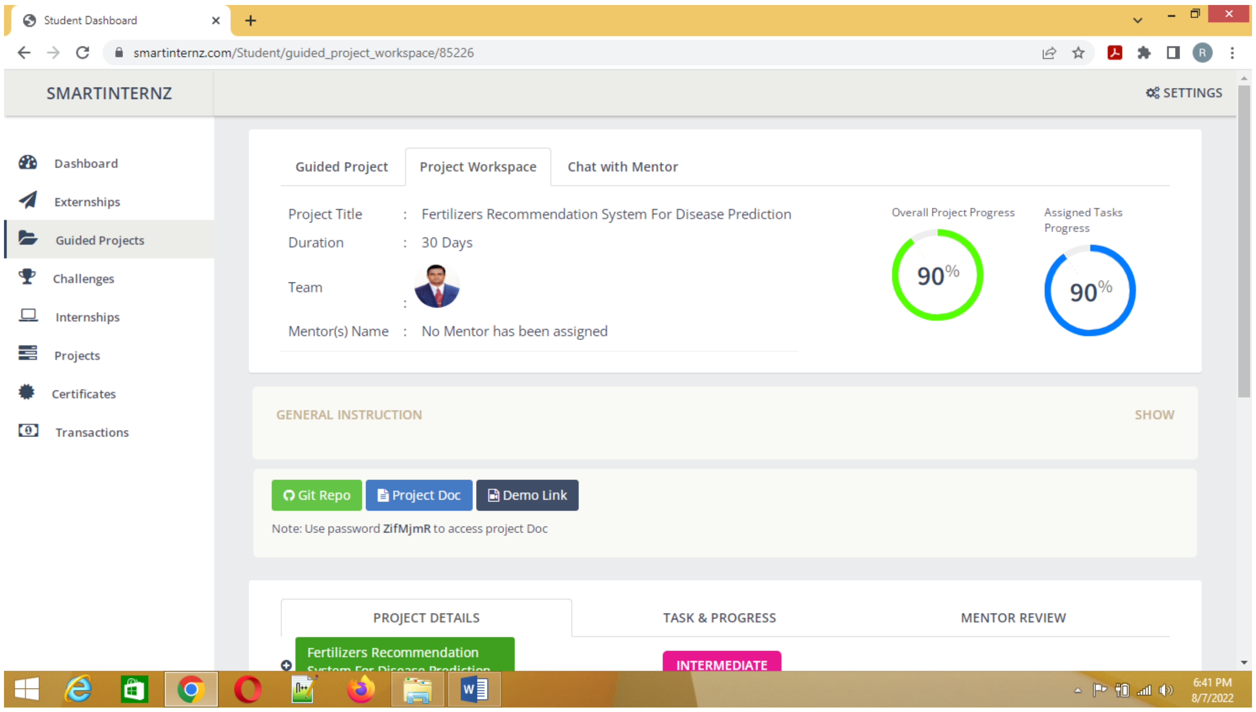
Fertilizers Recommendation system for Disease Prediction
Screen shots

1. Project workspace



SMARTINTERNZ

Guided Project | Project Workspace | Chat with Mentor

Project Title : Fertilizers Recommendation System For Disease Prediction
Duration : 30 Days
Team :
Mentor(s) Name : No Mentor has been assigned

Overall Project Progress 90%
Assigned Tasks Progress 90%

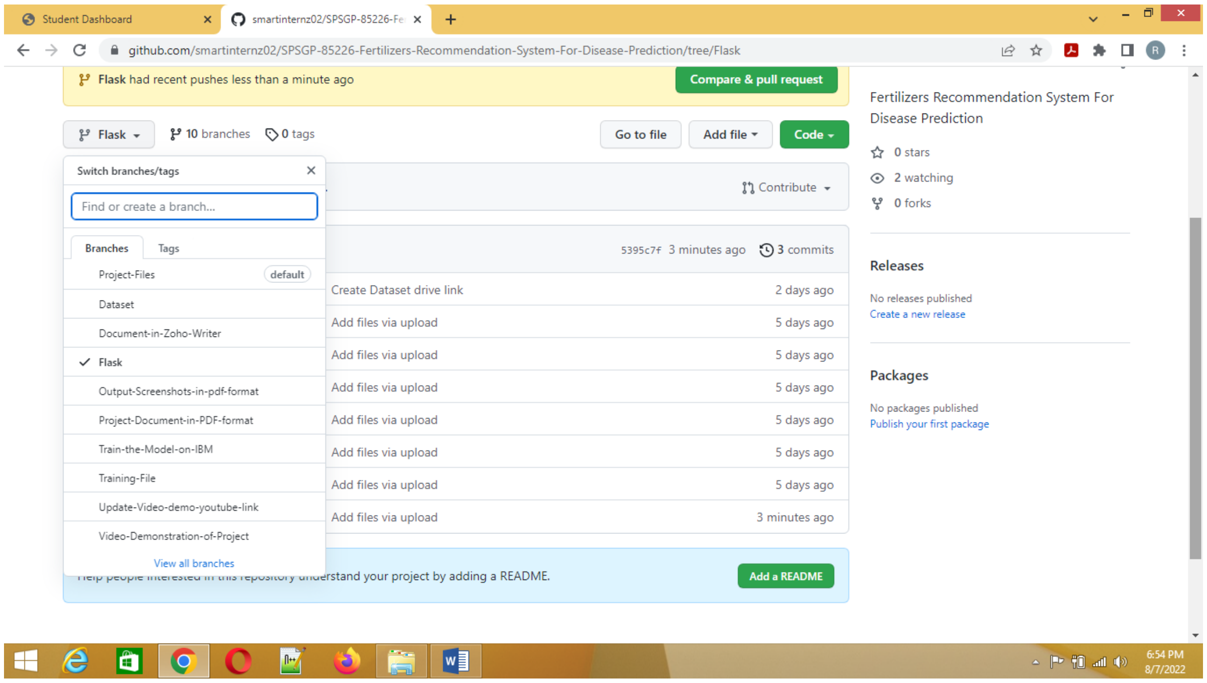
GENERAL INSTRUCTION SHOW

Git Repo | Project Doc | Demo Link

Note: Use password ZifMjmR to access project Doc

PROJECT DETAILS | TASK & PROGRESS | MENTOR REVIEW

2. GitHub repository branches



github.com/smartinternz02/SPSGP-85226-Fertilizers-Recommendation-System-For-Disease-Prediction/tree/Flask

Flask had recent pushes less than a minute ago

Flask | 10 branches | 0 tags

Switch branches/tags

Branches: Project-Files (default), Dataset, Document-in-Zoho-Writer, ✓ Flask, Output-Screenshots-in-pdf-format, Project-Document-in-PDF-format, Train-the-Model-on-IBM, Training-File, Update-Video-demo-youtube-link, Video-Demonstration-of-Project

Create Dataset drive link	2 days ago
Add files via upload	5 days ago
Add files via upload	5 days ago
Add files via upload	5 days ago
Add files via upload	5 days ago
Add files via upload	5 days ago
Add files via upload	5 days ago
Add files via upload	3 minutes ago

5395c7f 3 minutes ago 3 commits

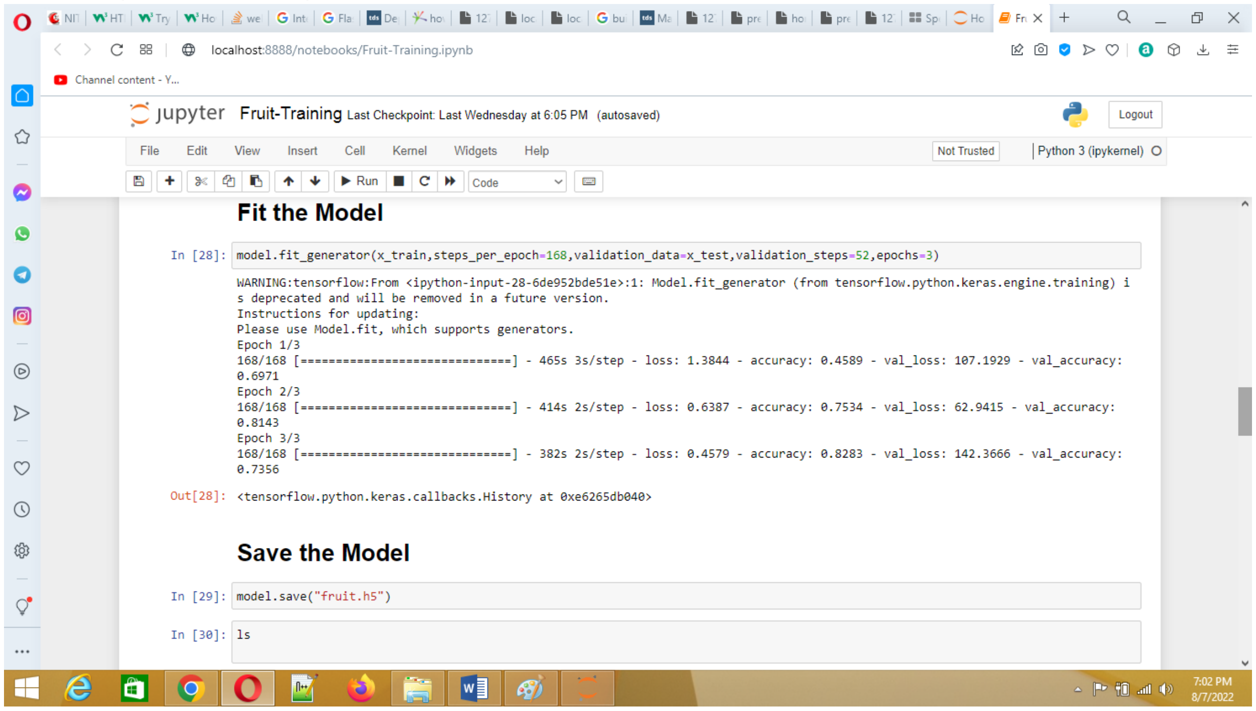
Fertilizers Recommendation System For Disease Prediction
0 stars
2 watching
0 forks

Releases
No releases published

Packages
No packages published

3. Jupyter notebook output

Training and Testing of Fruit dataset

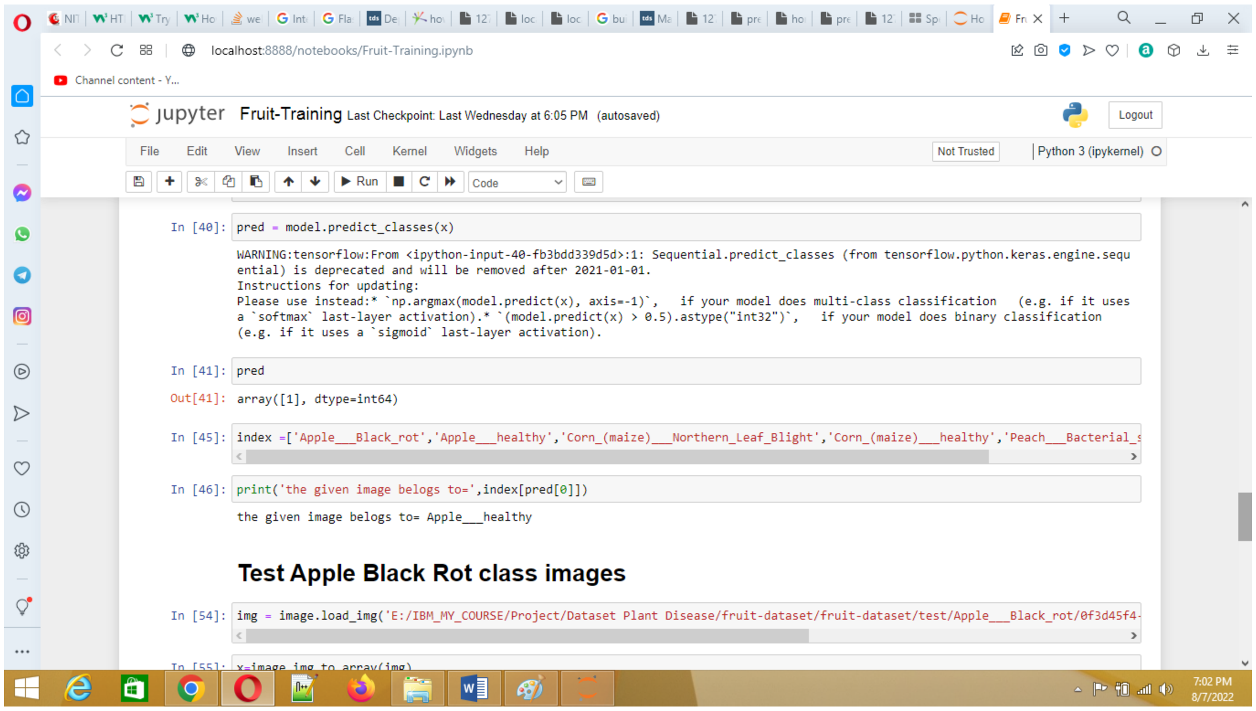

localhost:8888/notebooks/Fruit-Training.ipynb
jupyter Fruit-Training Last Checkpoint: Last Wednesday at 6:05 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
7:02 PM
8/7/2022

Fit the Model

```
In [28]: model.fit_generator(x_train,steps_per_epoch=168,validation_data=x_test,validation_steps=52,epochs=3)

         WARNING:tensorflow:From <ipython-input-28-6de952bde51e>:1: Model.fit_generator (from tensorflow.python.keras.engine.training) i
         s deprecated and will be removed in a future version.
         Instructions for updating:
         Please use Model.fit, which supports generators.
         Epoch 1/3
         168/168 [==============================] - 465s 3s/step - loss: 1.3844 - accuracy: 0.4589 - val_loss: 107.1929 - val_accuracy:
         0.6971
         Epoch 2/3
         168/168 [==============================] - 414s 2s/step - loss: 0.6387 - accuracy: 0.7534 - val_loss: 62.9415 - val_accuracy:
         0.8143
         Epoch 3/3
         168/168 [==============================] - 382s 2s/step - loss: 0.4579 - accuracy: 0.8283 - val_loss: 142.3666 - val_accuracy:
         0.7356

Out[28]: <tensorflow.python.keras.callbacks.History at 0xe6265db040>
```

Save the Model

```
In [29]: model.save("fruit.h5")

In [30]: ls
```

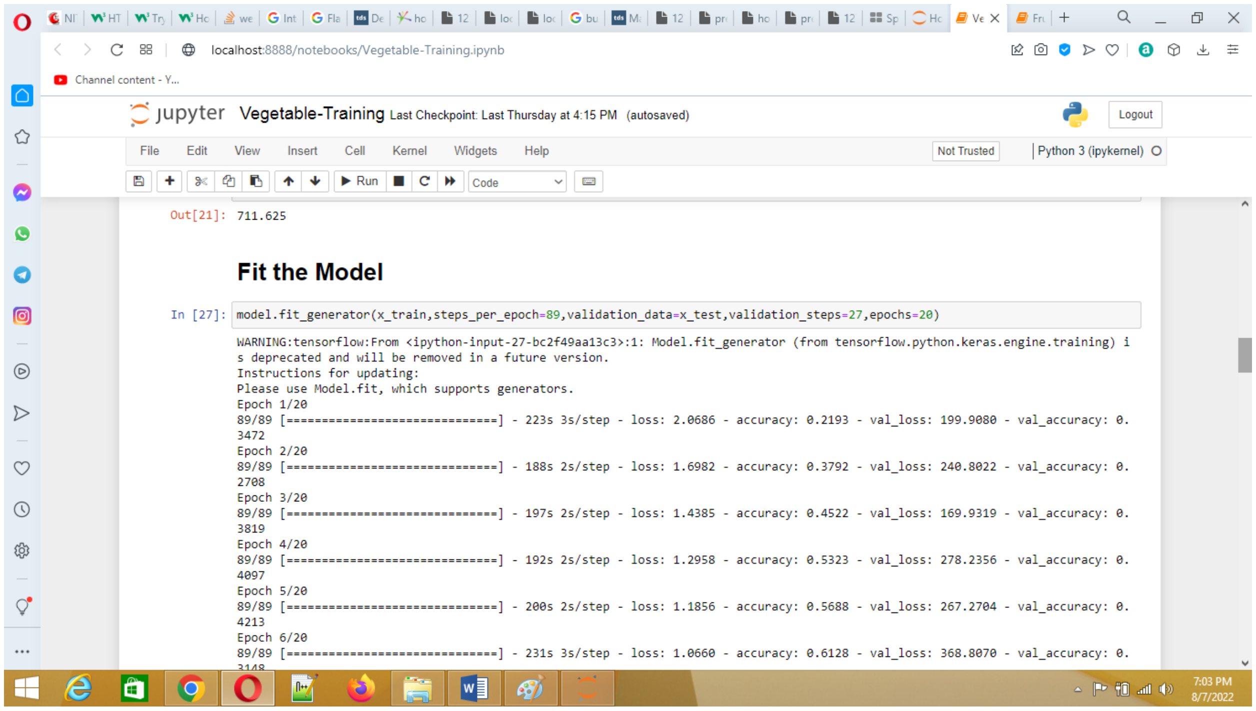

localhost:8888/notebooks/Fruit-Training.ipynb
jupyter Fruit-Training Last Checkpoint: Last Wednesday at 6:05 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
7:02 PM
8/7/2022

```
In [40]: pred = model.predict_classes(x)

         WARNING:tensorflow:From <ipython-input-40-fb3bdd339d5d>:1: Sequential.predict_classes (from tensorflow.python.keras.engine.sequ
         ential) is deprecated and will be removed after 2021-01-01.
         Instructions for updating:
         Please use instead:* `np.argmax(model.predict(x), axis=-1)`,   if your model does multi-class classification   (e.g. if it uses
         a `softmax` last-layer activation).* `(model.predict(x) > 0.5).astype("int32")`,   if your model does binary classification
         (e.g. if it uses a `sigmoid` last-layer activation).

In [41]: pred

Out[41]: array([1], dtype=int64)

In [45]: index =['Apple___Black_rot','Apple___healthy','Corn_(maize)___Northern_Leaf_Blight','Corn_(maize)___healthy','Peach___Bacterial_s

In [46]: print('the given image belogs to=',index[pred[0]])

         the given image belogs to= Apple___healthy
```

Test Apple Black Rot class images

```
In [54]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Apple___Black_rot/0f3d45f4-

In [55]: x=image.img_to_array(img)
```

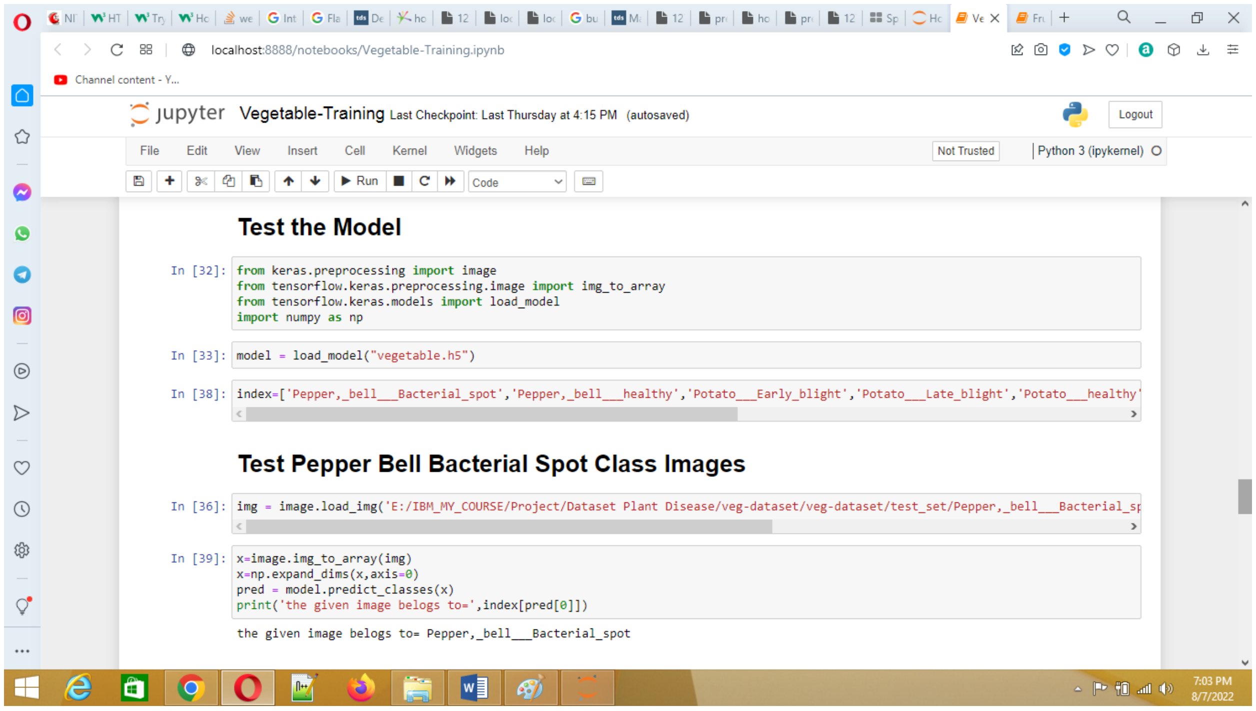
Train and Test Vegetable dataset


localhost:8888/notebooks/Vegetable-Training.ipynb
Channel content - Y...
jupyter Vegetable-Training Last Checkpoint: Last Thursday at 4:15 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
7:03 PM
8/7/2022

```
Out[21]: 711.625
```

Fit the Model

```
In [27]: model.fit_generator(x_train,steps_per_epoch=89,validation_data=x_test,validation_steps=27,epochs=20)

WARNING:tensorflow:From <ipython-input-27-bc2f49aa13c3>:1: Model.fit_generator (from tensorflow.python.keras.engine.training) i
s deprecated and will be removed in a future version.
Instructions for updating:
Please use Model.fit, which supports generators.
Epoch 1/20
89/89 [==============================] - 223s 3s/step - loss: 2.0686 - accuracy: 0.2193 - val_loss: 199.9080 - val_accuracy: 0.
3472
Epoch 2/20
89/89 [==============================] - 188s 2s/step - loss: 1.6982 - accuracy: 0.3792 - val_loss: 240.8022 - val_accuracy: 0.
2708
Epoch 3/20
89/89 [==============================] - 197s 2s/step - loss: 1.4385 - accuracy: 0.4522 - val_loss: 169.9319 - val_accuracy: 0.
3819
Epoch 4/20
89/89 [==============================] - 192s 2s/step - loss: 1.2958 - accuracy: 0.5323 - val_loss: 278.2356 - val_accuracy: 0.
4097
Epoch 5/20
89/89 [==============================] - 200s 2s/step - loss: 1.1856 - accuracy: 0.5688 - val_loss: 267.2704 - val_accuracy: 0.
4213
Epoch 6/20
89/89 [==============================] - 231s 3s/step - loss: 1.0660 - accuracy: 0.6128 - val_loss: 368.8070 - val_accuracy: 0.
3148
```


localhost:8888/notebooks/Vegetable-Training.ipynb
Channel content - Y...
jupyter Vegetable-Training Last Checkpoint: Last Thursday at 4:15 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
7:03 PM
8/7/2022

Test the Model

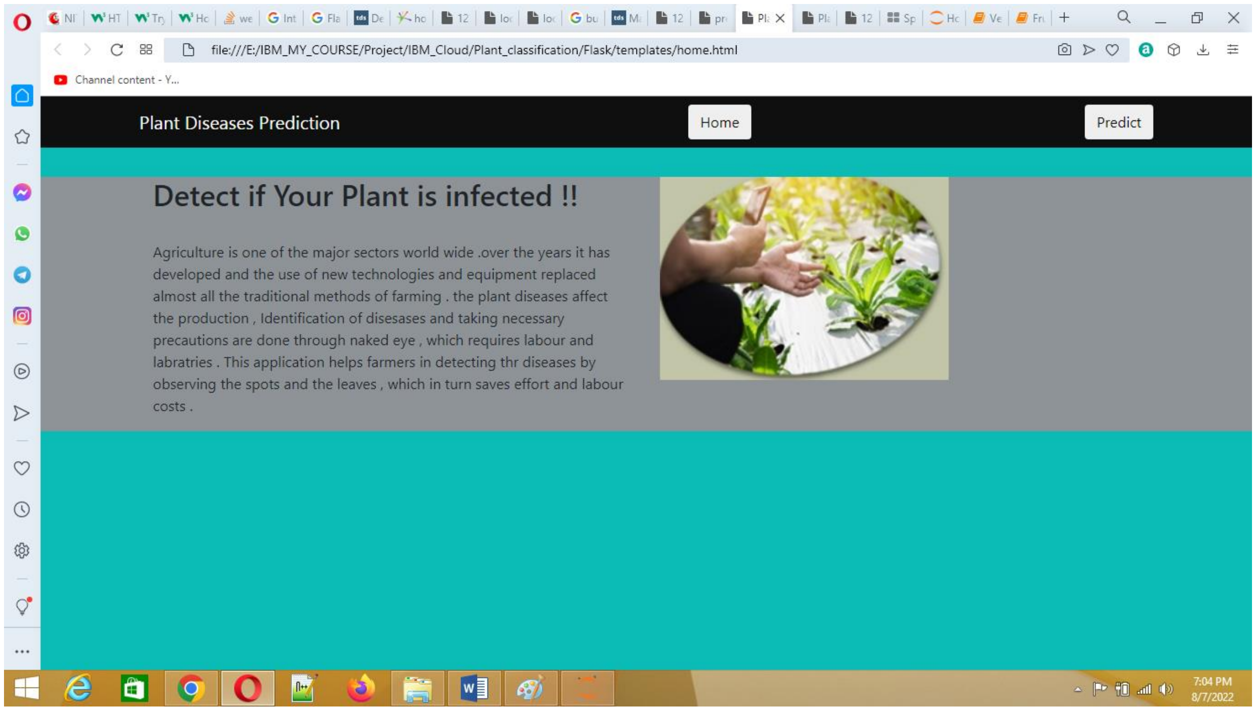
```
In [32]: from keras.preprocessing import image
         from tensorflow.keras.preprocessing.image import img_to_array
         from tensorflow.keras.models import load_model
         import numpy as np

In [33]: model = load_model("vegetable.h5")

In [38]: index=['Pepper,_bell___Bacterial_spot','Pepper,_bell___healthy','Potato___Early_blight','Potato___Late_blight','Potato___healthy'
```

Test Pepper Bell Bacterial Spot Class Images

```
In [36]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/veg-dataset/veg-dataset/test_set/Pepper,_bell___Bacterial_sp

In [39]: x=image.img_to_array(img)
         x=np.expand_dims(x,axis=0)
         pred = model.predict_classes(x)
         print('the given image belogs to=',index[pred[0]])

         the given image belogs to= Pepper,_bell___Bacterial_spot
```

4. Flask web deployment

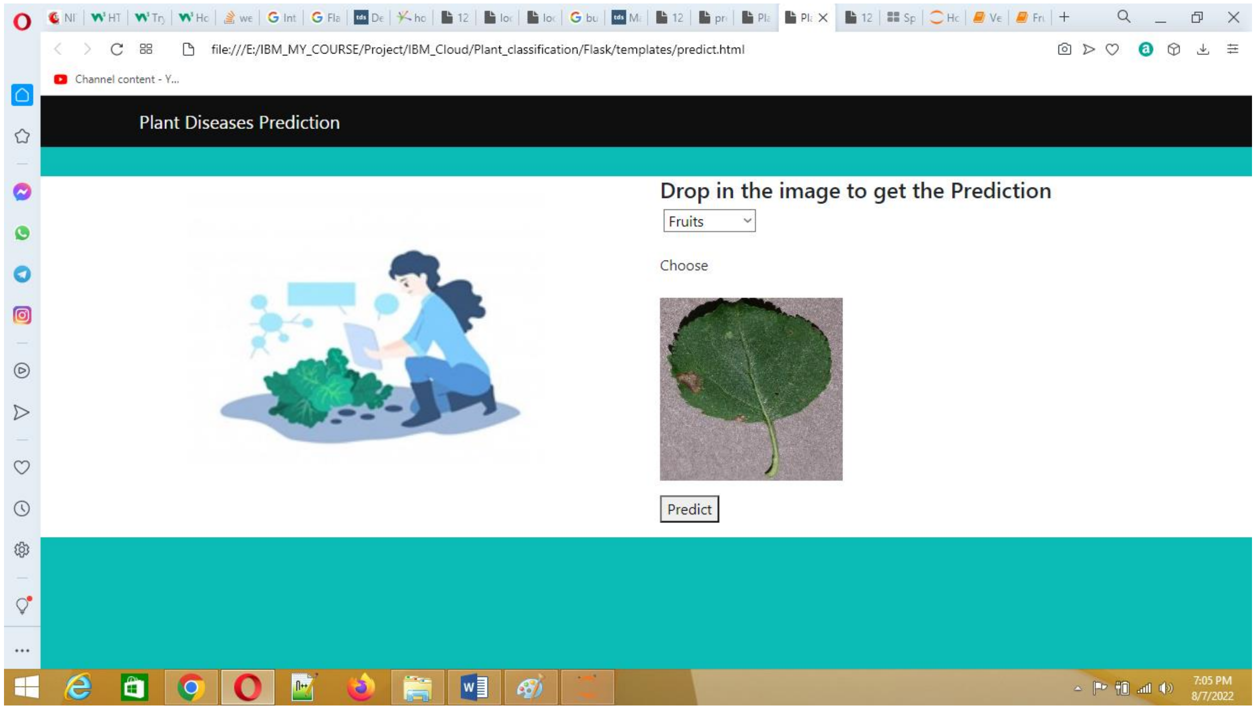
Home.html


file:///E:/IBM_MY_COURSE/Project/IBM_Cloud/Plant_classification/Flask/templates/home.html
Plant Diseases Prediction
Home
Predict

Detect if Your Plant is infected !!

Agriculture is one of the major sectors world wide .over the years it has developed and the use of new technologies and equipment replaced almost all the traditional methods of farming . the plant diseases affect the production , Identification of disesases and taking necessary precautions are done through naked eye , which requires labour and labratries . This application helps farmers in detecting thr diseases by observing the spots and the leaves , which in turn saves effort and labour costs .

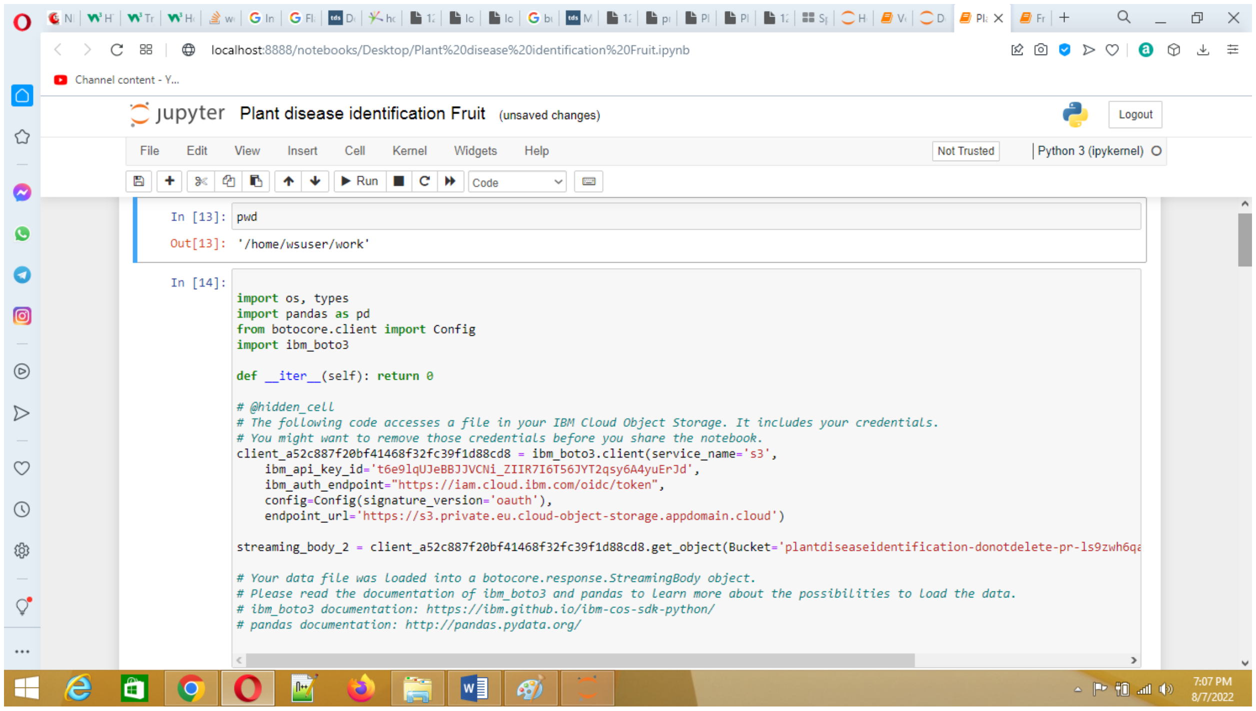
Predict.html


file:///E:/IBM_MY_COURSE/Project/IBM_Cloud/Plant_classification/Flask/templates/predict.html
Plant Diseases Prediction
Drop in the image to get the Prediction
Fruits
Choose
Predict

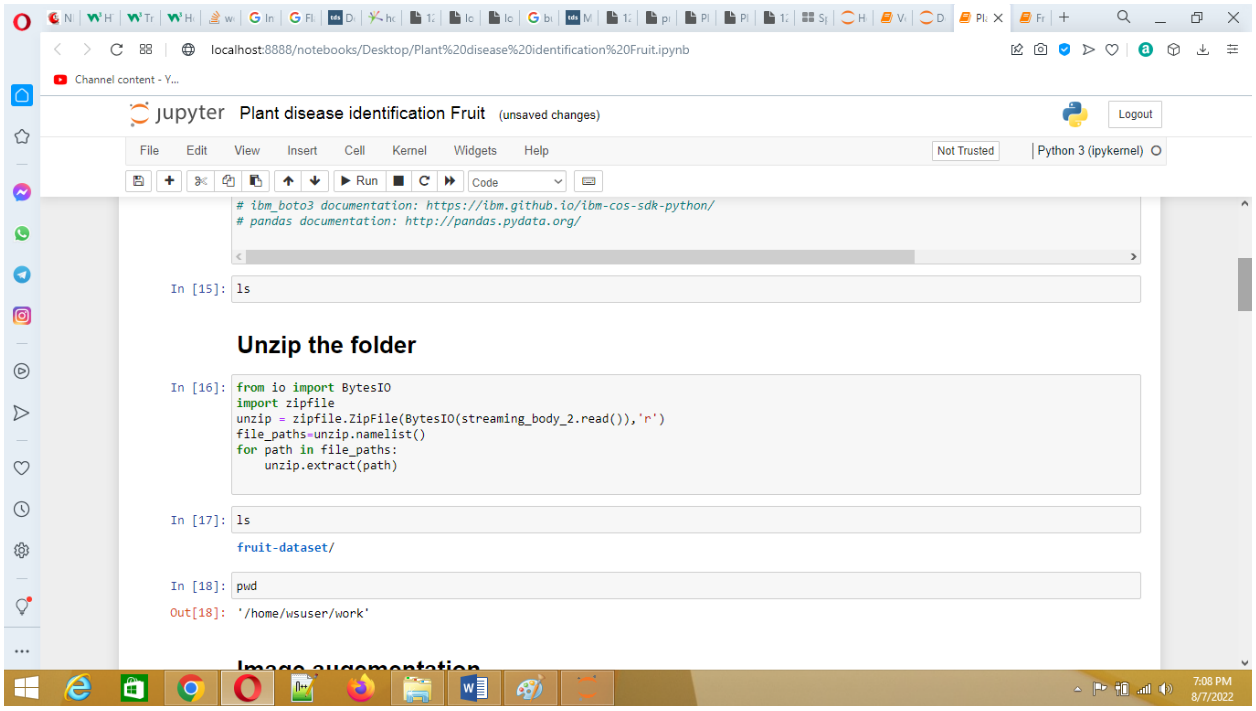
5. IBM Cloud training

Due to CUH limit exceeds, I have downloaded the notebooks and opened in Jupyter notebook

(i). Fruit dataset:


localhost:8888/notebooks/Desktop/Plant%20disease%20identification%20Fruit.ipynb
jupyter Plant disease identification Fruit (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code

```
In [13]: pwd
Out[13]: '/home/wsuser/work'

In [14]:
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.
client_a52c887f20bf41468f32fc39f1d88cd8 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='t6e9lqUJeBBJJVCNi_ZIIR7I6T56JYT2qsy6A4yuErJd',
    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')

streaming_body_2 = client_a52c887f20bf41468f32fc39f1d88cd8.get_object(Bucket='plantdiseaseidentification-donotdelete-pr-ls9zwh6qa

# 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/
```

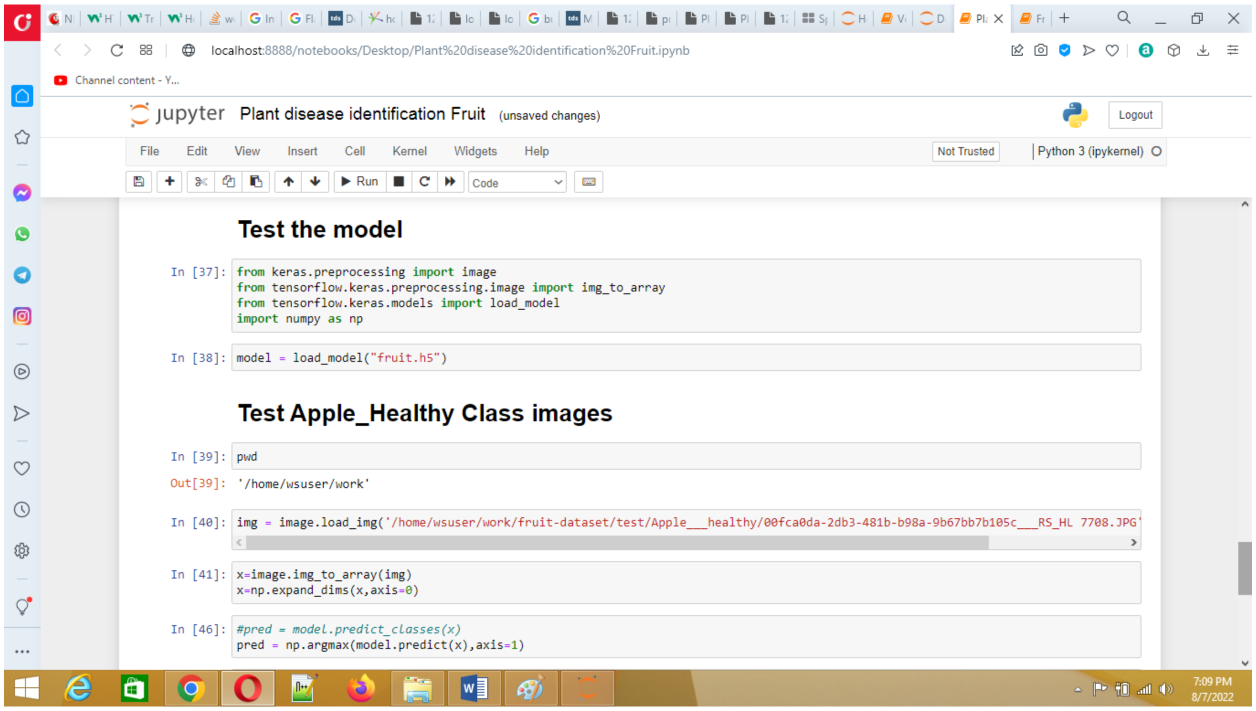

localhost:8888/notebooks/Desktop/Plant%20disease%20identification%20Fruit.ipynb
jupyter Plant disease identification Fruit (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code

```
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/

In [15]: ls
```

Unzip the folder

```
In [16]: 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 [17]: ls
         fruit-dataset/

In [18]: pwd
Out[18]: '/home/wsuser/work'
```

Image augmentation



localhost:8888/notebooks/Desktop/Plant%20disease%20identification%20Fruit.ipynb

jupyter Plant disease identification Fruit (unsaved changes)

Not Trusted | Python 3 (ipykernel)

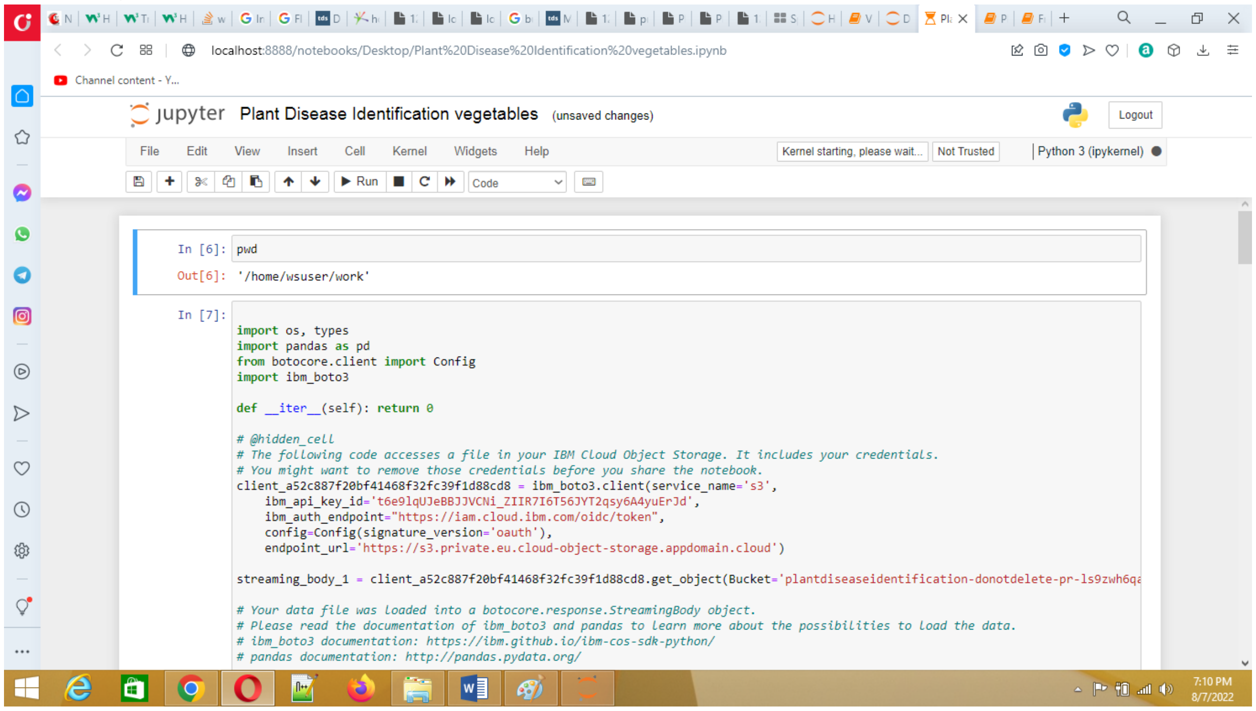
Test the model

```
In [37]: from keras.preprocessing import image
         from tensorflow.keras.preprocessing.image import img_to_array
         from tensorflow.keras.models import load_model
         import numpy as np

In [38]: model = load_model("fruit.h5")
```

Test Apple_Healthy Class images

```
In [39]: pwd

Out[39]: '/home/wsuser/work'

In [40]: img = image.load_img('/home/wsuser/work/fruit-dataset/test/Apple___healthy/00fca0da-2db3-481b-b98a-9b67bb7b105c___RS_HL 7708.JPG'

In [41]: x=image.img_to_array(img)
         x=np.expand_dims(x,axis=0)

In [46]: #pred = model.predict_classes(x)
         pred = np.argmax(model.predict(x),axis=1)
```

(ii). Vegetable dataset

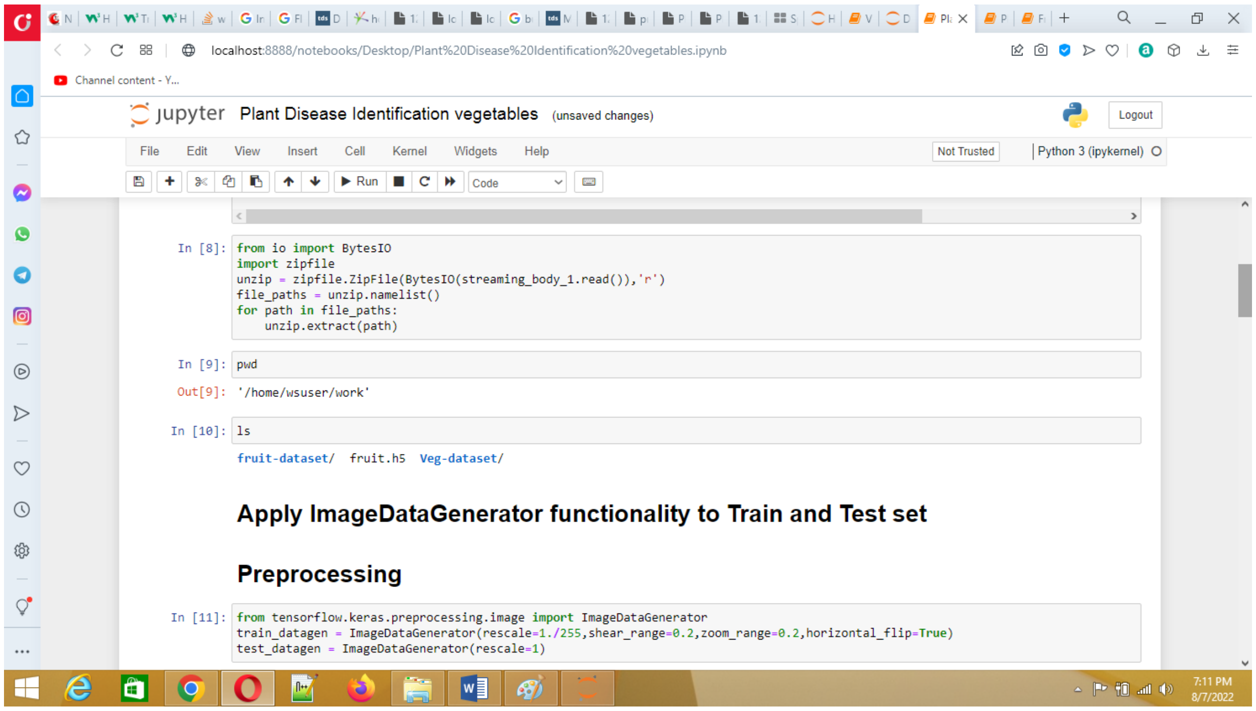


localhost:8888/notebooks/Desktop/Plant%20Disease%20Identification%20vegetables.ipynb

jupyter Plant Disease Identification vegetables (unsaved changes)

Kernel starting, please wait... | Not Trusted | Python 3 (ipykernel)

```
In [6]: pwd

Out[6]: '/home/wsuser/work'

In [7]:
        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.
        client_a52c887f20bf41468f32fc39f1d88cd8 = ibm_boto3.client(service_name='s3',
            ibm_api_key_id='t6e9lqUJeBBJJVCNi_ZIIR7I6T56JYT2qsy6A4yuErJd',
            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')

        streaming_body_1 = client_a52c887f20bf41468f32fc39f1d88cd8.get_object(Bucket='plantdiseaseidentification-donotdelete-pr-ls9zwh6qa

        # 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/
```

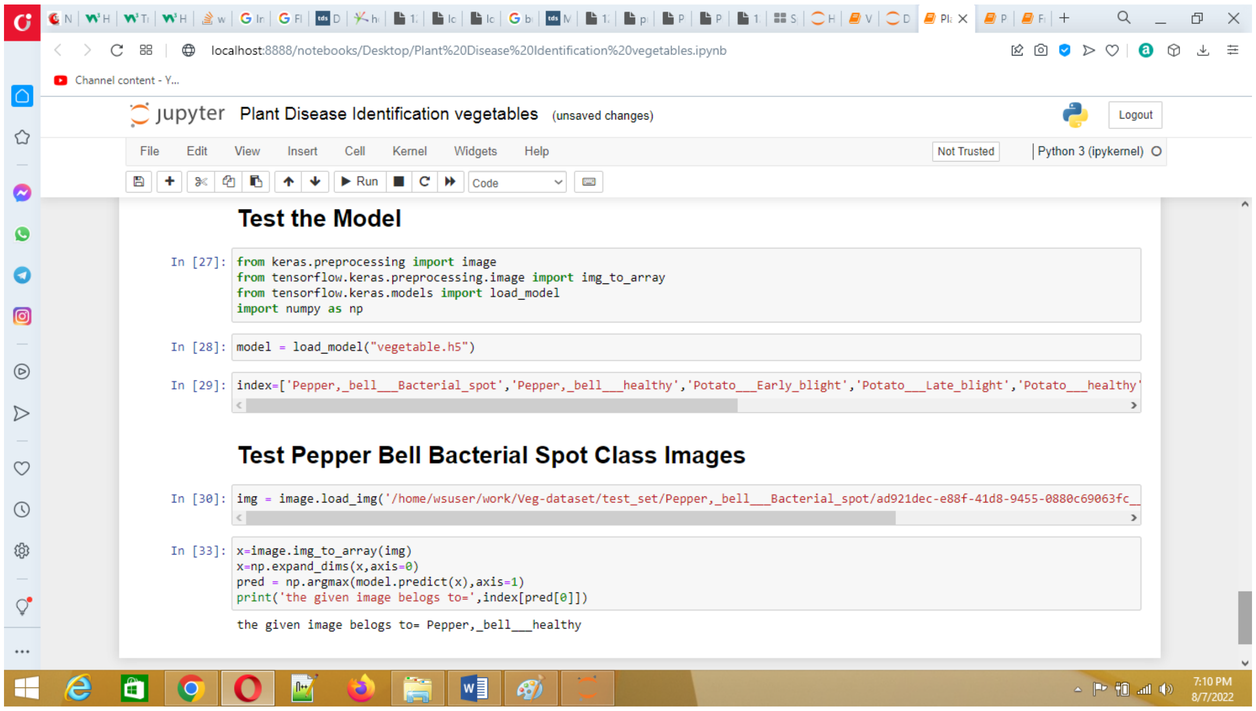


```
In [8]: from io import BytesIO
        import zipfile
        unzip = zipfile.ZipFile(BytesIO(streaming_body_1.read()),'r')
        file_paths = unzip.namelist()
        for path in file_paths:
            unzip.extract(path)

In [9]: pwd

Out[9]: '/home/wsuser/work'

In [10]: ls

         fruit-dataset/  fruit.h5  Veg-dataset/
```

Apply ImageDataGenerator functionality to Train and Test set

Preprocessing

```
In [11]: from tensorflow.keras.preprocessing.image import ImageDataGenerator
         train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
         test_datagen = ImageDataGenerator(rescale=1)
```



Test the Model

```
In [27]: from keras.preprocessing import image
         from tensorflow.keras.preprocessing.image import img_to_array
         from tensorflow.keras.models import load_model
         import numpy as np

In [28]: model = load_model("vegetable.h5")

In [29]: index=['Pepper,_bell___Bacterial_spot','Pepper,_bell___healthy','Potato___Early_blight','Potato___Late_blight','Potato___healthy'
```

Test Pepper Bell Bacterial Spot Class Images

```
In [30]: img = image.load_img('/home/wsuser/work/Veg-dataset/test_set/Pepper,_bell___Bacterial_spot/ad921dec-e88f-41d8-9455-0880c69063fc_

In [33]: x=image.img_to_array(img)
         x=np.expand_dims(x,axis=0)
         pred = np.argmax(model.predict(x),axis=1)
         print('the given image belogs to=',index[pred[0]])

         the given image belogs to= Pepper,_bell___healthy
```