```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 1,
   "id": "9cc6d54f",
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Collecting watson-machine-learning-client\n",
      "  Downloading watson_machine_learning_client-1.0.391-py3-none-any.whl (538 kB)\n",
     "Requirement already satisfied: boto3 in c:\\users\\karup\\anaconda3\\lib\\site-packages (from watson-machine-learning-client) (1.21.32)\n",
     "Requirement already satisfied: tqdm in c:\\users\\karup\\anaconda3\\lib\\site-packages (from watson-machine-learning-client) (4.64.0)\n",
     "Requirement already satisfied: pandas in c:\\users\\karup\\anaconda3\\lib\\site-packages (from watson-machine-learning-client) (1.4.2)\n",
     "Requirement already satisfied: tabulate in c:\\users\\karup\\anaconda3\\lib\\site-packages (from watson-machine-learning-client) (0.8.9)\n",
     "Requirement already satisfied: ibm-cos-sdk in c:\\users\\karup\\anaconda3\\lib\\site-packages (from watson-machine-learning-client) (2.11.0)\n",
     "Requirement already satisfied: requests in c:\\users\\karup\\anaconda3\\lib\\site-packages (from watson-machine-learning-client) (2.27.1)\n",
     "Requirement already satisfied: certifi in c:\\users\\karup\\anaconda3\\lib\\site-packages (from watson-machine-learning-client) (2021.10.8)\n",
     "Requirement already satisfied: urllib3 in c:\\users\\karup\\anaconda3\\lib\\site-packages (from watson-machine-learning-client) (1.26.9)\n",
     "Requirement already satisfied: lomond in c:\\users\\karup\\anaconda3\\lib\\site-packages (from watson-machine-learning-client) (0.3.3)\n",
```

"Requirement already satisfied: botocore<1.25.0,>=1.24.32 in c:\\users\\karup\\anaconda3\\lib\\site-packages (from boto3->watson-machine-learning-client) (1.24.32)\n",

"Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in c:\\users\\karup\\anaconda3\\lib\\site-packages (from boto3->watson-machine-learning-client) (0.5.0)\n",

"Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in c:\\users\\karup\\anaconda3\\lib\\site-packages (from boto3->watson-machine-learning-client) (0.10.0)\n",

"Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in c:\\users\\karup\\anaconda3\\lib\\site-packages (from botocore<1.25.0,>=1.24.32->boto3->watson-machine-learning-client) (2.8.2)\n",

"Requirement already satisfied: six>=1.5 in c:\\users\\karup\\anaconda3\\lib\\site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.25.0,>=1.24.32->boto3->watson-machine-learning-client) (1.16.0)\n",

"Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in c:\\users\\karup\\anaconda3\\lib\\site-packages (from ibm-cos-sdk->watson-machine-learning-client) (2.11.0)\n",

"Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in c:\\users\\karup\\anaconda3\\lib\\site-packages (from ibm-cos-sdk->watson-machine-learning-client) (2.11.0)\n",

"Requirement already satisfied: charset-normalizer~=2.0.0 in c:\\users\\karup\\anaconda3\\lib\\site-packages (from requests->watson-machine-learning-client) (2.0.4)\n",

"Requirement already satisfied: idna<4,>=2.5 in c:\\users\\karup\\anaconda3\\lib\\site-packages (from requests->watson-machine-learning-client) (3.3)\n",

"Requirement already satisfied: numpy>=1.18.5 in c:\\users\\karup\\anaconda3\\lib\\site-packages (from pandas->watson-machine-learning-client) (1.21.5)\n",

"Requirement already satisfied: pytz>=2020.1 in c:\\users\\karup\\anaconda3\\lib\\site-packages (from pandas->watson-machine-learning-client) (2021.3)\n",

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

"Installing collected packages: watson-machine-learning-client\n",

"Successfully installed watson-machine-learning-client-1.0.391\n"

]

```
  }
 ],
 "source": [
  "!pip install  watson-machine-learning-client"
 ]
},
{
 "cell_type": "code",
 "execution_count": 2,
 "id": "21444c5a",
 "metadata": {},
 "outputs": [],
 "source": [
  "from ibm_watson_machine_learning import APIClient\n",
  "wml_credentials={\n",
  "    \"url\":\"https://us-south.ml.cloud.ibm.com\",\n",
  "    \"apikey\":\"BPFGcOrCf3sroRy3uKOPGozsmIL-5oVDv4A_lru2IpMS\"\n",
  "    \n",
  "}\n",
  "\n",
  "client=APIClient(wml_credentials)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 3,
 "id": "c33fac4b",
 "metadata": {},
 "outputs": [],
```

```
 "source": [
  "client=APIClient(wml_credentials)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 4,
 "id": "0092fe7d",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "<ibm_watson_machine_learning.client.APIClient at 0x1894751f2e0>"
    ]
   },
   "execution_count": 4,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "client"
 ]
},
{
 "cell_type": "code",
 "execution_count": 5,
 "id": "d32ad377",
```

```
  "metadata": {},
  "outputs": [],
  "source": [
   " def guid_from_space_name(client, space_name):\n",
   "     space=client.spaces.get_details()\n",
   "     #print(space)\n",
   "     return(next(item for item in space['resources'] if item['entity']['name']== space_name)['metadata']['id'])"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 6,
  "id": "7de3338b",
  "metadata": {},
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "Space UID =ba02adea-7e10-4237-81e7-eaf084fe4102\n"
    ]
   }
  ],
  "source": [
   "space_uid=guid_from_space_name(client,'imageclassification') #imageclassification is the deployment space name\n",
   "print(\"Space UID =\"+space_uid)"
  ]
 },
```

```
{
 "cell_type": "code",
 "execution_count": 7,
 "id": "2a4bc55d",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "'SUCCESS'"
    ]
   },
   "execution_count": 7,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "client.set.default_space(space_uid)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 9,
 "id": "0d2120a9",
 "metadata": {},
 "outputs": [
  {
   "name": "stdout",
```

```
   "output_type": "stream",
   "text": [
    "Successfully saved model content to file: 'nutrition.tar.gz'\n"
   ]
  },
  {
   "data": {
    "text/plain": [
     "'C:\\\\Users\\\\karup\\\\Downloads/nutrition.tar.gz'"
    ]
   },
   "execution_count": 9,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "client.repository.download(\"f3e12114-24f4-4bae-9d60-2897d27e7ce6\", 'nutrition.tar.gz')"
 ]
},
{
 "cell_type": "code",
 "execution_count": 10,
 "id": "58ec3b2d",
 "metadata": {},
 "outputs": [],
 "source": [
  "from keras.models import load_model\n",
  "from keras.preprocessing import image"
```

```
 ]
},
{
 "cell_type": "code",
 "execution_count": 11,
 "id": "921c9ffa",
 "metadata": {},
 "outputs": [],
 "source": [
  "model = load_model(\"nutrition.h5\")"
 ]
},
{
 "cell_type": "code",
 "execution_count": 13,
 "id": "d0970dbd",
 "metadata": {},
 "outputs": [],
 "source": [
  "import numpy as np"
 ]
},
{
 "cell_type": "code",
 "execution_count": 14,
 "id": "59461786",
 "metadata": {},
 "outputs": [
  {
```

```
 "name": "stdout",
 "output_type": "stream",
 "text": [
  "1/1 [==============================] - 1s 696ms/step\n"
 ]
},
{
 "data": {
  "text/plain": [
   "array([0], dtype=int64)"
  ]
 },
 "execution_count": 14,
 "metadata": {},
 "output_type": "execute_result"
}
],
"source": [
 "from tensorflow.keras.utils import load_img\n",
 "from tensorflow.keras.utils import img_to_array\n",
 "#loading of the image\n",
 "img = load_img(\"apple.jpg\", grayscale=False,target_size=(64,64))\n",
 "#image to array \n",
 "x = img_to_array(img)\n",
 "#changing the shape\n",
 "x= np.expand_dims(x,axis = 0)\n",
 "predict_x=model.predict(x)\n",
 "classes_x=np.argmax(predict_x,axis = -1)\n",
 "classes_x"
```

```
 ]
},
{
 "cell_type": "code",
 "execution_count": 15,
 "id": "3207a0d1",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "'APPLES'"
    ]
   },
   "execution_count": 15,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "index=['APPLES', 'BANANA', 'ORANGE', 'PINEAPPLE', 'WATERMELON']\n",
  "result=str(index[classes_x[0]])\n",
  "result"
 ]
},
{
 "cell_type": "code",
 "execution_count": 16,
 "id": "f1700a44",
```

```
"metadata": {},
"outputs": [
 {
  "name": "stdout",
  "output_type": "stream",
  "text": [
   "1/1 [==============================] - 0s 40ms/step\n"
  ]
 },
 {
  "data": {
   "text/plain": [
    "array([1], dtype=int64)"
   ]
  },
  "execution_count": 16,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "from tensorflow.keras.utils import load_img\n",
 "from tensorflow.keras.utils import img_to_array\n",
 "#loading of the image\n",
 "img = load_img(\"banana.jpg\", grayscale=False,target_size=(64,64))\n",
 "#image to array \n",
 "x = img_to_array(img)\n",
 "#changing the shape\n",
 "x= np.expand_dims(x,axis = 0)\n",
```

```
  "predict_x=model.predict(x)\n",
  "classes_x=np.argmax(predict_x,axis = -1)\n",
  "classes_x"
 ]
},
{
 "cell_type": "code",
 "execution_count": 17,
 "id": "892ded33",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "'BANANA'"
    ]
   },
   "execution_count": 17,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "index=['APPLES', 'BANANA', 'ORANGE', 'PINEAPPLE', 'WATERMELON']\n",
  "result=str(index[classes_x[0]])\n",
  "result"
 ]
},
{
```

```
"cell_type": "code",
"execution_count": 20,
"id": "09bcd8ca",
"metadata": {},
"outputs": [
 {
  "name": "stdout",
  "output_type": "stream",
  "text": [
   "1/1 [==============================] - 0s 40ms/step\n"
  ]
 },
 {
  "data": {
   "text/plain": [
    "array([3], dtype=int64)"
   ]
  },
  "execution_count": 20,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "from tensorflow.keras.utils import load_img\n",
 "from tensorflow.keras.utils import img_to_array\n",
 "#loading of the image\n",
 "img = load_img(\"Test_Image5.jpg\", grayscale=False,target_size=(64,64))\n",
 "#image to array \n",
```

```
  "x = img_to_array(img)\n",
  "#changing the shape\n",
  "x= np.expand_dims(x,axis = 0)\n",
  "predict_x=model.predict(x)\n",
  "classes_x=np.argmax(predict_x,axis = -1)\n",
  "classes_x"
 ]
},
{
 "cell_type": "code",
 "execution_count": 21,
 "id": "744424f9",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "'PINEAPPLE'"
    ]
   },
   "execution_count": 21,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "index=['APPLES', 'BANANA', 'ORANGE', 'PINEAPPLE', 'WATERMELON']\n",
  "result=str(index[classes_x[0]])\n",
  "result"
```

```
  ]
 }
],
"metadata": {
 "kernelspec": {
  "display_name": "Python 3 (ipykernel)",
  "language": "python",
  "name": "python3"
 },
 "language_info": {
  "codemirror_mode": {
   "name": "ipython",
   "version": 3
  },
  "file_extension": ".py",
  "mimetype": "text/x-python",
  "name": "python",
  "nbconvert_exporter": "python",
  "pygments_lexer": "ipython3",
  "version": "3.9.12"
 }
},
"nbformat": 4,
"nbformat_minor": 5
}
```