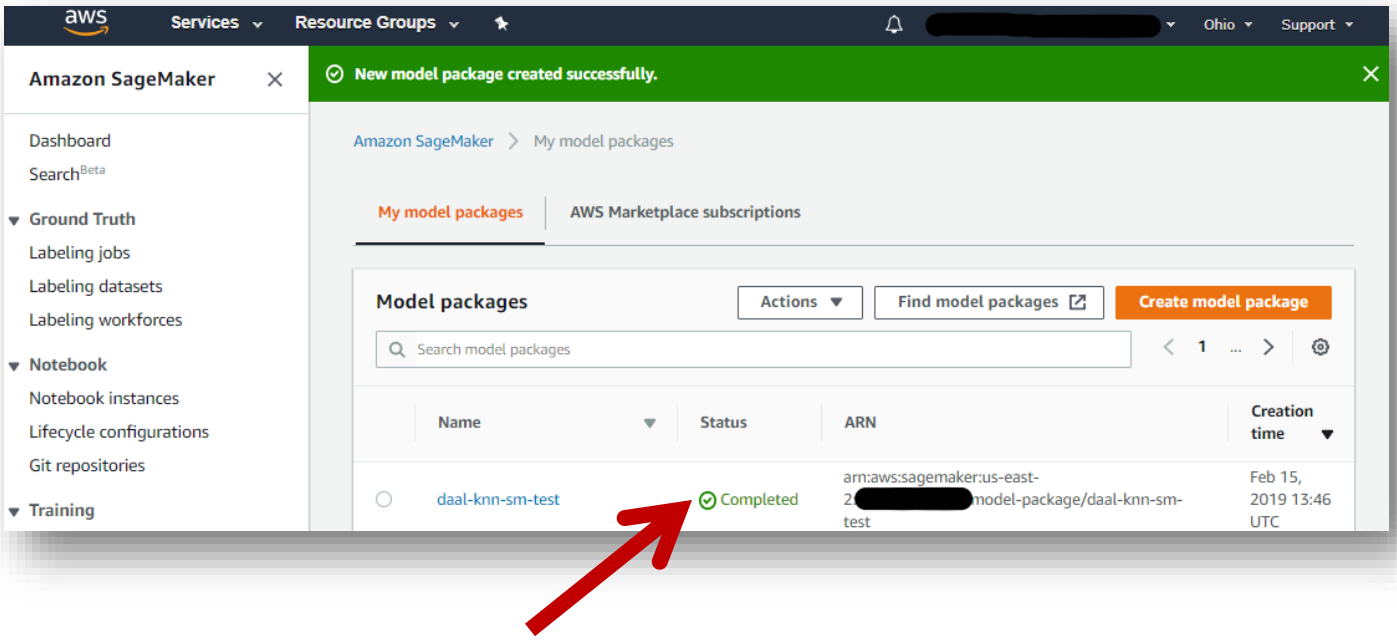
12. Type model package name and click “Next”



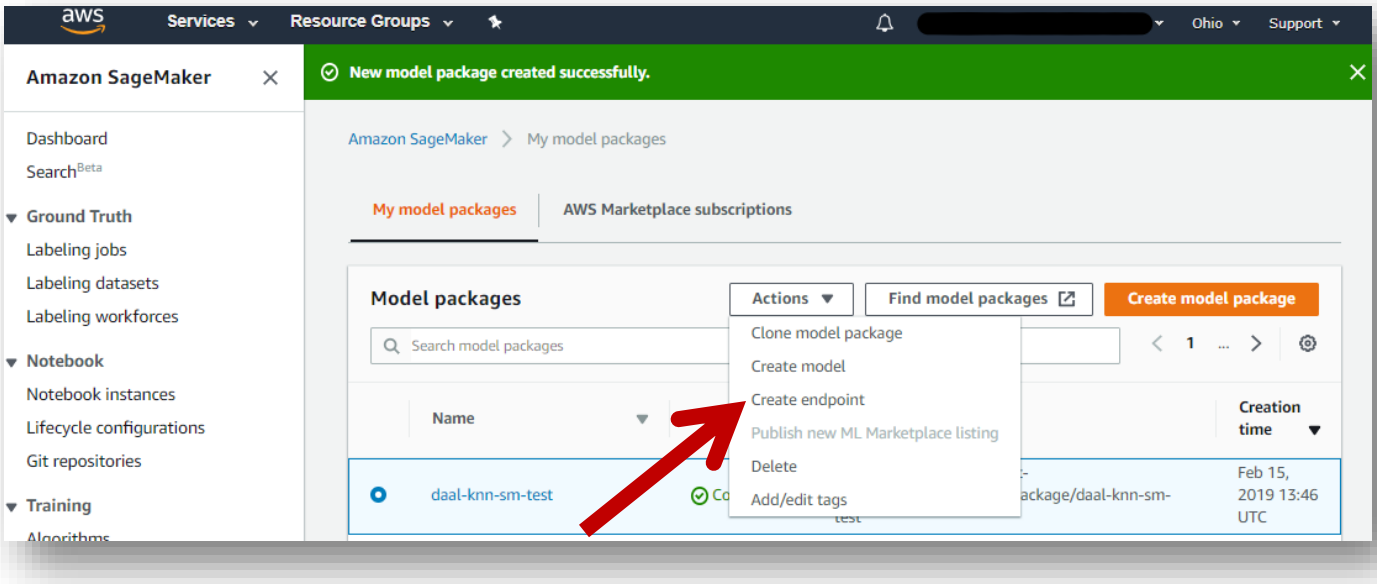
13. Click “Create model package”



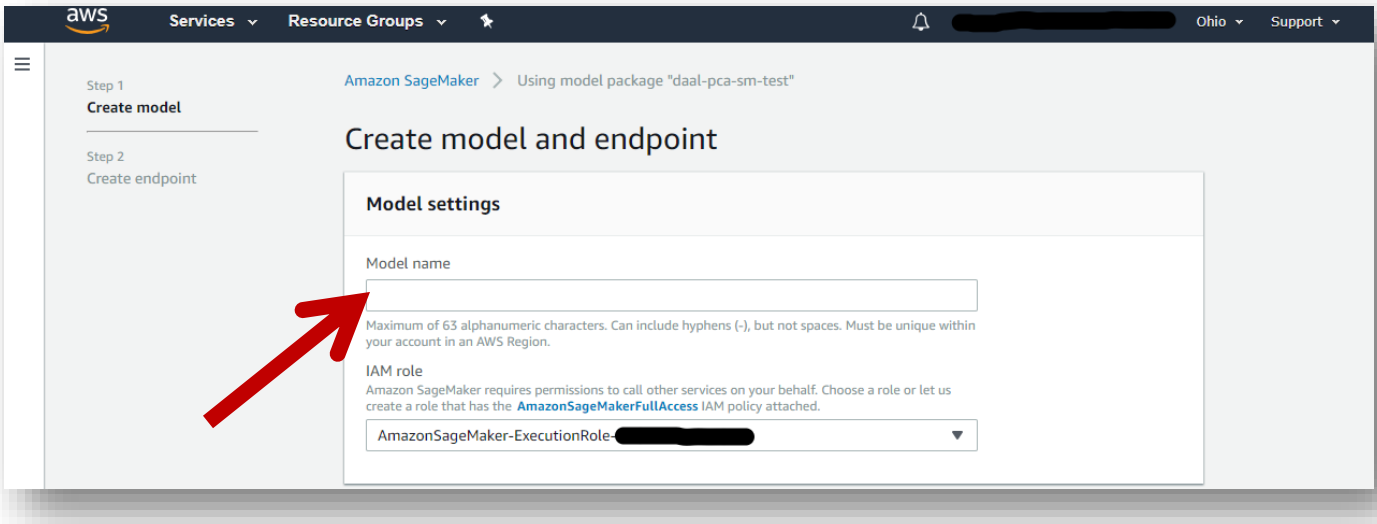
14. Wait until package is created



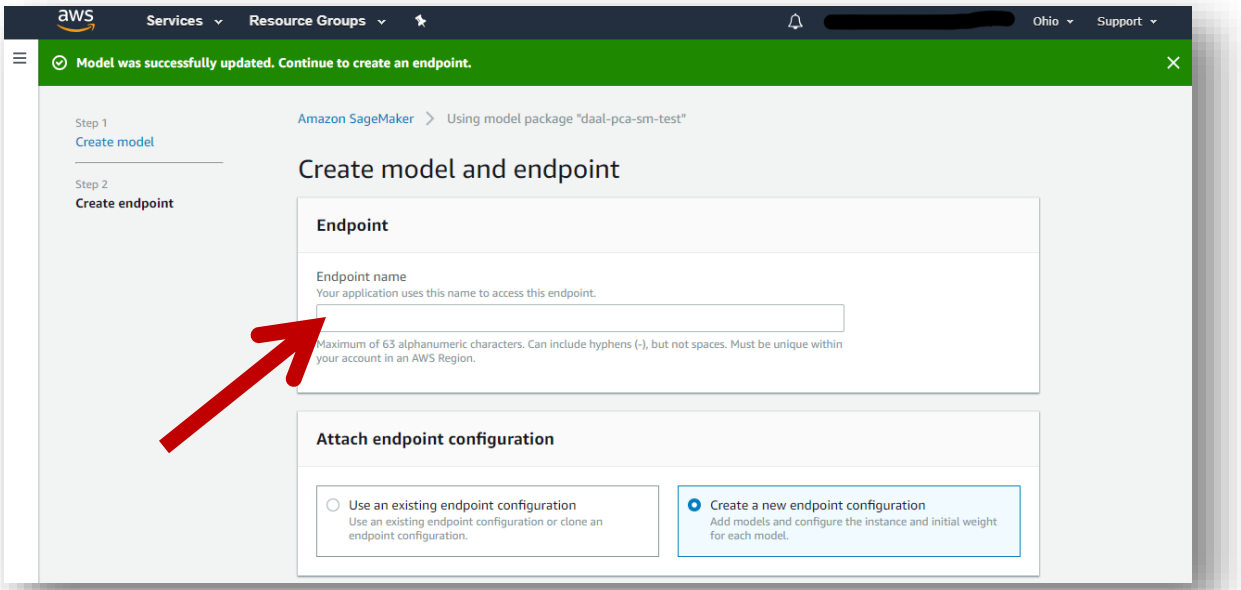
15. Select package and take action “Create endpoint”

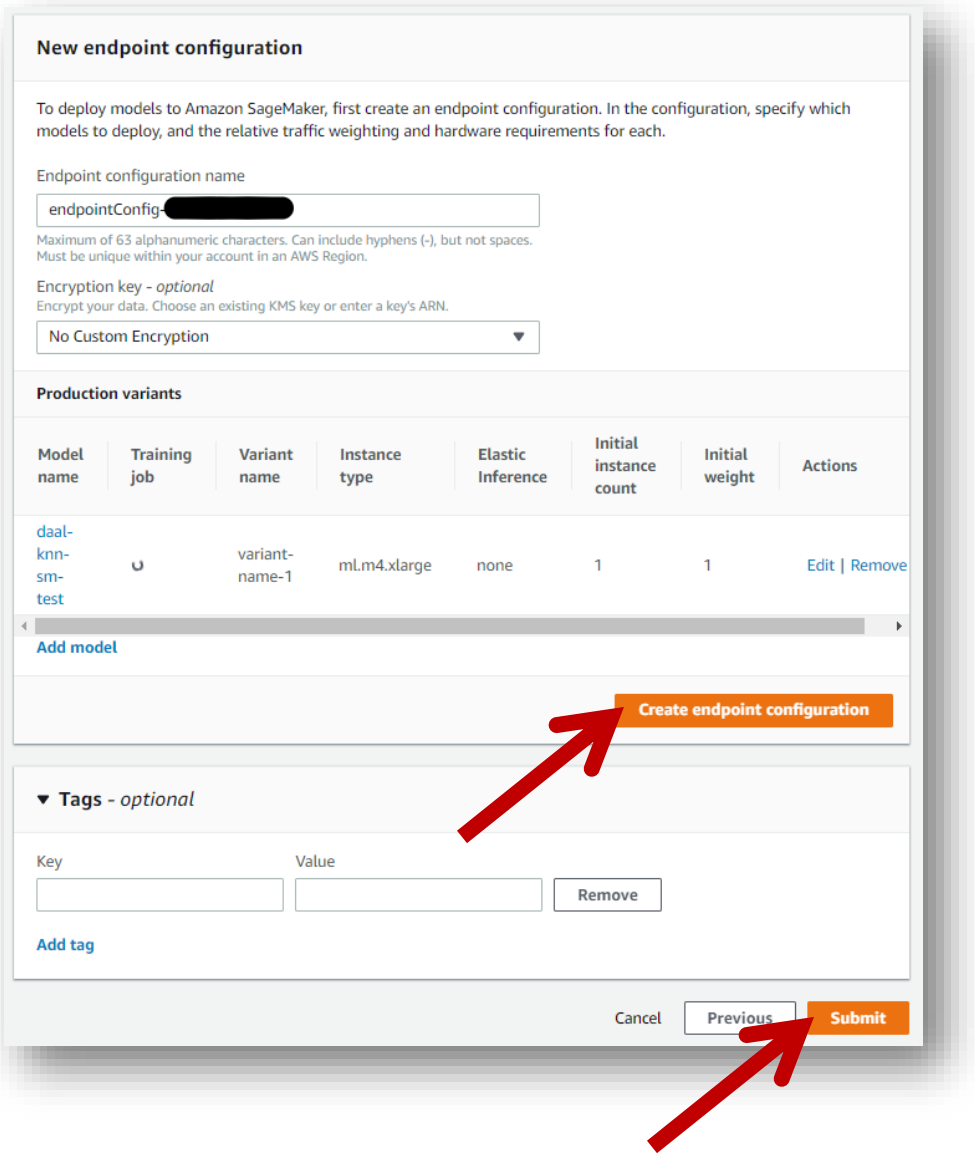


16. Type model name and click “Next”

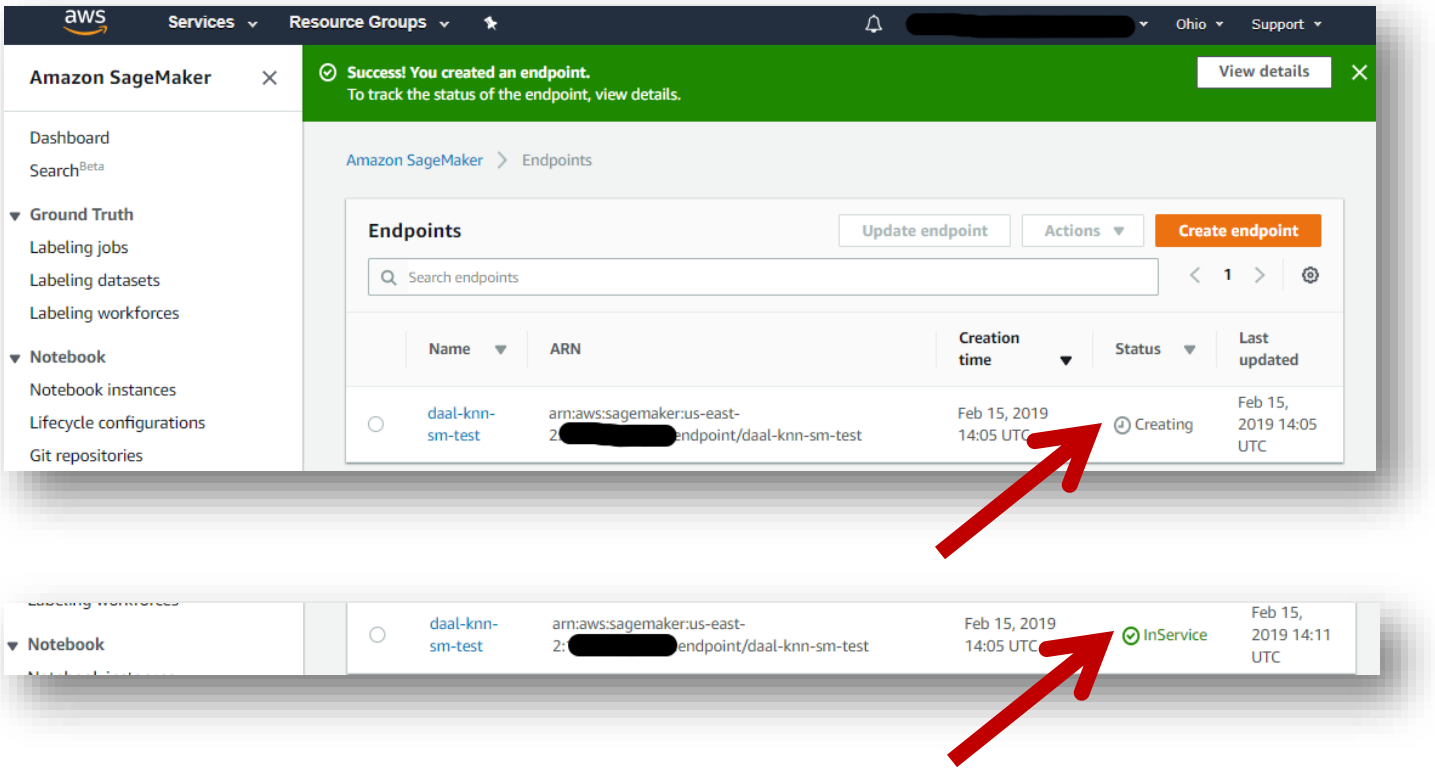


17. Type endpoint name, edit and create endpoint configuration and click “Submit”





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-172-31-22-46:~$ 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
```

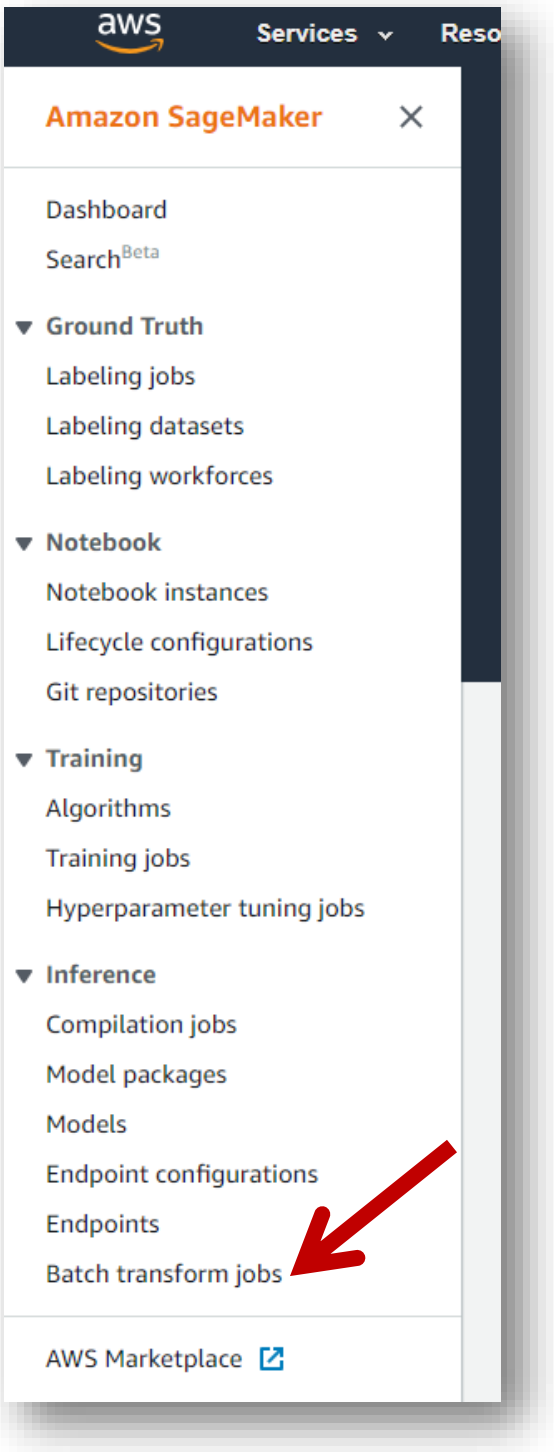
Then, see content of output file:

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

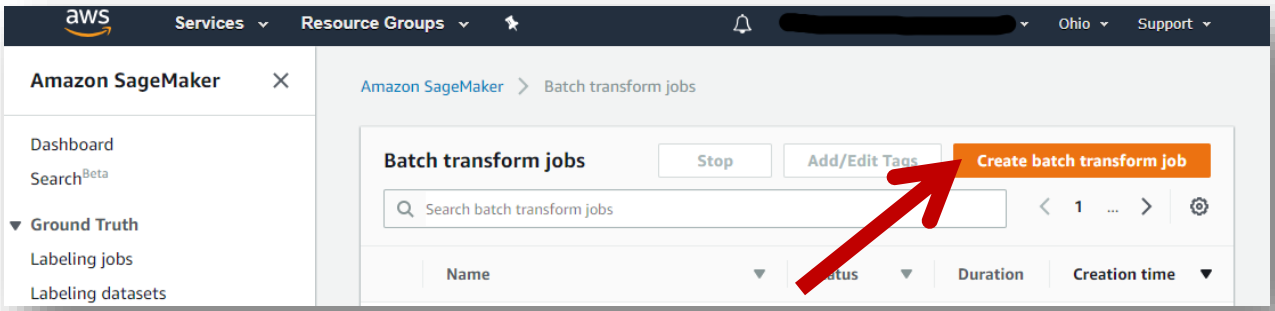
Batch transform job as alternative to endpoint

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

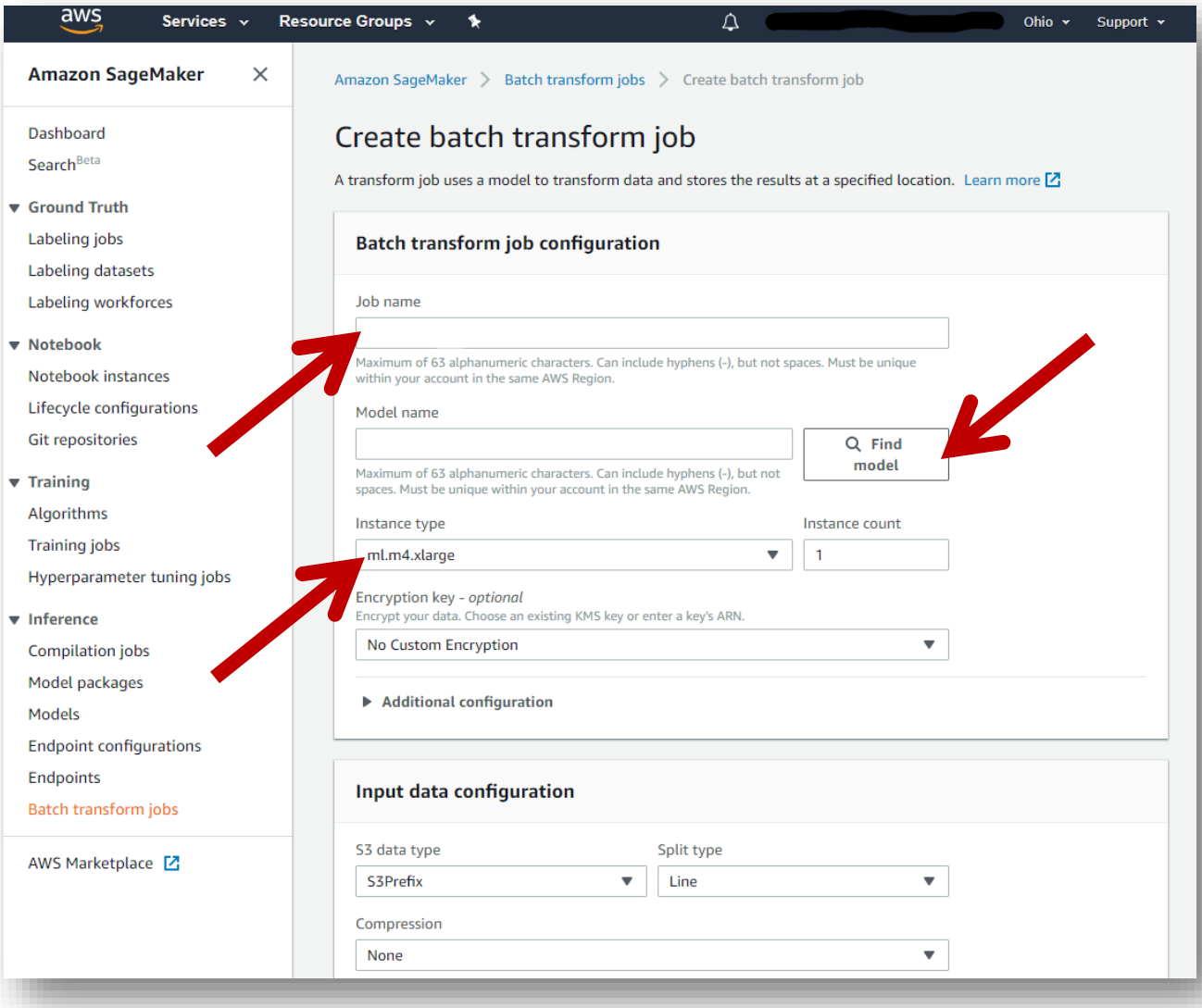
1. Go to “Batch transform job” page



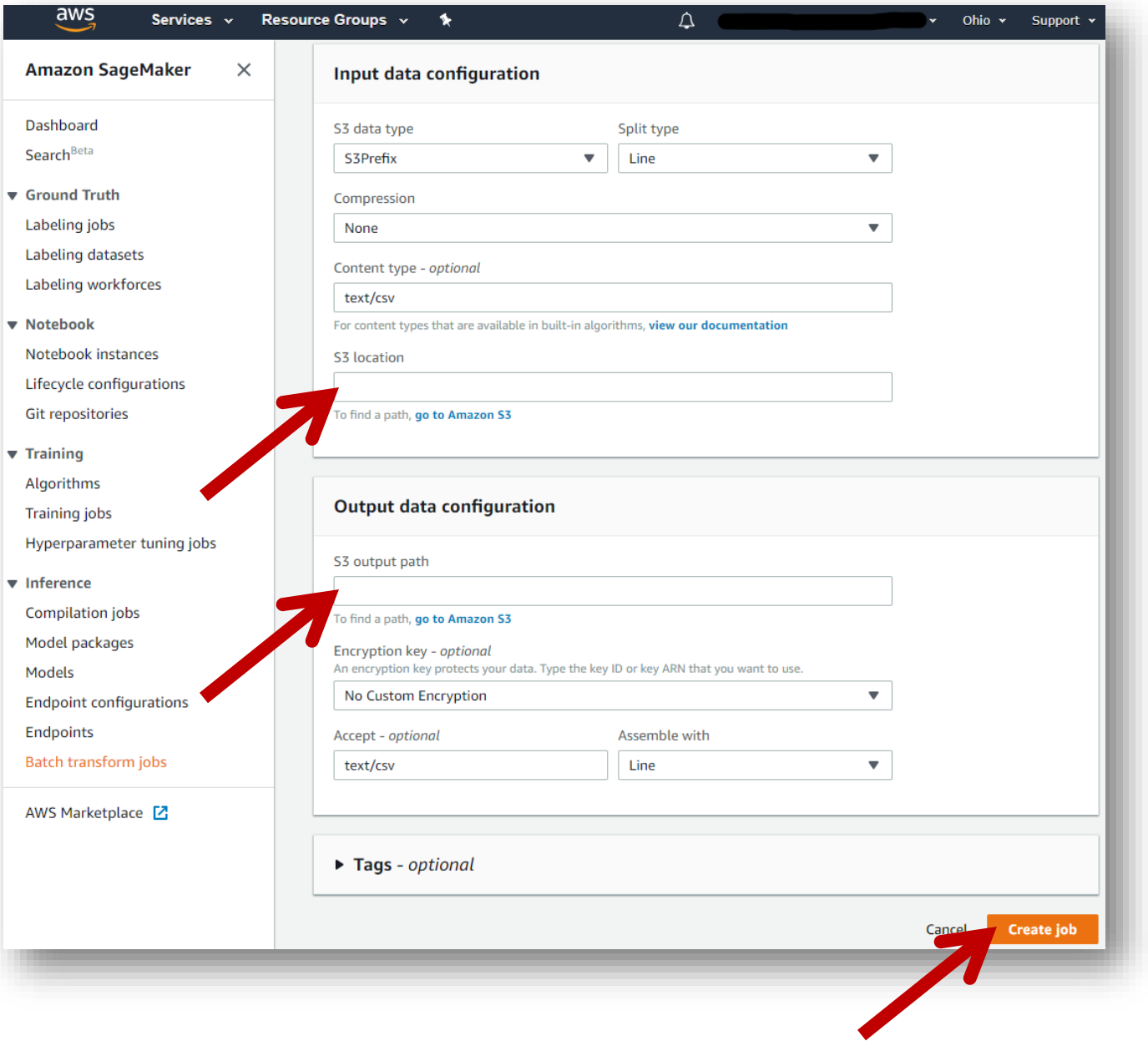
2. Click “Create batch transform job”



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



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



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