

Date :19 November 2022

Team ID :PNT2022TMID21496

Project Name : AI-powered Nutrition Analyzer for Fitness Enthusiasts

SPRINT 4

To deploy Model in IBM

```
In [ ]: import os, types
import pandas as pd
from botocore.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 cr
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='hf_FuQ1swprHhc8J0qVLLpVUQQUDcXsmpXdvdDqHWJ1X',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.eu.cloud-object-storage.appdomain.cloud')

bucket = 'aipowerednutritionanalyzer-donotdelete-pr-h5ft7bnq1cih7n'
object_key = 'IBM Cloud.zip'

streaming_body_2 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilit
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

```
In [36]: from io import BytesIO
import zipfile
unzip = zipfile.ZipFile(BytesIO(streaming_body_2.read()), 'r')
file_paths=unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

```
In [37]: ls
```

[Dataset/](#)

Image Preprocessing

```
In [40]: #Importing The ImageDataGenerator Library
from keras.preprocessing.image import ImageDataGenerator
```

Image Data Augmentation

```
In [41]: #Configure ImageDataGenerator Class
train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal
test_datagen=ImageDataGenerator(rescale=1./255)
```

Applying Image DataGenerator Functionality To TrainsetAnd Testset

```
In [42]: #Applying Image DataGenerator Functionality To Trainset And Testset
x_train = train_datagen.flow_from_directory(r'Dataset/TRAIN_SET/',target_size=(64, 64),batch_size=32)

Found 4118 images belonging to 5 classes.

In [43]: #Applying Image DataGenerator Functionality To Testset
x_test = test_datagen.flow_from_directory(r'Dataset/TEST_SET',target_size=(64, 64),batch_size=32)

Found 1312 images belonging to 5 classes.

In [44]: #checking the number of classes
print(x_train.class_indices)

{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON': 4}

In [45]: #checking the number of classes
print(x_test.class_indices)

{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON': 4}

In [46]: from collections import Counter as c
c(x_train.labels)

Out[46]: Counter({0: 995, 1: 1354, 2: 1019, 3: 275, 4: 475})
```

Model Building

Importing The Model Building Libraries

```
In [47]: import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dropout
```

Initializing The Model

```
In [48]: model=Sequential()
```

Adding CNN Layers

```
In [49]: # Initializing the CNN
classifier = Sequential()

In [50]: # First convolution layer and pooling
classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))

In [51]: # Second convolution layer and pooling
classifier.add(Conv2D(32, (3, 3), activation='relu'))

In [52]: # input_shape is going to be the pooled feature maps from the previous convolution layer
classifier.add(MaxPooling2D(pool_size=(2, 2)))
```

```
In [53]: # Flattening the Layers
classifier.add(Flatten())
```

Adding Dense Layers

```
In [54]: classifier.add(Dense(units=128, activation='relu'))
classifier.add(Dense(units=5, activation='softmax'))
```

```
In [55]: #summary of our model
classifier.summary()
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0
conv2d_1 (Conv2D)	(None, 29, 29, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 32)	0
flatten (Flatten)	(None, 6272)	0
dense (Dense)	(None, 128)	802944
dense_1 (Dense)	(None, 5)	645
=====		
Total params: 813,733		
Trainable params: 813,733		
Non-trainable params: 0		

Configure The Learning Process

```
In [56]: # Compiling the CNN
# categorical_crossentropy for more than 2
classifier.compile(optimizer='adam',
                  loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
                  metrics=['accuracy'])
```

Train The Model

```
In [57]: #Fitting the Model
classifier.fit_generator(generator=x_train, steps_per_epoch = len(x_train), epochs=20,
                       validation_data=(x_train))
```

/tmp/wsuser/ipykernel_164/945480658.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generator s.

```
classifier.fit_generator(generator=x_train, steps_per_epoch = len(x_train), epochs=20,
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/util/dispatch.py:
1096: UserWarning: "`sparse_categorical_crossentropy` received `from_logits=True`, but the
`output` argument was produced by a sigmoid or softmax activation and thus does not repres
ent logits. Was this intended?"
return dispatch_target(*args, **kwargs)
```

```

Epoch 1/20
824/824 [=====] - 27s 32ms/step - loss: 0.6143 - accuracy: 0.7661
- val_loss: 0.4850 - val_accuracy: 0.8193
Epoch 2/20
824/824 [=====] - 26s 32ms/step - loss: 0.4217 - accuracy: 0.8477
- val_loss: 0.3541 - val_accuracy: 0.8703
Epoch 3/20
824/824 [=====] - 26s 31ms/step - loss: 0.3961 - accuracy: 0.8473
- val_loss: 0.3525 - val_accuracy: 0.8686
Epoch 4/20
824/824 [=====] - 26s 31ms/step - loss: 0.3502 - accuracy: 0.8618
- val_loss: 0.3550 - val_accuracy: 0.8492
Epoch 5/20
824/824 [=====] - 26s 31ms/step - loss: 0.3470 - accuracy: 0.8706
- val_loss: 0.3377 - val_accuracy: 0.8662
Epoch 6/20
824/824 [=====] - 26s 32ms/step - loss: 0.3172 - accuracy: 0.8781
- val_loss: 0.2676 - val_accuracy: 0.9002
Epoch 7/20
824/824 [=====] - 25s 31ms/step - loss: 0.2990 - accuracy: 0.8866
- val_loss: 0.4165 - val_accuracy: 0.8373
Epoch 8/20
824/824 [=====] - 25s 31ms/step - loss: 0.2995 - accuracy: 0.8859
- val_loss: 0.2453 - val_accuracy: 0.9089
Epoch 9/20
824/824 [=====] - 26s 31ms/step - loss: 0.2770 - accuracy: 0.8963
- val_loss: 0.2388 - val_accuracy: 0.9143
Epoch 10/20
824/824 [=====] - 25s 31ms/step - loss: 0.2577 - accuracy: 0.9014
- val_loss: 0.2458 - val_accuracy: 0.9019
Epoch 11/20
824/824 [=====] - 26s 31ms/step - loss: 0.2498 - accuracy: 0.9029
- val_loss: 0.1894 - val_accuracy: 0.9310
Epoch 12/20
824/824 [=====] - 26s 31ms/step - loss: 0.2384 - accuracy: 0.9094
- val_loss: 0.2005 - val_accuracy: 0.9259
Epoch 13/20
824/824 [=====] - 26s 31ms/step - loss: 0.2093 - accuracy: 0.9223
- val_loss: 0.1598 - val_accuracy: 0.9417
Epoch 14/20
824/824 [=====] - 26s 31ms/step - loss: 0.2022 - accuracy: 0.9225
- val_loss: 0.1593 - val_accuracy: 0.9410
Epoch 15/20
824/824 [=====] - 26s 31ms/step - loss: 0.1880 - accuracy: 0.9308
- val_loss: 0.1552 - val_accuracy: 0.9437
Epoch 16/20
824/824 [=====] - 26s 31ms/step - loss: 0.1761 - accuracy: 0.9305
- val_loss: 0.1466 - val_accuracy: 0.9483
Epoch 17/20
824/824 [=====] - 25s 31ms/step - loss: 0.1817 - accuracy: 0.9371
- val_loss: 0.1491 - val_accuracy: 0.9444
Epoch 18/20
824/824 [=====] - 26s 31ms/step - loss: 0.1729 - accuracy: 0.9322
- val_loss: 0.1506 - val_accuracy: 0.9500
Epoch 19/20
824/824 [=====] - 26s 31ms/step - loss: 0.1378 - accuracy: 0.9473
- val_loss: 0.1685 - val_accuracy: 0.9371
Epoch 20/20
824/824 [=====] - 26s 31ms/step - loss: 0.1523 - accuracy: 0.9420
- val_loss: 0.0996 - val_accuracy: 0.9638
<keras.callbacks.History at 0x7f9dd66469d0>

```

Out[57]:

Saving The Model

```
In [63]: classifier.save('foodnutrition.h5')
```

```
In [64]: #compress File
!tar -zcvf fruit-classification-model.tgz foodnutrition.h5

foodnutrition.h5
```

IBM Deployment

```
In [61]: !pip install ibm_watson_machine_learning
```

```
Requirement already satisfied: ibm_watson_machine_learning in /opt/conda/envs/Python-3.9/1
ib/python3.9/site-packages (1.0.257)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-p
ackages (from ibm_watson_machine_learning) (0.8.9)
Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python
3.9/site-packages (from ibm_watson_machine_learning) (4.8.2)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-pac
kages (from ibm_watson_machine_learning) (0.3.3)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-pa
ckages (from ibm_watson_machine_learning) (2022.9.24)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-pa
ckages (from ibm_watson_machine_learning) (1.26.7)
Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/pytho
n3.9/site-packages (from ibm_watson_machine_learning) (2.11.0)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/pyt
hon3.9/site-packages (from ibm_watson_machine_learning) (1.3.4)
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-
packages (from ibm_watson_machine_learning) (21.3)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-p
ackages (from ibm_watson_machine_learning) (2.26.0)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/pyt
hon3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (0.10.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.
9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (2.1
1.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/
python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm_watson_machine_learning) (2.11.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/1
ib/python3.9/site-packages (from ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->ibm_watson
_machine_learning) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/si
te-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/s
ite-packages (from pandas<1.5.0,>=0.24.2->ibm_watson_machine_learning) (1.20.3)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-p
ackages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->
ibm_watson_machine_learning) (1.15.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/si
te-packages (from requests->ibm_watson_machine_learning) (3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/li
b/python3.9/site-packages (from requests->ibm_watson_machine_learning) (2.0.4)
Requirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-
packages (from importlib-metadata->ibm_watson_machine_learning) (3.6.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/
python3.9/site-packages (from packaging->ibm_watson_machine_learning) (3.0.4)
```

```
In [65]: from ibm_watson_machine_learning import APIClient
```

```
In [66]: wml_credentials = {
    "url": "https://eu-de.ml.cloud.ibm.com",
    "apikey": "H9IXE32x2IZ4QPw1C5enIOYL79ZJ-Bx5j7z2acJBcgwa"
}
```

11/19/22, 11:34 PM

IBM_AI_Powered_Nutrition_Analyzer

In [67]: client = APIClient(wml_credentials)

In [69]: client.spaces.list()

Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50

ID	NAME	CREATED
586f7344-73c3-4780-b378-bc31fe7e902c	Fruit_Classifier	2022-11-19T16:32:48.301Z

In [89]: space_id = "586f7344-73c3-4780-b378-bc31fe7e902c"

In [90]: client.set.default_space(space_id)

Out[90]: 'SUCCESS'

In [91]: client.software_specifications.list()

NAME	ASSET_ID	TYPE
default_py3.6	0062b8c9-8b7d-44a0-a9b9-46c416adcbd9	base
kernel-spark3.2-scala2.12	020d69ce-7ac1-5e68-ac1a-31189867356a	base
pytorch-onnx_1.3-py3.7-edt	069ea134-3346-5748-b513-49120e15d288	base
scikit-learn_0.20-py3.6	09c5a1d0-9c1e-4473-a344-eb7b665ff687	base
spark-mllib_3.0-scala_2.12	09f4cff0-90a7-5899-b9ed-1ef348aebdee	base
pytorch-onnx_rt22.1-py3.9	0b848dd4-e681-5599-be41-b5f6fccc6471	base
ai-function_0.1-py3.6	0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda	base
shiny-r3.6	0e6e79df-875e-4f24-8ae9-62dcc2148306	base
tensorflow_2.4-py3.7-horovod	1092590a-307d-563d-9b62-4eb7d64b3f22	base
pytorch_1.1-py3.6	10ac12d6-6b30-4ccd-8392-3e922c096a92	base
tensorflow_1.15-py3.6-ddl	111e41b3-de2d-5422-a4d6-bf776828c4b7	base
autoai-kb_rt22.2-py3.10	125b6d9a-5b1f-5e8d-972a-b251688ccf40	base
runtime-22.1-py3.9	12b83a17-24d8-5082-900f-0ab31fbfd3cb	base
scikit-learn_0.22-py3.6	154010fa-5b3b-4ac1-82af-4d5ee5abbc85	base
default_r3.6	1b70aec3-ab34-4b87-8aa0-a4a3c8296a36	base
pytorch-onnx_1.3-py3.6	1bc6029a-cc97-56da-b8e0-39c3880dbbe7	base
kernel-spark3.3-r3.6	1c9e5454-f216-59dd-a20e-474a5cdf5988	base
pytorch-onnx_rt22.1-py3.9-edt	1d362186-7ad5-5b59-8b6c-9d0880bde37f	base
tensorflow_2.1-py3.6	1eb25b84-d6ed-5dde-b6a5-3fbdf1665666	base
spark-mllib_3.2	20047f72-0a98-58c7-9ff5-a77b012eb8f5	base
tensorflow_2.4-py3.8-horovod	217c16f6-178f-56bf-824a-b19f20564c49	base
runtime-22.1-py3.9-cuda	26215f05-08c3-5a41-a1b0-da66306ce658	base
do_py3.8	295addb5-9ef9-547e-9bf4-92ae3563e720	base
autoai-ts_3.8-py3.8	2aa0c932-798f-5ae9-abd6-15e0c2402fb5	base
tensorflow_1.15-py3.6	2b73a275-7cbf-420b-a912-eae7f436e0bc	base
kernel-spark3.3-py3.9	2b7961e2-e3b1-5a8c-a491-482c8368839a	base
pytorch_1.2-py3.6	2c8ef57d-2687-4b7d-acce-01f94976dac1	base
spark-mllib_2.3	2e51f700-bca0-4b0d-88dc-5c6791338875	base
pytorch-onnx_1.1-py3.6-edt	32983cea-3f32-4400-8965-dde874a8d67e	base
spark-mllib_3.0-py37	36507ebe-8770-55ba-ab2a-eafe787600e9	base
spark-mllib_2.4	390d21f8-e58b-4fac-9c55-d7ceda621326	base
autoai-ts_rt22.2-py3.10	396b2e83-0953-5b86-9a55-7ce1628a406f	base
xgboost_0.82-py3.6	39e31acd-5f30-41dc-ae44-60233c80306e	base
pytorch-onnx_1.2-py3.6-edt	40589d0e-7019-4e28-8daa-fb03b6f4fe12	base
pytorch-onnx_rt22.2-py3.10	40e73f55-783a-5535-b3fa-0c8b94291431	base
default_r36py38	41c247d3-45f8-5a71-b065-8580229facf0	base
autoai-ts_rt22.1-py3.9	4269d26e-07ba-5d40-8f66-2d495b0c71f7	base
autoai-obm_3.0	42b92e18-d9ab-567f-988a-4240ba1ed5f7	base
pmml-3.0_4.3	493bcb95-16f1-5bc5-bee8-81b8af80e9c7	base
spark-mllib_2.4-r_3.6	49403dff-92e9-4c87-a3d7-a42d0021c095	base
xgboost_0.90-py3.6	4ff8d6c2-1343-4c18-85e1-689c965304d3	base
pytorch-onnx_1.1-py3.6	50f95b2a-bc16-43bb-bc94-b0bed208c60b	base
autoai-ts_3.9-py3.8	52c57136-80fa-572e-8728-a5e7cbb42cde	base
spark-mllib_2.4-scala_2.11	55a70f99-7320-4be5-9fb9-9edb5a443af5	base
spark-mllib_3.0	5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9	base
autoai-obm_2.0	5c2e37fa-80b8-5e77-840f-d912469614ee	base
spss-modeler_18.1	5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b	base
cuda-py3.8	5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e	base
runtime-22.2-py3.10-xc	5e8cddff-db4a-5a6a-b8aa-2d4af9864dab	base
autoai-kb_3.1-py3.7	632d4b22-10aa-5180-88f0-f52dfb6444d7	base

Note: Only first 50 records were displayed. To display more use 'limit' parameter.

In [92]: `software_space_uid = client.software_specifications.get_id_by_name("tensorflow_rt22.1-py3.9")`

In [93]: `software_space_uid`

Out[93]: `'acd9c798-6974-5d2f-a657-ce06e986df4d'`

In [102]: `model_details = client.repository.store_model(model="fruit-classification-model.tgz", meta_data={client.repository.ModelMetaNames.NAME: "cnn model", client.repository.ModelMetaNames.TYPE: "tensorflow_2.7",`

```
client.repository.ModelMetaNames.SOFTWARE_SPEC_UID : software_space_uid  
}))
```

```
In [103... model_id = client.repository.get_model_id(model_details)
```

```
In [104... model_id
```

```
Out[104]: '45aa8f8f-7856-4ea8-b129-472cf649e8d9'
```