

IBM Cloud Model Training

Date	15 October 2022
Team ID	PNT2022TMID33204
Project Name	AI Based Food Analyzer for fitness Enthusiasts

We trained the model on a local machine and tested, now we going to train the model on IBM Cloud, for that - Go to IBM Cloud and Create a New dashboard in IBM Watson Studio and then create a Jupiter Notebook

IBM Watson Studio

Welcome, Azhagan!

Take a tutorial
Step through implementing a Data fabric use case in a sample project.

Work with data
Create a project for your team to prepare data, find insights, or build models.

Learn what's new
Stay current with new features, enhancements, and other changes.

Quick start

- Create data pipelines with DataStage
- Build customer profiles with IBM Match 360 with Watson
- Catalog and govern data with Watson Knowledge Catalog

Projects

- Food Classifier (Nov 18, 2022 02:59 PM)
- Image Classification (Nov 17, 2022 07:58 PM)

Notifications

The data cannot be displayed. The service might be restarting. Wait a few minutes and refresh the page. If the problem persists, contact your administrator.

Deployments

- fruitclassification (Today at 05:39 PM)

Projects / Food Classifier

Overview Assets Jobs Manage

Find assets

Import assets New asset +

5 assets

- All assets

Asset types

- Data (4)
- Notebooks (1)

All assets

Name	Last modified
model_training(1) Notebook	13 minutes ago Modified by you
r_124_100.jpg JPEG	2 hours ago Modified by you
ple.jpeg JPEG	2 hours ago Modified by you
data.zip application/x-zip-compressed	5 hours ago Modified by you
Dataset.rar application/octet-stream	5 hours ago Modified by you

Items per page: 20 1-5 of 5 items 1 of 1 pages

Data in this project

Drop data files here or browse for files to upload

Import Necessary Libraries and import the Datasets

The screenshot shows the IBM Cloud Training notebook interface. The notebook is titled "model_training(1)". The code in the cell is as follows:

```
In [72]: import os, types
import pandas as pd
from boto3.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
                              ibm_api_key_id='jHFXFr1UuMyf17Nb-mGYksEt0GCLoUe3gek2zne5uVL6',
                              ibm_auth_endpoint='https://iam.cloud.ibm.com/oidc/token',
                              config=Config(signature_version='oauth'),
                              endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'foodclassifier-donotdelete-pr-fr9zinqr81fncj'
object_key = 'data.zip'
```

The right sidebar shows the "Data" panel with a "Files" tab. It lists three files: "Dataset.rar", "data.zip", and "ple.jpeg". Each file has an "Insert to code" button.

Save the model as tar.gz file

```
Epoch 1/2

/tmp/wsuser/ipykernel_164/3924664455.py:1: UserWarning: `Model.fit_generator` is deprecated and will be
removed in a future version. Please use `Model.fit`, which supports generators.
  classifier.fit_generator(

526/526 [=====] - 31s 59ms/step - loss: 0.1591 - accuracy: 0.9455 - val_loss:
0.0732 - val_accuracy: 0.9725
Epoch 2/2
526/526 [=====] - 31s 58ms/step - loss: 0.0011 - accuracy: 1.0000 - val_loss:
0.0109 - val_accuracy: 1.0000

Out[83]: <keras.callbacks.History at 0x7f6139d87790>

In [84]: classifier.save('fruits.h5')

In [85]: !tar -zcvf Food-Classifier-model_new.tgz fruits.h5

fruits.h5
```

And Import the API from IBM Cloud. Create a new Deployment Space in the Watson Machine Learning Platform. For that go through the Necessary Steps and

The screenshot shows the Watson Machine Learning Platform interface. The deployment space is named "fruitclassification". The "Assets" tab is selected, showing a list of assets. The table has columns "Name" and "Last modified".

Name	Last modified
CNN Model	2 hours ago

Use the (Dallas) URL and API keys in the Program...

API keys

Create, view, and work with API keys that you have access to manage. IBM Cloud API keys are associated with a user's identity and can be used to access cloud platform and classic infrastructure APIs, depending on the access that is assigned to the user. The following table displays a list of API keys created in this account. [Learn more.](#)

Looking for more options to manage API Keys? Try [IBM Cloud® Secrets Manager](#) for creating and leasing API keys dynamically and storing them securely in your own dedicated instance.

View: My IBM Cloud API keys

API keys associated with a user's identity have the same access that the user is assigned across all accounts. To update the access for an API key, assign or remove access for the user.

Status	Name	Description	Date Created
	Newkey		2022-11-20 10:05 GMT

[Create](#) +

Using the API Key from the Access(IAM)

```
In [88]: from ibm_watson_machine_learning import APIClient
wml_credentials={
    "url" : "https://us-south.ml.cloud.ibm.com",
    "apikey":"6RuJNsiiYGeN8-aHFYUkDZHfqbh6Xcd-buHgPRoA7Fci"
}

client=APIClient(wml_credentials)

In [89]: def guid_from_space_name(client, space_name):
        space=client.spaces.get_details()
        return(next(item for item in space['resources'] if item['entity']['name']==space_name)['metadata']['id'])

In [90]: space_uid=guid_from_space_name(client, 'fruitclassification')
print("Space UID = "+space_uid)

Space UID = febef359-356f-48f8-9541-ba82916a0754

In [91]: client.set.default_space(space_uid)

Out[91]: 'SUCCESS'
```

Save the model and Download it

```
In [93]: software_spec_uid = client.software_specifications.get_uid_by_name("tensorflow_rt22.1-py3.9")

In [94]: software_spec_uid

Out[94]: 'acd9c798-6974-5d2f-a657-ce06e986df4d'

In [95]: !tar -zcvf Food-Classififer-model_new.tgz fruits.h5

          fruits.h5

In [96]: model_details=client.repository.store_model(model='Food-Classififer-model_new.tgz',meta_props={
          client.repository.ModelMetaNames.NAME:"CNN",
          client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
          client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid
          })
          model_id=client.repository.get_model_id(model_details)

In [99]: model_id

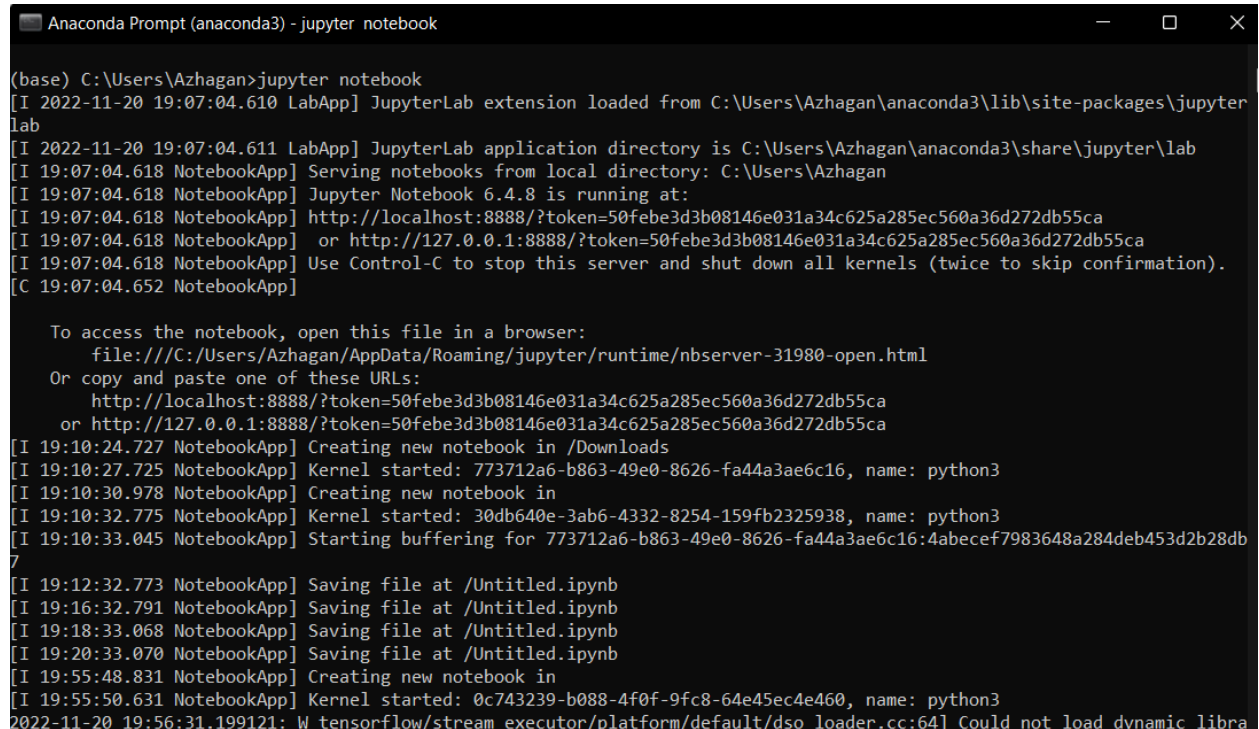
Out[99]: 'e1106521-6922-457f-ad66-ffa5a833da9c'

In [101]: client.repository.download(model_id,'my_model1.tar.gz')

          Successfully saved model content to file: 'my_model1.tar.gz'

Out[101]: '/home/wsuser/work/my_model1.tar.gz'
```

Now Go to Anaconda Prompt and open a Jupyter Notebook




```
Anaconda Prompt (anaconda3) - jupyter notebook

(base) C:\Users\Azhagan>jupyter notebook
[I 2022-11-20 19:07:04.610 LabApp] JupyterLab extension loaded from C:\Users\Azhagan\anaconda3\lib\site-packages\jupyterlab
[I 2022-11-20 19:07:04.611 LabApp] JupyterLab application directory is C:\Users\Azhagan\anaconda3\share\jupyterlab
[I 19:07:04.618 NotebookApp] Serving notebooks from local directory: C:\Users\Azhagan
[I 19:07:04.618 NotebookApp] Jupyter Notebook 6.4.8 is running at:
[I 19:07:04.618 NotebookApp] http://localhost:8888/?token=50febe3d3b08146e031a34c625a285ec560a36d272db55ca
[I 19:07:04.618 NotebookApp] or http://127.0.0.1:8888/?token=50febe3d3b08146e031a34c625a285ec560a36d272db55ca
[I 19:07:04.618 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 19:07:04.652 NotebookApp]

To access the notebook, open this file in a browser:
    file:///C:/Users/Azhagan/AppData/Roaming/jupyter/runtime/nbserver-31980-open.html
Or copy and paste one of these URLs:
    http://localhost:8888/?token=50febe3d3b08146e031a34c625a285ec560a36d272db55ca
    or http://127.0.0.1:8888/?token=50febe3d3b08146e031a34c625a285ec560a36d272db55ca
[I 19:10:24.727 NotebookApp] Creating new notebook in /Downloads
[I 19:10:27.725 NotebookApp] Kernel started: 773712a6-b863-49e0-8626-fa44a3ae6c16, name: python3
[I 19:10:30.978 NotebookApp] Creating new notebook in
[I 19:10:32.775 NotebookApp] Kernel started: 30db640e-3ab6-4332-8254-159fb2325938, name: python3
[I 19:10:33.045 NotebookApp] Starting buffering for 773712a6-b863-49e0-8626-fa44a3ae6c16:4abecef7983648a284deb453d2b28db
7
[I 19:12:32.773 NotebookApp] Saving file at /Untitled.ipynb
[I 19:16:32.791 NotebookApp] Saving file at /Untitled.ipynb
[I 19:18:33.068 NotebookApp] Saving file at /Untitled.ipynb
[I 19:20:33.070 NotebookApp] Saving file at /Untitled.ipynb
[I 19:55:48.831 NotebookApp] Creating new notebook in
[I 19:55:50.631 NotebookApp] Kernel started: 0c743239-b088-4f0f-9fc8-64e45ec4e460, name: python3
2022-11-20 19:56:31.199121: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
```

Now in Jupyter Notebook , Import the IBM Credentials and Download the model to the Local Machine as a Tar.gz file

jupyter IBM_cloud_pred Last Checkpoint: an hour ago (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Requirement already satisfied: colorama in c:\users\azhagan\anaconda3\lib\site-packages (from tqdm->watson-machine-learning-client) (0.4.4)

```
In [3]: from ibm_watson_machine_learning import APIClient
wml_credentials={
    "url" : "https://us-south.ml.cloud.ibm.com",
    "apikey":"6RuJNsiiYGeN8-aHFYUKDZHfqbh6Xcd-buHgPRoA7Fci"
}

client=APIClient(wml_credentials)
```

```
In [4]: def guid_from_space_name(client, space_name):
        space=client.spaces.get_details()
        return(next(item for item in space['resources'] if item['entity']['name']==space_name)['metadata']['id'])
```

```
In [5]: space_uid=guid_from_space_name(client, 'fruitclassification')
print("Space UID = "+space_uid)

Space UID = febef359-356f-48f8-9541-ba82916a0754
```

```
In [6]: client.set.default_space(space_uid)
```

```
Out[6]: 'SUCCESS'
```

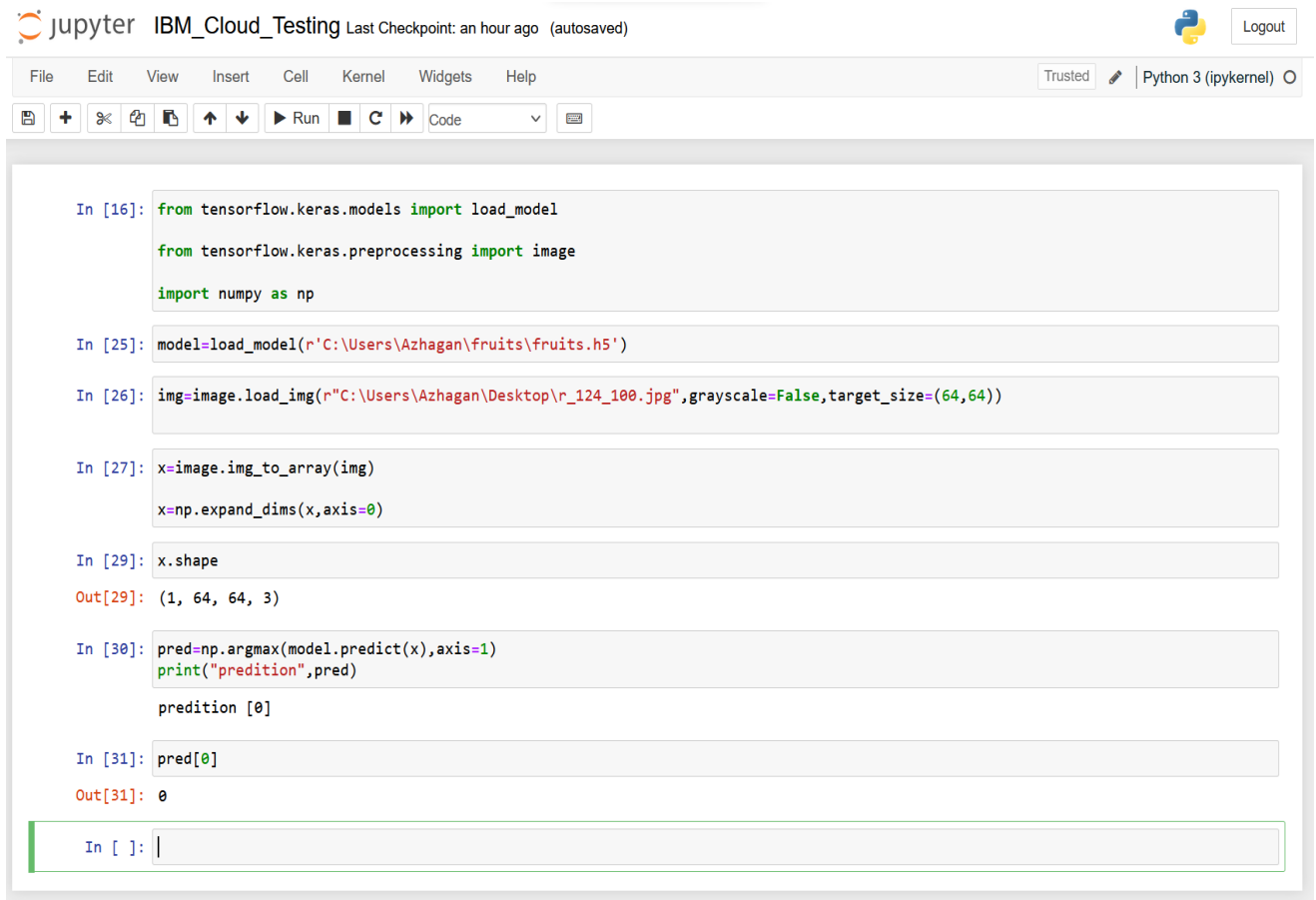
```
In [7]: client.repository.download("e1106521-6922-457f-ad66-ffa5a833da9c", 'fruits.tar.gz')

Successfully saved model content to file: 'fruits.tar.gz'
```

```
Out[7]: 'C:\\Users\\Azhagan\\fruits.tar.gz'
```

```
In [ ]:
```

As Usual Test the Downloaded IBM Cloud Trained model with some Test images



The image shows a Jupyter Notebook interface with the title "IBM_Cloud_Testing" and a status "Last Checkpoint: an hour ago (autosaved)". The interface includes a top menu bar with options like File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu is a toolbar with icons for saving, adding cells, undo, redo, and running code. The notebook contains several code cells:

```
In [16]: from tensorflow.keras.models import load_model
         from tensorflow.keras.preprocessing import image
         import numpy as np

In [25]: model=load_model(r'C:\Users\Azhagan\fruits\fruits.h5')

In [26]: img=image.load_img(r"C:\Users\Azhagan\Desktop\r_124_100.jpg",grayscale=False,target_size=(64,64))

In [27]: x=image.img_to_array(img)
         x=np.expand_dims(x,axis=0)

In [29]: x.shape
Out[29]: (1, 64, 64, 3)

In [30]: pred=np.argmax(model.predict(x),axis=1)
         print("predition",pred)
         predition [0]

In [31]: pred[0]
Out[31]: 0

In [ ]: |
```
