PROJECT DEVELOPMENT PHASE

SPRINT 4

DATE	18 November 2022
TEAM ID	PNT2022TMID35502
PROJECT NAME	Detecting Parkinson's Disease using Machine Learning

Sprint 4 Task

- Deployment of ML Model using IBM cloud.

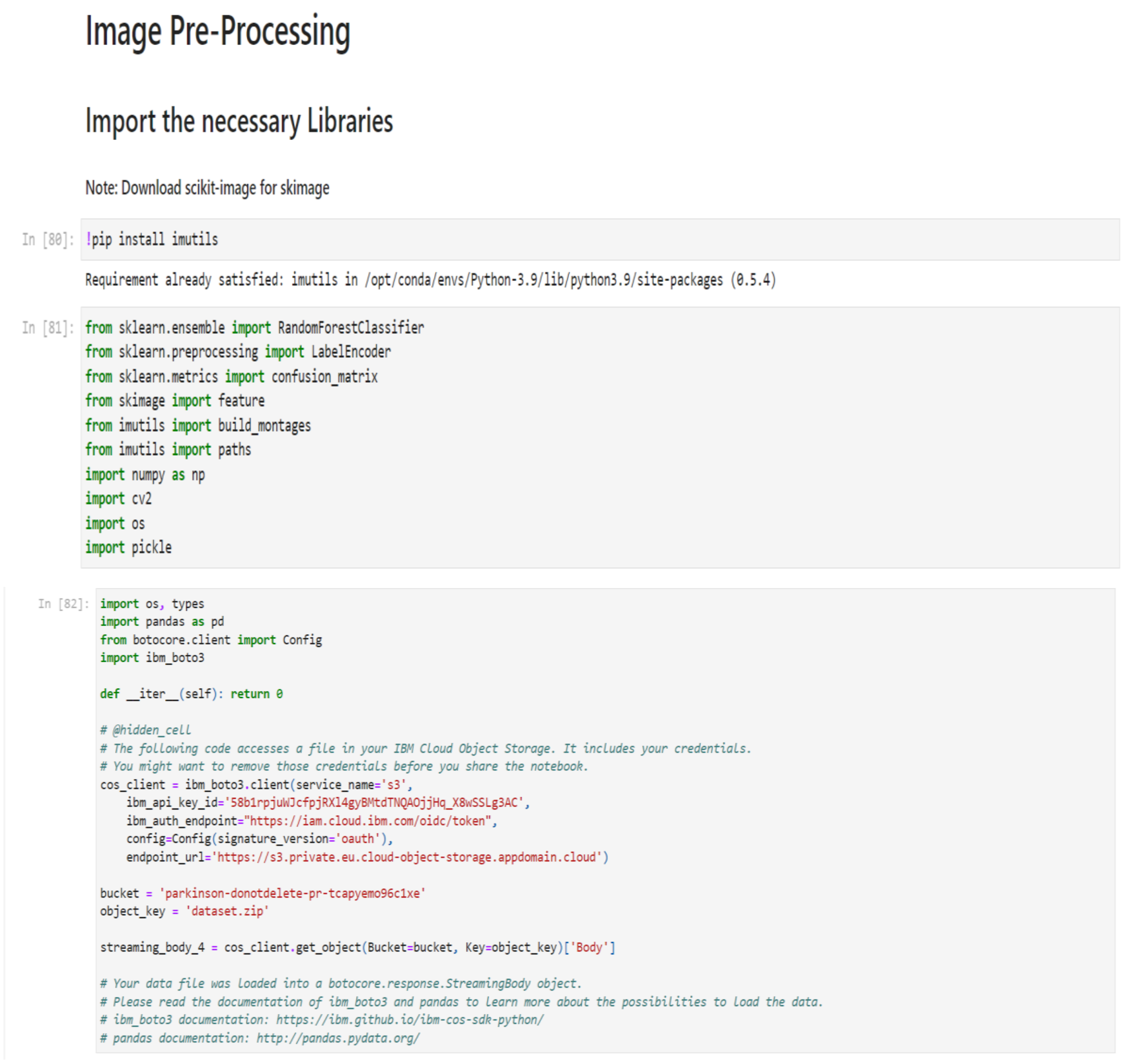


Image Pre-Processing

Import the necessary Libraries

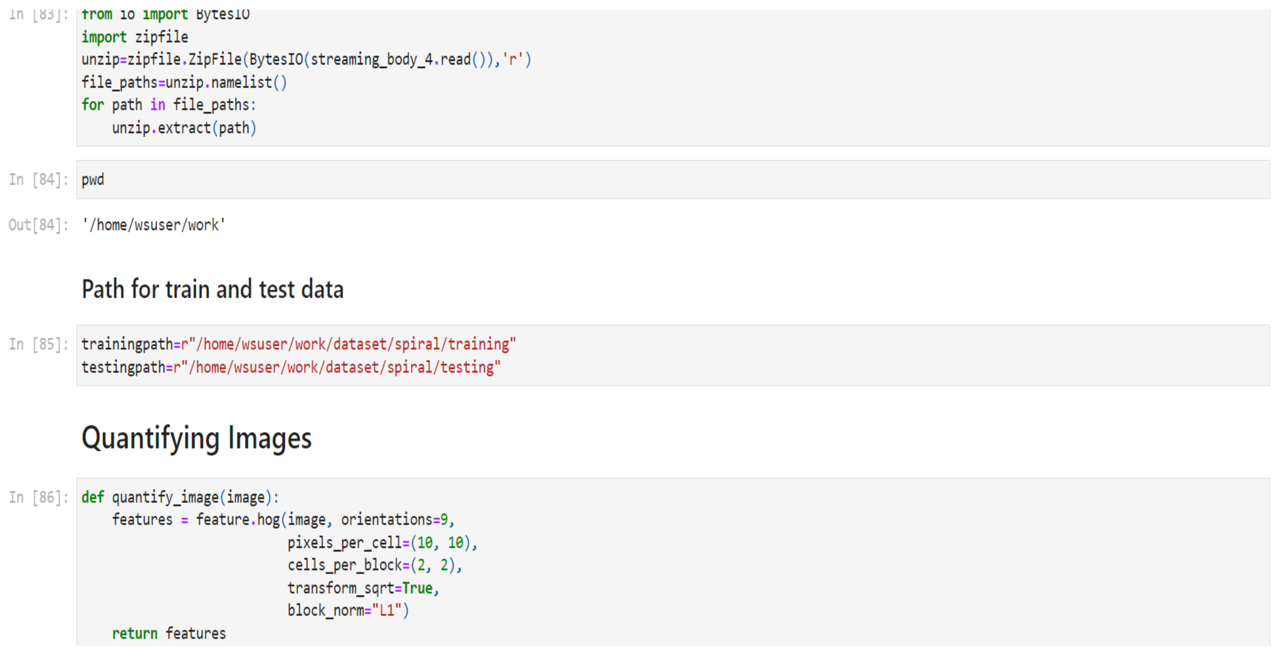
Note: Download scikit-image for skimage

```
In [80]: !pip install imutils
```

```
Requirement already satisfied: imutils in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (0.5.4)
```

```
In [81]: from sklearn.ensemble import RandomForestClassifier
         from sklearn.preprocessing import LabelEncoder
         from sklearn.metrics import confusion_matrix
         from skimage import feature
         from imutils import build_montages
         from imutils import paths
         import numpy as np
         import cv2
         import os
         import pickle
```

```
In [82]: 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='58b1rpjuWJcfpjRXl4gyBMtdTNQAOjjHq_X8wSSLg3AC',
             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 = 'parkinson-donotdelete-pr-tcapyemo96c1xe'
         object_key = 'dataset.zip'

         streaming_body_4 = 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 [83]: from io import BytesIO
         import zipfile
         unzip=zipfile.ZipFile(BytesIO(streaming_body_4.read()),'r')
         file_paths=unzip.namelist()
         for path in file_paths:
             unzip.extract(path)
```

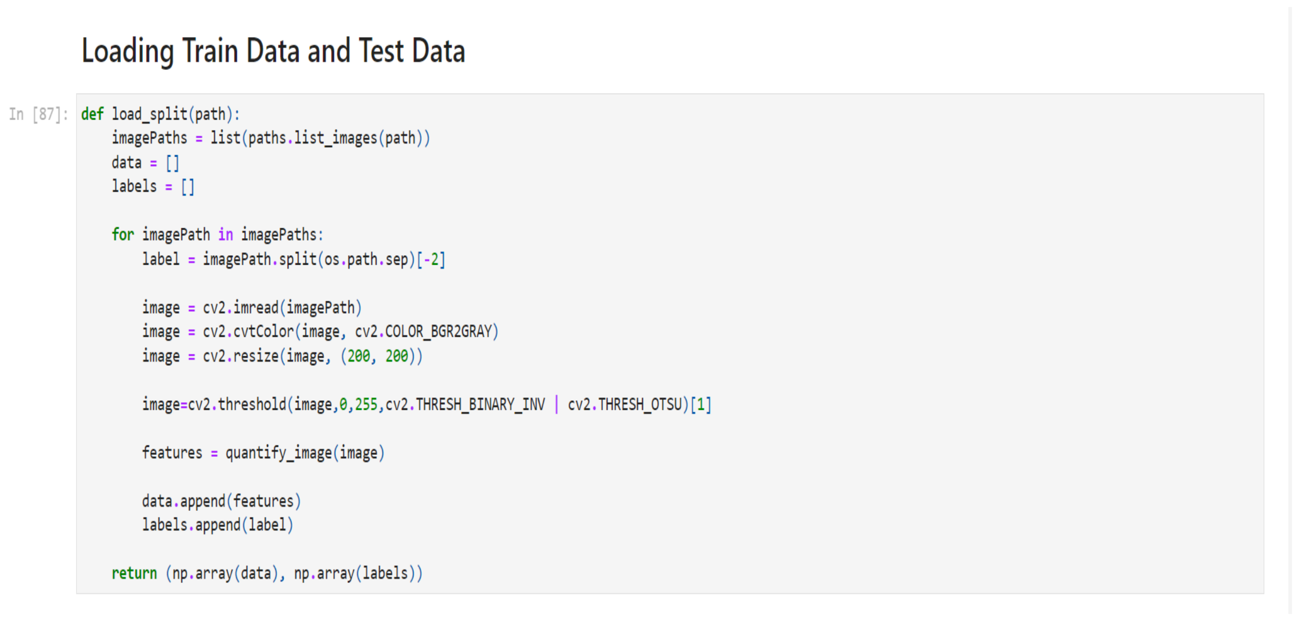
```
In [84]: pwd

Out[84]: '/home/wsuser/work'
```

Path for train and test data

```
In [85]: trainingpath=r"/home/wsuser/work/dataset/spiral/training"
         testingpath=r"/home/wsuser/work/dataset/spiral/testing"
```

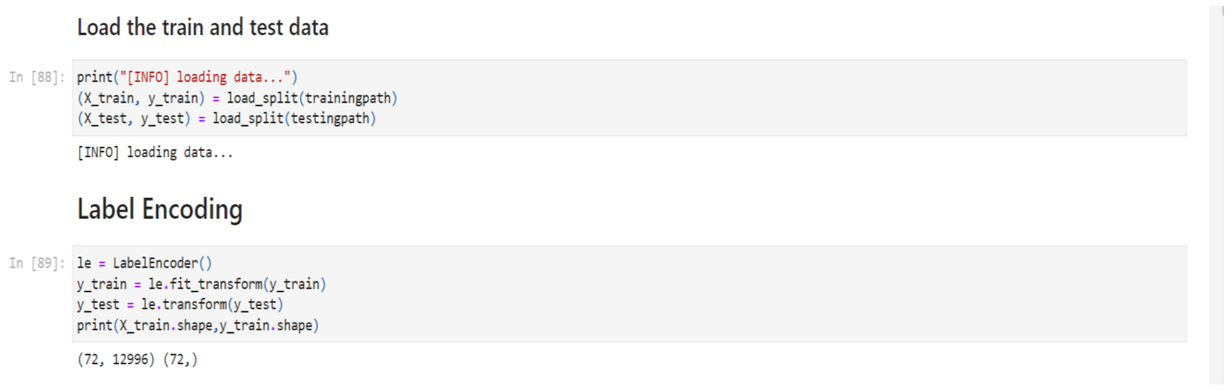
Quantifying Images

```
In [86]: def quantify_image(image):
             features = feature.hog(image, orientations=9,
                                     pixels_per_cell=(10, 10),
                                     cells_per_block=(2, 2),
                                     transform_sqrt=True,
                                     block_norm="L1")
             return features
```



Loading Train Data and Test Data

```
In [87]: def load_split(path):
             imagePaths = list(paths.list_images(path))
             data = []
             labels = []

             for imagePath in imagePaths:
                 label = imagePath.split(os.path.sep)[-2]

                 image = cv2.imread(imagePath)
                 image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
                 image = cv2.resize(image, (200, 200))

                 image=cv2.threshold(image,0,255,cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)[1]

                 features = quantify_image(image)

                 data.append(features)
                 labels.append(label)

             return (np.array(data), np.array(labels))
```



Load the train and test data

```
In [88]: print("[INFO] loading data...")
         (X_train, y_train) = load_split(trainingpath)
         (X_test, y_test) = load_split(testingpath)

         [INFO] loading data...
```

Label Encoding

```
In [89]: le = LabelEncoder()
         y_train = le.fit_transform(y_train)
         y_test = le.transform(y_test)
         print(X_train.shape,y_train.shape)

         (72, 12996) (72,)
```

Model Building

Training The Model

In [90]:
```
print("[INFO] training model")
model = RandomForestClassifier(n_estimators=100)
model.fit(X_train, y_train)
```

```
[INFO] training model
```

Out[90]:
```
RandomForestClassifier()
```

Testing The Model

In [91]:
```
testingpath=list(paths.list_images(testingpath))
idxs=np.arange(0,len(testingpath))
idxs=np.random.choice(idxs,size=(25,),replace=False)
images=[]
```

In [92]:
```
for i in idxs:
    image=cv2.imread(testingpath[i])
    output=image.copy()

    # load the input image,convert to grayscale and resize

    output=cv2.resize(output,(128,128))
    image=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
    image=cv2.resize(image,(200,200))
    image=cv2.threshold(image,0,255,cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)[1]

    #quantify the image and make predictions based on the  extracted feature using last trained random forest
    features=quantify_image(image)
    preds=model.predict([features])
    label=le.inverse_transform(preds)[0]
    #the set of output images
    if label=="healthy":
        color=(0,255,0)
    else:
        color=(0,0,255)

    cv2.putText(output,label,(3,20),cv2.FONT_HERSHEY_SIMPLEX,0.5,color,2)
    images.append(output)
```

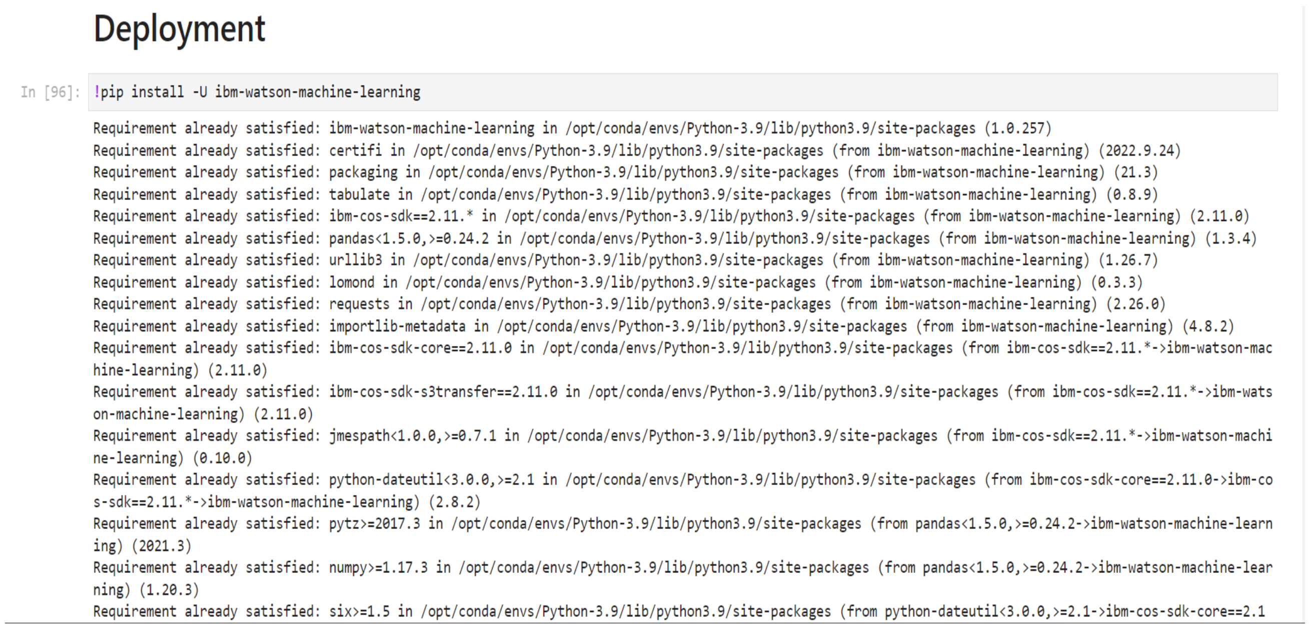
Model Evaluation

In [94]:
```
predictions = model.predict(X_test)

cm = confusion_matrix(y_test, predictions).flatten()
print(cm)
(tn, fp, fn, tp) = cm
accuracy = (tp + tn) / float(cm.sum())
print(accuracy)
```

```
[14  1  5 10]
0.8
```

Save The Model

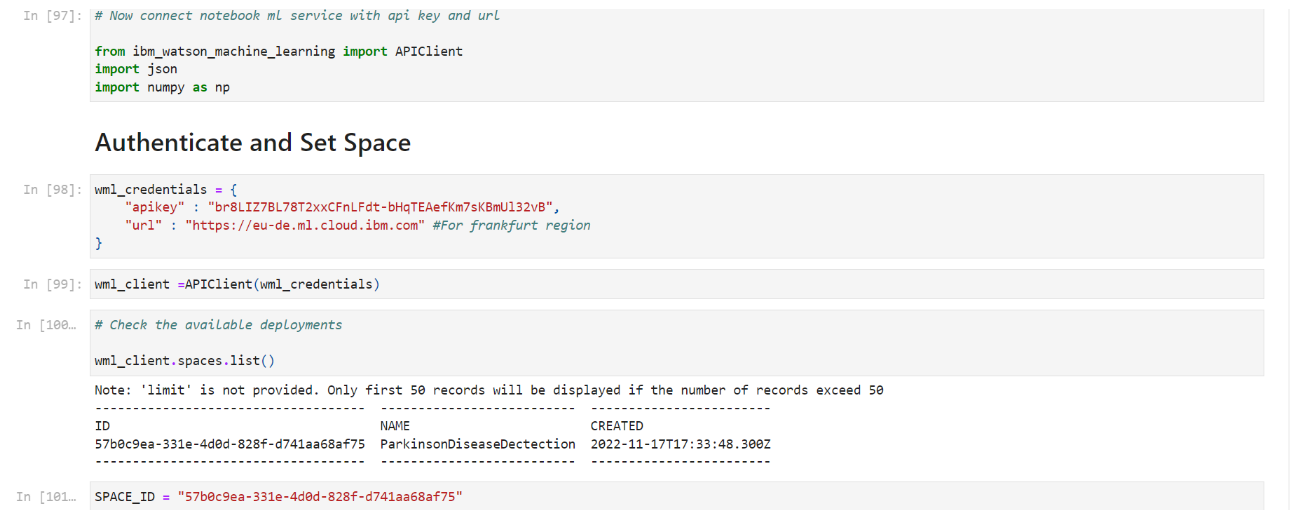
In [95]:
```
pickle.dump(model,open('parkinson.pkl','wb'))
```



Deployment

```
In [96]: !pip install -U ibm-watson-machine-learning
```

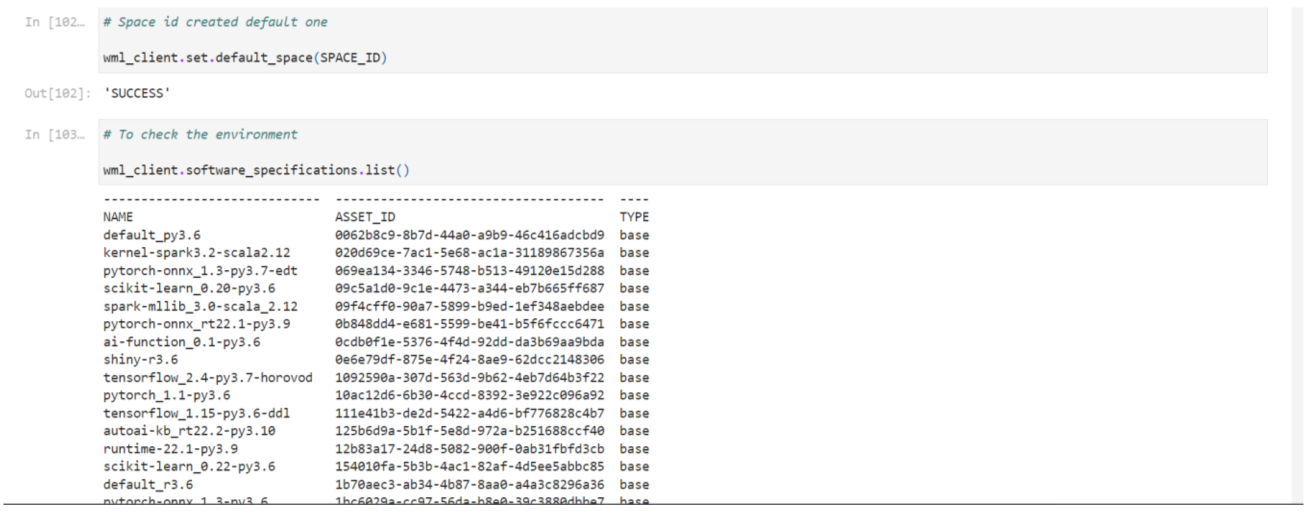
```
Requirement already satisfied: ibm-watson-machine-learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.257)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2022.9.24)
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (21.3)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (0.8.9)
Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/python3.9/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/python3.9/site-packages (from ibm-watson-machine-learning) (1.3.4)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (1.26.7)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (0.3.3)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2.26.0)
Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (4.8.2)
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-mac
hine-learning) (2.11.0)
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-wats
on-machine-learning) (2.11.0)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm-watson-machi
ne-learning) (0.10.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk-core==2.11.0->ibm-co
s-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/site-packages (from pandas<1.5.0,>=0.24.2->ibm-watson-machine-learn
ing) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm-watson-machine-lear
ning) (1.20.3)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.1
```



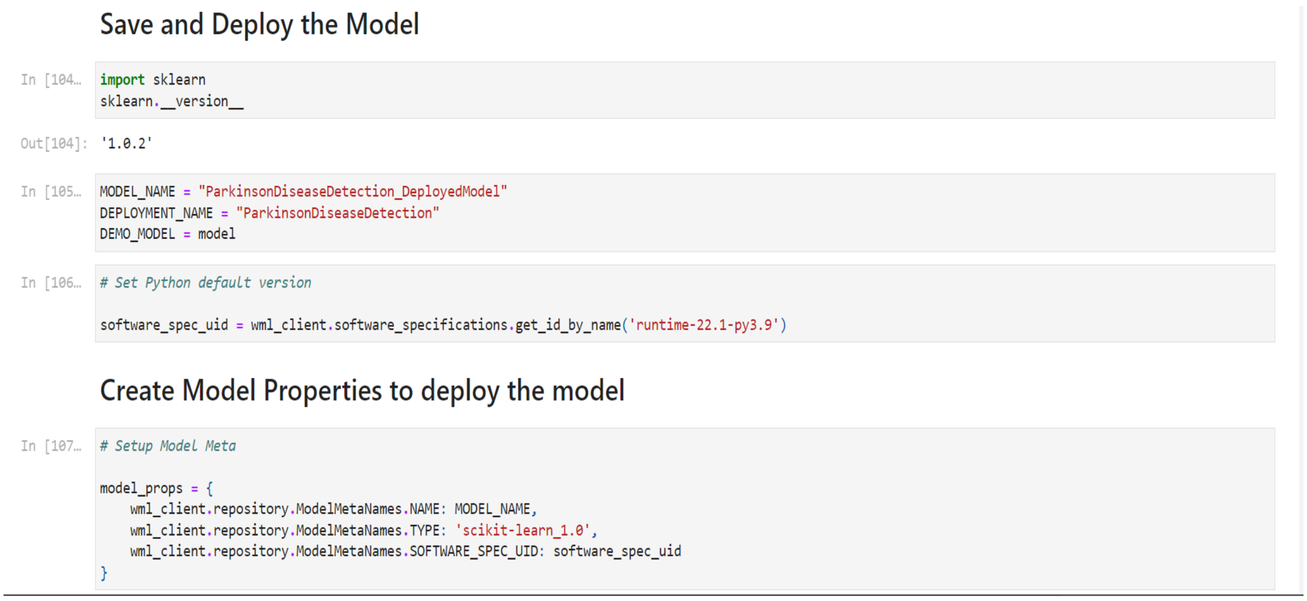
```
In [97]: # Now connect notebook ml service with api key and url

         from ibm_watson_machine_learning import APIClient
         import json
         import numpy as np
```

Authenticate and Set Space

```
In [98]: wml_credentials = {
             "apikey" : "br8LIZ7BL78T2xxCFnLFdt-bHqTEAefKm7sKBmUl32vB",
             "url" : "https://eu-de.ml.cloud.ibm.com" #For frankfurt region
         }

In [99]: wml_client =APIClient(wml_credentials)

In [100… # Check the available deployments

         wml_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
         57b0c9ea-331e-4d0d-828f-d741aa68af75  ParkinsonDiseaseDectection  2022-11-17T17:33:48.300Z
         ------------------------------------  --------------------------  ------------------------

In [101… SPACE_ID = "57b0c9ea-331e-4d0d-828f-d741aa68af75"
```



```
In [102… # Space id created default one

         wml_client.set_default_space(SPACE_ID)

Out[102]: 'SUCCESS'

In [103… # To check the environment

         wml_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
```



Save and Deploy the Model

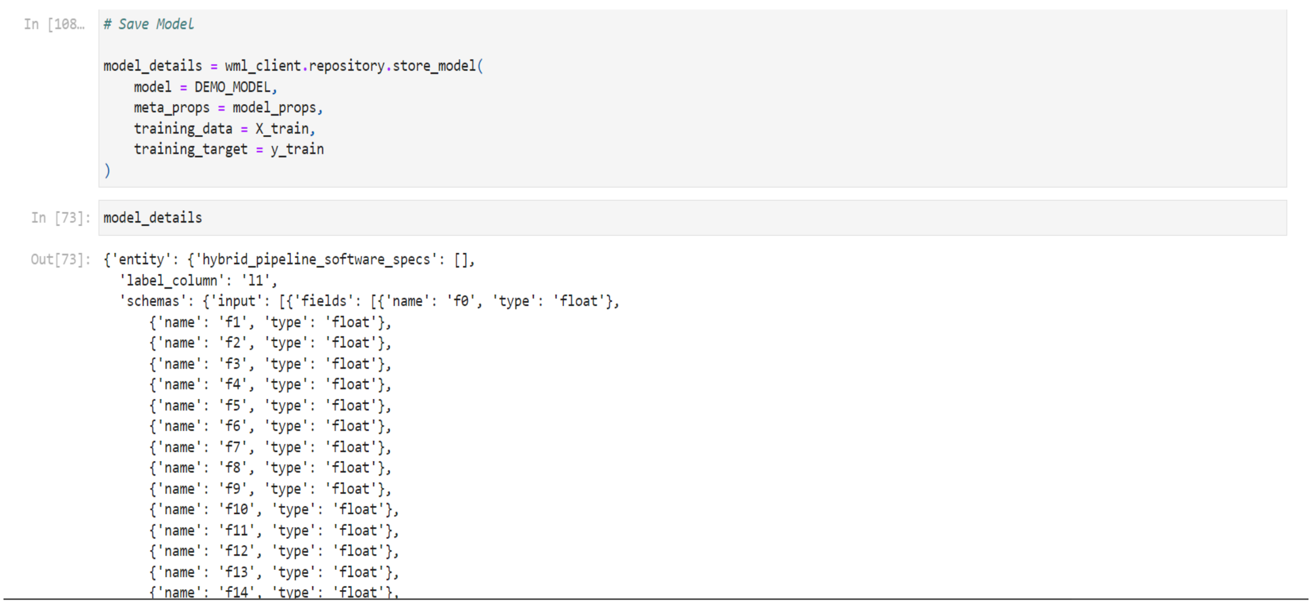
```
In [104…  import sklearn
          sklearn.__version__

Out[104]: '1.0.2'
```

```
In [105…  MODEL_NAME = "ParkinsonDiseaseDetection_DeployedModel"
          DEPLOYMENT_NAME = "ParkinsonDiseaseDetection"
          DEMO_MODEL = model
```

```
In [106…  # Set Python default version

          software_spec_uid = wml_client.software_specifications.get_id_by_name('runtime-22.1-py3.9')
```

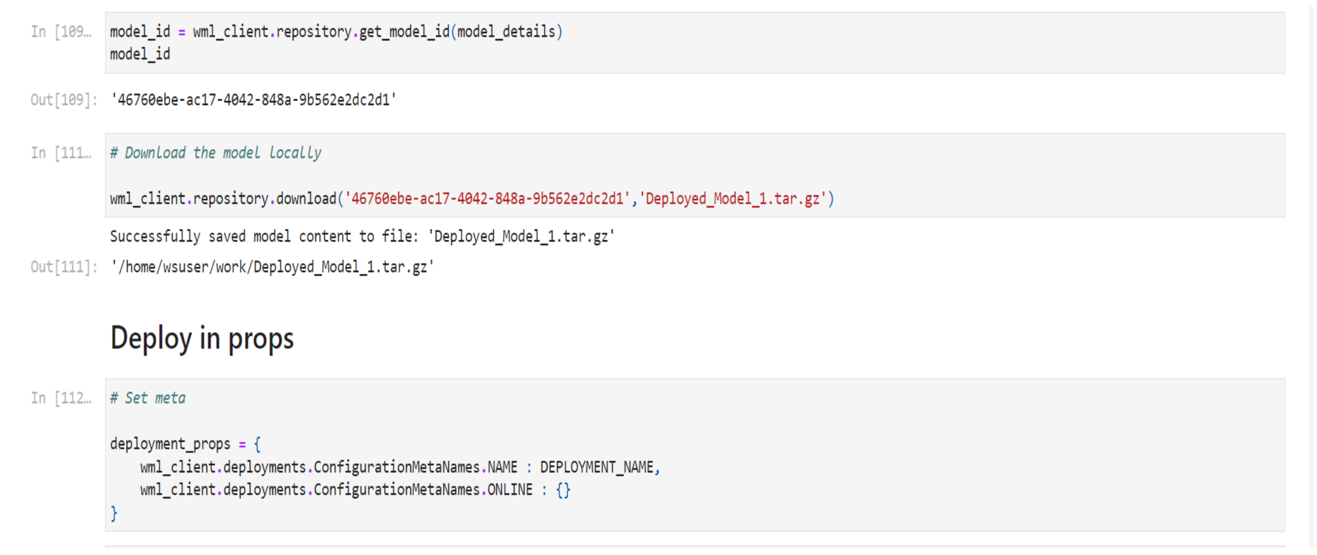
Create Model Properties to deploy the model

```
In [107…  # Setup Model Meta

          model_props = {
              wml_client.repository.ModelMetaNames.NAME: MODEL_NAME,
              wml_client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.0',
              wml_client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid
          }
```



```
In [108…  # Save Model

          model_details = wml_client.repository.store_model(
              model = DEMO_MODEL,
              meta_props = model_props,
              training_data = X_train,
              training_target = y_train
          )
```

```
In [73]:  model_details

Out[73]:  {'entity': {'hybrid_pipeline_software_specs': [],
            'label_column': 'l1',
            'schemas': {'input': [{'fields': [{'name': 'f0', 'type': 'float'},
                {'name': 'f1', 'type': 'float'},
                {'name': 'f2', 'type': 'float'},
                {'name': 'f3', 'type': 'float'},
                {'name': 'f4', 'type': 'float'},
                {'name': 'f5', 'type': 'float'},
                {'name': 'f6', 'type': 'float'},
                {'name': 'f7', 'type': 'float'},
                {'name': 'f8', 'type': 'float'},
                {'name': 'f9', 'type': 'float'},
                {'name': 'f10', 'type': 'float'},
                {'name': 'f11', 'type': 'float'},
                {'name': 'f12', 'type': 'float'},
                {'name': 'f13', 'type': 'float'},
                {'name': 'f14', 'type': 'float'},
```



```
In [109…  model_id = wml_client.repository.get_model_id(model_details)
          model_id

Out[109]: '46760ebe-ac17-4042-848a-9b562e2dc2d1'
```

```
In [111…  # Download the model locally

          wml_client.repository.download('46760ebe-ac17-4042-848a-9b562e2dc2d1','Deployed_Model_1.tar.gz')

          Successfully saved model content to file: 'Deployed_Model_1.tar.gz'

Out[111]: '/home/wsuser/work/Deployed_Model_1.tar.gz'
```

Deploy in props

```
In [112…  # Set meta

          deployment_props = {
              wml_client.deployments.ConfigurationMetaNames.NAME : DEPLOYMENT_NAME,
              wml_client.deployments.ConfigurationMetaNames.ONLINE : {}
          }
```

```
In [113…  # Deploy

          deployment = wml_client.deployments.create(
              artifact_uid = model_id,
              meta_props = deployment_props
          )
```

```
#########################################################################################

Synchronous deployment creation for uid: '46760ebe-ac17-4042-848a-9b562e2dc2d1' started

#########################################################################################


initializing
Note: online_url is deprecated and will be removed in a future release. Use serving_urls instead.

ready


-------------------------------------------------------------------------------------------------
Successfully finished deployment creation, deployment_uid='7dac085a-2669-4969-a9e1-4628b36838a6'
-------------------------------------------------------------------------------------------------
```