

Train The Model On IBM

Team ID	PNT2022TMID30319
Project Name	Fertilizers Recommendation System For Disease Prediction

In IBM cloud using IBM watson machine learning service to train the both fruit and vegetable.

Fruit model :

```
← → ↺ dataplatform.cloud.ibm.com/analytics/notebooks/v2/6c7fe95e-7e7a-44aa-9248-abad30e0e238/view?projectid=469ab094-92a3-4910-9ff3-ac6fef56a726...
Gmail YouTube Maps News

IBM Watson Studio Search in your workspaces Buy Vignesh G's Account Dallas VG

Projects / FruitSet / Fruit_Set

In [5]: 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 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='y-hhKQEmvOomBDJAbK5YgepYBH1u5Fyvh-SluVlxDD2K',
                              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 = 'fruitset-donotdelete-pr-8fqk5yezvb2gvm'
object_key = 'Fertilizers_Recommendation_System_For_Disease_Prediction.zip'

streaming_body_1 = 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 possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/

In [6]: from io import BytesIO
import zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_1.read()), 'r')
file_paths=unzip.namelist()
```

```
← → ↺ dataplatform.cloud.ibm.com/analytics/notebooks/v2/6c7fe95e-7e7a-44aa-9248-abad30e0e238/view?projectid=469ab094-92a3-4910-9ff3-ac6fef56a726...
Gmail YouTube Maps News

IBM Watson Studio Search in your workspaces Buy Vignesh G's Account Dallas VG

Projects / FruitSet / Fruit_Set

model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())

32*(3*3*3+1)
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(6,activation='softmax'))
model.summary()

Model: "sequential"
Layer (type) Output Shape Param #
-----
conv2d (Conv2D) (None, 126, 126, 32) 896
max_pooling2d (MaxPooling2D) (None, 63, 63, 32) 0
flatten (Flatten) (None, 127008) 0
dense (Dense) (None, 300) 38102700
dense_1 (Dense) (None, 150) 45150
dense_2 (Dense) (None, 6) 906
-----
Total params: 38,149,652
Trainable params: 38,149,652
Non-trainable params: 0
```

```
← → ↻ dataplatform.cloud.ibm.com/analytics/notebooks/v2/6c7fe95e-7e7a-44aa-9248-abad30e0e238/view?projectid=469ab094-92a3-4910-9ff3-ac6fef56a726...
Gmail YouTube Maps News

IBM Watson Studio Search in your workspaces Buy Vignesh G's Account Dallas VG

Projects / FruitSet / Fruit_Set

cos_client = ibm_boto3.client(service_name='s3',
                              ibm_api_key_id='y-hmKQEmvOoWBDJAbK5YgepY8H1u5FyvH-S1uV1x0D02K',
                              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 = 'fruitset-donotdelete-pr-8fqk5yezvb2gvm'
object_key = '01a8cc9f-074a-4866-87c8-bb5a9e3895b4__RS_LB_2968.JPG'

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 possibilities to Load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/

In [87]: from PIL import Image
img=Image.open(streaming_body_2)
target_size=(128,128)
img=img.resize(target_size)
import numpy as np
x=img.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred=model.predict(x)
y=np.argmax(pred,axis=1)
index=['Apple__Black_rot','Apple__healthy','Corn_(maize)__Northern_Leaf_Blight','Corn_(maize)__healthy','Peach__Bacterial_spot','Peach__healthy']
index[y[0]]

Out[87]: 'Apple__healthy'
```

Vegetable Model :

```
← → ↻ dataplatform.cloud.ibm.com/analytics/notebooks/v2/0df928-5c67-4ebd-8e12-c4a4b675c180/view?projectid=469ab094-92a3-4910-9ff3-ac6fef56a726...
Gmail YouTube Maps News

IBM Watson Studio Search in your workspaces Buy Vignesh G's Account Dallas VG

Projects / FruitSet / Vegetable_Set

In [24]: import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return @

# @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='y-hmKQEmvOoWBDJAbK5YgepY8H1u5FyvH-S1uV1x0D02K',
                              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 = 'fruitset-donotdelete-pr-8fqk5yezvb2gvm'
object_key = 'Fertilizers_Recommendation_System_For_Disease_Prediction.zip'

streaming_body_1 = 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 possibilities to Load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/

In [25]: from io import BytesIO
import zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_1.read()),'r')
file_paths=unzip.namelist()
for path in file_paths:
```

```
← → ↻ dataplatform.cloud.ibm.com/analytics/notebooks/v2/0fdff928-5c67-4ebd-8e12-c4a4b675c180/view?projectid=469ab094-92a3-4910-9ff3-ac6fef56a726...
Gmail YouTube Maps News

IBM Watson Studio Search in your workspaces Buy ⓘ ⓘ Vignesh G's Account Dallas VG

Projects / FruitSet / Vegetable_Set

In [36]: model.summary()
Model: "sequential"

Layer (type) Output Shape Param #
-----
conv2d (Conv2D) (None, 126, 126, 32) 896

max_pooling2d (MaxPooling2D) (None, 63, 63, 32) 0

conv2d_1 (Conv2D) (None, 61, 61, 32) 9248

max_pooling2d_1 (MaxPooling2D) (None, 30, 30, 32) 0

flatten (Flatten) (None, 28800) 0

dense (Dense) (None, 300) 8640300

dense_1 (Dense) (None, 150) 45150

dense_2 (Dense) (None, 9) 1359

-----
Total params: 8,696,953
Trainable params: 8,696,953
Non-trainable params: 0
```

```
← → ↻ dataplatform.cloud.ibm.com/analytics/notebooks/v2/0fdff928-5c67-4ebd-8e12-c4a4b675c180/view?projectid=469ab094-92a3-4910-9ff3-ac6fef56a726...
Gmail YouTube Maps News

IBM Watson Studio Search in your workspaces Buy ⓘ ⓘ Vignesh G's Account Dallas VG

Projects / FruitSet / Vegetable_Set

cos_client = ibm_boto3.client(service_name='s3',
                              ibm_api_key_id='y-hmKQEmvOoHBDJAbK5YgepY8H1u5fyvh-SluVlx0D2K',
                              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 = 'fruitset-donotdelete-pr-8fgk5yezvb2gvm'
object_key = '01a8cc9f-074a-4866-87c8-bb5a9e3895b4__RS_LB_2968.JPG'

streaming_body_3 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

In [165]: img=Image.open(streaming_body_3)
          target_size=(128,128)
          img=img.resize(target_size)

In [166]: import numpy as np
          x=img.img_to_array(img)
          x=np.expand_dims(x,axis=0)

In [167]: pred=model.predict(x)
          y=np.argmax(pred,axis=1)
          y[0]

Out[167]: 6

In [168]: index=['Pepper_bell_Bacterial_spot','Pepper_bell_healthy','Potato_Early_blight','Potato_Late_blight','Potato_healthy','Tomato_Bacterial_spot','Tomato_Late_blight']
          index[y[0]]

Out[168]: 'Tomato_Late_blight'
```