How to Use Kaggle

Current Page: Models

Models

Use and share pre-trained models

What is Kaggle Models

Kaggle Models provides a way to discover, use, and share models models for machine learning and generative AI applications. Kaggle Models is a repository of pre-trained models that are deeply integrated with Kaggle's platform including for easy to use in Kaggle Competitions and Notebooks. Like Datasets, Kaggle Models organize community activity that enrich models' usefulness: every model page will contain discussions, public notebooks, and usage statistics like downloads and upvotes that make models more useful.

Kaggle Models is a new product which the Kaggle team will continue to develop and improve based on what the community would like to see. If you'd like to make suggestions for improvements or new features or report bugs, we recommend you create a new topic on the Product Feedback forum.

Where do Models come from?

Kaggle Models come from a variety of sources including partners that we collaborate with on releases like Meta's Llama 2 and Alibaba's Qwen, integrations with modeling libraries like Keras, and the community of millions of Kagglers sharing fine-tuned variants and other innovations.

Finding Kaggle Models

You can find Kaggle Models by using the Models landing page. There are a number of filters and sorts plus free text search. For instances you can search by:

- Filtering to Keras models
- Filtering by the task tag you want (e.g., classification)
- Filtering by model size
- Searching "Gemma" or other keywords in the free text search
- Sorting by number of upvotes
- Etc.

performing well or are otherwise popular for tasks relevant to your use case. Competitors commonly share which models they're using in public notebooks and in discussion write-ups. When you fork a notebook that has a model from Kaggle Models attached to it, your copy will also have the same model attached.

Finally, you can also search for models from within the notebook editor. Use the "Add Models" component in the right-hand pane of the editor to search and attach models to your notebooks. This works similarly to Datasets.

Understanding the model detail page

When you click on a model you will be taken to the "detail page" for that model. For example, this is the detail page for a BERT model. The model detail page contains an overview tab with a Model Card (metadata and information about how the model was trained, what its acceptable use cases are, any limitations, etc.), a framework and variation explorer, and a usage dashboard. There are tabs for notebooks and discussions. If a model is useful, you can upvote it.

Beyond the overall metadata, a model detail page also organizes all variations and frameworks for a given model. For example:

- **Variations**: The same model with different numbers of parameters, e.g., small, medium, and large.
- **Frameworks**: The same model with different ML library compatibility, e.g., TensorFlow, PyTorch, etc.

You can view and use the specific framework and variation that you want by selecting it in the file explorer on the overview page beneath the Model Card. From here, you can use click "New Notebook" to attach it to a new notebook to start using the model.

Using Kaggle Models

There's two broad ways that Kaggle Models are useful: on Kaggle and outside of Kaggle (e.g., in production applications or using non-Kaggle tools like Colab, etc.).

On Kaggle
Currently, Kaggle Models very useful within the context of Competitions, specifically for use within Notebooks. Start by either forking a notebook that has a model attached (you can view the attached models on the "Input" tab of any notebook), creating a new notebook on a model, or adding a model to a new notebook from the right-hand pane of the editor.

You'll be prompted to confirm your framework and model variations(s), then simply copy and paste the starter code to load the model.

Outside of Kaggle
Many developers will need to download models in code outside of Kaggle. There are a few different methods: via the kagglehub Python library, via our Kaggle CLI, or by calling the API directly.

Before providing instructions for each of these methods, it's helpful to know that you will need to know how to authenticate in order to access certain models like Gemma that require Kaggle credentials in order to confirm that user consent to the custom license has been verified. Obtain credentials from the “Settings” page when logged-in to Kaggle and clicking on the "Create New Token" button under the "API" section.

The examples below allow you to download the `2b` PyTorch variation for the google/gemma model. If a model doesn't have a restricted license like Gemma, you'll be able to skip the `kagglehub.login()` steps in the examples below.

Method 1. Via the kagglehub Python library

See kagglehub documentation.

```
import kagglehub

# Authenticate
kagglehub.login() # This will prompt you for your credentials.
# We also offer other ways to authenticate (credential file & env
variables): https://github.com/Kaggle/kagglehub?tab=readme-ov-
file#authenticate

# Download latest version
path = kagglehub.model_download("google/gemma/pyTorch/2b")

# Download specific version (here version 1)
```

```
path = kagglehub.model_download("google/gemma/pyTorch/2b/1")

print("Path to model files:", path)
```

Method 2. Via the Kaggle CLI

See documentation. Follow steps here to authenticate with credentials.

```
# Authenticate with credentials

# Download specific version (here version 1)
kaggle models instances versions download google/gemma/pyTorch/2b/1
```

Method 3. Calling the API directly

```
# Authenticate with credentials
export KAGGLE_USERNAME=xyz
export KAGGLE_KEY=xyz

# With Curl
curl -L -o ~/Downloads/model.tar.gz
https://www.kaggle.com/api/v1/models/google/gemma/pyTorch/2b/1/download -
u $KAGGLE_USERNAME:$KAGGLE_KEY

# Download specific version (here version 1)
wget
https://www.kaggle.com/api/v1/models/google/gemma/pyTorch/2b/1/download -
-user=$KAGGLE_USERNAME --password=$KAGGLE_KEY --auth-no-challenge
```

Creating a Model

Kaggle has recently introduced the ability for the community to publish models to Kaggle Models. There are a few ways to accomplish this including exclusively via the UI. We recommend using a combination of `kagglehub`, our Python client library, to manage artifact creation and uploas and the UI to manage documentation and collaborative features.

Uploading using kagglehub Python client library (preferred)

See kagglehub documentation.

1. Install with `pip install kagglehub`
2. In a Python environment (e.g. Jupyter Notebook, IPython, etc.), run the below code

```
import kagglehub

# Other ways to authenticate also available:
https://github.com/Kaggle/kagglehub?tab=readme-ov-file#authenticate
kagglehub.login()

# For PyTorch framework & `2b` variation.
# Replace the framework with "jax", "other" based on which framework you
are uploading to.
kagglehub.model_upload('google/gemma/pyTorch/2b',
'path/to/local/model/files', 'Apache 2.0')

# Run the same command again to upload a new version for an existing
variation.
```

Uploading using the Kaggle CLI

See documentation. Follow steps here to authenticate with credentials.

1. Install with `!pip install kaggle`
2. In a terminal, run the below code

```
export MODEL_DIR="path/to/local/model/files"
# Go to https://www.kaggle.com/settings, download your API token file and
store it at ~/.kaggle/kaggle.json

# Create a JSON file with the metadata:
https://paste.googleplex.com/5354472119730176
vim $MODEL_DIR/model-instance-metadata.json

!kaggle models instances create -p $MODEL_DIR -r zip

# To create a new version for the instance
!kaggle models instances versions create -p $MODEL_DIR -r zip
google/gemma// -n ""
```

Upload via the UI

1. Go to: https://www.kaggle.com/models?new=true and follow the steps including setting “Creating As” to the Organization Profile you want to publish under
2. To add new Variations once your model is initially created:
    a. Scroll down to the "Model Variations" section.
    b. Click on the "New Variation" button to open the "Add/Edit" Variations modal.
    c. Select the ML framework for which you want to update weights / assets for.
    d. Click on the "Add new variation" button
    e. Select the weight / assets files to upload
    f. Enter the variation slug
        a. For example, `7b`
        b. Select a license
            a. Click on the "Create" button and wait until your instance has been fully processed.
            b. Click on "Go to model detail page".
            c. In the "Model Variations" section, you should see your variation in the drop-down.
            d. If you select it, confirm that you have all the files you were expecting under the "File Explorer" section.
            e. To upload a new version for an existing variation. Use the "New Version" button.

Documenting models

Documenting your model is easiest to do via the UI.

a. When viewing your model page, you will see a section at the top called “Pending Actions”.
b. Follow each of these steps to complete your model’s documentation:
    a. Add a description (model card)
    b. Add model instance descriptions including example code
    c. Add a subtitle
    d. Add tags
    e. Specify provenance and other metadata
    f. Publish a notebook (we recommend making it public after your model is made public)
c. Once your model is made public, you can also optionally generate a DOI from the “Metadata” section of your model.
d. Once you’re done, you can make your model public from the “Settings” tab on the model page.

e. You can now promote your model!
f. You'll be automatically subscribed to email and site notifications when any discussion topics are created