

```
pip install --extra-index-url=https://pypi.nvidia.com "cudf-cu12==24.12.*" "dask-cudf-cu12==
```



Downloading datashader-0.17.0-py3-none-any.whl (18.3 MB)

18.3/18.3 MB 92.8 MB/s eta 0:00:00

Using cached pynvml-11.4.1-py3-none-any.whl (46 kB)

Downloading scikit_image-0.24.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (14.9 MB)

14.9/14.9 MB 95.7 MB/s eta 0:00:00

Downloading dnnlib-2021.11.26-py3-none-any.whl (27 kB)

Start coding or [generate](#) with AI.

```
import tensorflow_datasets as tfds
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import numpy as np
import matplotlib.pyplot as plt

# Load the dataset
dataset, info = tfds.load("cats_vs_dogs", with_info=True, as_supervised=True)

# Get training data
train_data = dataset['train']

# Preprocessing the images
def preprocess_image(image, label):
    image = tf.image.resize(image, [64, 64]) # Resize to 64x64
    image = tf.cast(image, tf.float32) / 255.0 # Normalize to [0, 1]
    return image.numpy().flatten(), label.numpy() # Flatten the image

# Convert dataset into numpy arrays
X = []
y = []

for image, label in train_data:
    img, lbl = preprocess_image(image, label)
    X.append(img)
    y.append(lbl)

X = np.array(X)
y = np.array(y)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

from cuml.svm import SVC # GPU-accelerated SVM from cuML

# Train the GPU SVM classifier
svm_clf = SVC(kernel='rbf', C=1.0, gamma='scale')
svm_clf.fit(X_train, y_train)

# Make predictions
y_pred = svm_clf.predict(X_test)
```

```
# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy * 100:.2f}%')
```

 Accuracy: 68.54%

Start coding or [generate](#) with AI.