```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "7DRYqjaF1Hzi"
   },
   "source": [
    "IBM cloud deployment\n",
    "Team ID - PNT2022TMID39460"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "2uJVgXBn1PG7"
   },
   "source": [
    "Importing the required libraries"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "Cf7c189_1TBT",
    "outputId": "0d9cb04c-f3b2-4017-bbed-09afe587a508"
   },
```

```
"outputs": [
 {
  "name": "stdout",
  "output_type": "stream",
  "text": [
   "Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/\n",
   "Requirement already satisfied: tensorflow in /usr/local/lib/python3.7/dist-packages (2.9.2)\n",
   "Collecting tensorflow\n",
   "  Downloading tensorflow-2.10.0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (578.0 MB)\n",
   "\u001b[K     |████████████████████████████████| 578.0 MB 17 kB/s \n",
   "\u001b[?25hRequirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.3.0)\n",
   "Requirement already satisfied: gast<=0.4.0,>=0.2.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (0.4.0)\n",
   "Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.6.3)\n",
   "Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (3.3.0)\n",
   "Collecting flatbuffers>=2.0\n",
   "  Downloading flatbuffers-22.10.26-py2.py3-none-any.whl (26 kB)\n",
   "Collecting keras<2.11,>=2.10.0\n",
   "  Downloading keras-2.10.0-py2.py3-none-any.whl (1.7 MB)\n",
   "\u001b[K     |████████████████████████████████| 1.7 MB 52.9 MB/s \n",
   "\u001b[?25hRequirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.50.0)\n",
   "Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (0.2.0)\n",
   "Requirement already satisfied: protobuf<3.20,>=3.9.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (3.17.3)\n",
   "Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.15.0)\n",
   "Collecting tensorboard<2.11,>=2.10\n",
```

" Downloading tensorboard-2.10.1-py3-none-any.whl (5.9 MB)\n",

"\u001b[K |  | 5.9 MB 37.6 MB/s \n",

"\u001b[?25hCollecting tensorflow-estimator<2.11,>=2.10.0\n",

" Downloading tensorflow_estimator-2.10.0-py2.py3-none-any.whl (438 kB)\n",

"\u001b[K |  | 438 kB 64.6 MB/s \n",

"\u001b[?25hRequirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (4.1.1)\n",

"Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from tensorflow) (21.3)\n",

"Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.14.1)\n",

"Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (0.27.0)\n",

"Requirement already satisfied: keras-preprocessing>=1.1.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.1.2)\n",

"Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.21.6)\n",

"Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from tensorflow) (57.4.0)\n",

"Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (2.0.1)\n",

"Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (3.1.0)\n",

"Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (14.0.6)\n",

"Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.7/dist-packages (from astunparse>=1.6.0->tensorflow) (0.37.1)\n",

"Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dist-packages (from h5py>=2.9.0->tensorflow) (1.5.2)\n",

"Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.11,>=2.10->tensorflow) (1.0.1)\n",

"Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.11,>=2.10->tensorflow) (3.4.1)\n",

"Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.11,>=2.10->tensorflow) (2.23.0)\n",

"Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.11,>=2.10->tensorflow) (1.8.1)\n",

"Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.11,>=2.10->tensorflow) (0.4.6)\n",

"Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.11,>=2.10->tensorflow) (0.6.1)\n",

"Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.7/dist-packages (from tensorboard<2.11,>=2.10->tensorflow) (1.35.0)\n",

"Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.7/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.11,>=2.10->tensorflow) (0.2.8)\n",

"Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.11,>=2.10->tensorflow) (4.2.4)\n",

"Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.11,>=2.10->tensorflow) (4.9)\n",

"Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/dist-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.11,>=2.10->tensorflow) (1.3.1)\n",

"Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.7/dist-packages (from markdown>=2.6.8->tensorboard<2.11,>=2.10->tensorflow) (4.13.0)\n",

"Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata>=4.4->markdown>=2.6.8->tensorboard<2.11,>=2.10->tensorflow) (3.10.0)\n",

"Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.7/dist-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.11,>=2.10->tensorflow) (0.4.8)\n",

"Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard<2.11,>=2.10->tensorflow) (2022.9.24)\n",

"Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard<2.11,>=2.10->tensorflow) (1.24.3)\n",

"Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard<2.11,>=2.10->tensorflow) (2.10)\n",

"Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard<2.11,>=2.10->tensorflow) (3.0.4)\n",

"Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.11,>=2.10->tensorflow) (3.2.2)\n",

"Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging->tensorflow) (3.0.9)\n",

"Installing collected packages: tensorflow-estimator, tensorboard, keras, flatbuffers, tensorflow\n",

```
"  Attempting uninstall: tensorflow-estimator\n",
"    Found existing installation: tensorflow-estimator 2.9.0\n",
"    Uninstalling tensorflow-estimator-2.9.0:\n",
"      Successfully uninstalled tensorflow-estimator-2.9.0\n",
"  Attempting uninstall: tensorboard\n",
"    Found existing installation: tensorboard 2.9.1\n",
"    Uninstalling tensorboard-2.9.1:\n",
"      Successfully uninstalled tensorboard-2.9.1\n",
"  Attempting uninstall: keras\n",
"    Found existing installation: keras 2.9.0\n",
"    Uninstalling keras-2.9.0:\n",
"      Successfully uninstalled keras-2.9.0\n",
"  Attempting uninstall: flatbuffers\n",
"    Found existing installation: flatbuffers 1.12\n",
"    Uninstalling flatbuffers-1.12:\n",
"      Successfully uninstalled flatbuffers-1.12\n",
"  Attempting uninstall: tensorflow\n",
"    Found existing installation: tensorflow 2.9.2\n",
"    Uninstalling tensorflow-2.9.2:\n",
"      Successfully uninstalled tensorflow-2.9.2\n",
"Successfully installed flatbuffers-22.10.26 keras-2.10.0 tensorboard-2.10.1 tensorflow-2.10.0 tensorflow-estimator-2.10.0\n"
    ]
   }
  ],
  "source": [
   "!pip install tensorflow --upgrade"
  ]
 },
 {
  "cell_type": "code",
```

```
   "execution_count": null,
   "metadata": {
    "id": "ckIwSArd2Xya"
   },
   "outputs": [],
   "source": [
    "import numpy as np\n",
    "import tensorflow #open source used for both ML and DL for computation\n",
    "from tensorflow.keras.datasets import mnist #mnist dataset\n",
    "from tensorflow.keras.models import Sequential #it is a plain stack of layers\n",
    "from tensorflow.keras import layers #A Layer consists of a tensor- in tensor-out computat ion funct ion\n",
    "from tensorflow.keras.layers import Dense, Flatten #Dense-Dense Layer is the regular deeply connected r\n",
    "#faltten -used fot flattening the input or change the dimension\n",
    "from tensorflow.keras.layers import Conv2D #convolutional Layer\n",
    "from keras.utils import np_utils #used for one-hot encoding\n",
    "import matplotlib.pyplot as plt   #used for data visualization"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "faiaKpa-2hBG"
   },
   "source": [
    "Load data"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
```

```
    "metadata": {
      "id": "VTFMw5Tk2wH1"
    },
    "outputs": [],
    "source": [
      "(x_train, y_train), (x_test, y_test)=mnist.load_data () #splitting the mnist data into train and test\n",
      "Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz\n",
      "11493376/11490434 [==============================] - 0s 0us/step\n",
      "11501568/11490434 [==============================] - 0s 0us/step"
    ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "id": "P7GThTjA3Kh1",
      "outputId": "4fee61a1-221b-4234-f68b-285afd144f2d"
    },
    "outputs": [
     {
       "name": "stdout",
       "output_type": "stream",
       "text": [
         "(60000, 28, 28)\n",
         "(10000, 28, 28)\n"
       ]
     }
```

```
    ],
    "source": [
     "print (x_train.shape)  #shape is used for give the dimens ion values #60000-rows 28x28-pixels\n",
     "print (x_test.shape)"
    ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/"
     },
     "id": "QjwZixUf3Pus",
     "outputId": "4a152654-a8f3-4161-83a5-b5e89517f999"
    },
    "outputs": [
     {
      "data": {
       "text/plain": [
        "array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
        "          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
        "          0,   0],\n",
        "       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
        "          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
        "          0,   0],\n",
        "       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
        "          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
        "          0,   0],\n",
        "       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
```

```
"       0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
"       0,  0],\n",
"     [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
"       0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
"       0,  0],\n",
"     [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  3,\n",
"      18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,  0,  0,\n",
"       0,  0],\n",
"     [ 0,  0,  0,  0,  0,  0,  0,  0, 30,  36,  94, 154, 170,\n",
"      253, 253, 253, 253, 253, 225, 172, 253, 242, 195,  64,  0,  0,\n",
"       0,  0],\n",
"     [ 0,  0,  0,  0,  0,  0,  0, 49, 238, 253, 253, 253, 253,\n",
"      253, 253, 253, 253, 251,  93,  82,  82,  56,  39,  0,  0,  0,\n",
"       0,  0],\n",
"     [ 0,  0,  0,  0,  0,  0,  0, 18, 219, 253, 253, 253, 253,\n",
"      253, 198, 182, 247, 241,  0,  0,  0,  0,  0,  0,  0,  0,\n",
"       0,  0],\n",
"     [ 0,  0,  0,  0,  0,  0,  0,  0, 80, 156, 107, 253, 253,\n",
"      205,  11,  0,  43, 154,  0,  0,  0,  0,  0,  0,  0,  0,\n",
"       0,  0],\n",
"     [ 0,  0,  0,  0,  0,  0,  0,  0,  0, 14,  1, 154, 253,\n",
"       90,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
"       0,  0],\n",
"     [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, 139, 253,\n",
"      190,  2,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
"       0,  0],\n",
"     [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  11, 190,\n",
"      253,  70,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,\n",
"       0,  0],\n",
"     [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  35,\n",
"      241, 225, 160, 108,  1,  0,  0,  0,  0,  0,  0,  0,  0,\n",
```

```
"        0,   0],\n",
"      [ 0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
"       81, 240, 253, 253, 119,  25,   0,   0,   0,   0,   0,   0,   0,\n",
"        0,   0],\n",
"      [ 0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
"        0,  45, 186, 253, 253, 150,  27,   0,   0,   0,   0,   0,   0,\n",
"        0,   0],\n",
"      [ 0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
"        0,   0,  16,  93, 252, 253, 187,   0,   0,   0,   0,   0,   0,\n",
"        0,   0],\n",
"      [ 0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
"        0,   0,   0,   0, 249, 253, 249,  64,   0,   0,   0,   0,   0,\n",
"        0,   0],\n",
"      [ 0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
"        0,  46, 130, 183, 253, 253, 207,   2,   0,   0,   0,   0,   0,\n",
"        0,   0],\n",
"      [ 0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  39,\n",
"      148, 229, 253, 253, 253, 250, 182,   0,   0,   0,   0,   0,   0,\n",
"        0,   0],\n",
"      [ 0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  24, 114, 221,\n",
"      253, 253, 253, 253, 201,  78,   0,   0,   0,   0,   0,   0,   0,\n",
"        0,   0],\n",
"      [ 0,   0,   0,   0,   0,   0,   0,   0,  23,  66, 213, 253, 253,\n",
"      253, 253, 198,  81,   2,   0,   0,   0,   0,   0,   0,   0,   0,\n",
"        0,   0],\n",
"      [ 0,   0,   0,   0,   0,   0,  18, 171, 219, 253, 253, 253, 253,\n",
"      195,  80,   9,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
"        0,   0],\n",
"      [ 0,   0,   0,   0,  55, 172, 226, 253, 253, 253, 253, 244, 133,\n",
"       11,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
"        0,   0],\n",
```

```
       "       [  0,   0,   0,   0, 136, 253, 253, 253, 212, 135, 132,  16,   0,\n",
       "          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
       "          0,   0],\n",
       "       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
       "          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
       "          0,   0],\n",
       "       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
       "          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
       "          0,   0],\n",
       "       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
       "          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,\n",
       "          0,   0]], dtype=uint8)"
      ]
     },
     "execution_count": 10,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "x_train[0]"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/",
     "height": 283
    },
```

"id": "tX9Ahd7K3Tqw",
"outputId": "f6774578-e322-4f98-c960-1f605bb20158"
},
"outputs": [
{
"data": {
"text/plain": [
"<matplotlib.image.AxesImage at 0x7fa9daf6bf90>"
]
},
"execution_count": 11,
"metadata": {},
"output_type": "execute_result"
},
{
"data": {
"image/png":
"iVBORw0KGgoAAAANSUhEUgAAAPsAAAD4CAYAAAAq5pAIAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzA
AALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yLjIsIGh0dHA6
Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAOfklEQVR4nO3df6zV9X3H8deby4+r+KMgExmQaRnrytqJ3S22
kTY0RqNICthpyjLLMrPbbbWXBxT/q2nSSxSzYoG6JLewymXEqc1YGLW4o2Z1VbqhTAEmYU5bGEXqL2
LIL/hvvfH/dLc6v1+zvV8v+d8D7yfj+TmnPt9n+/5vjm5L77nfD/n+/2YuwvAhW9E1Q0AaA7CDgRB2IEgCD
sQBGEHghjZzI2NtjHerrHHN3CQQygkd1Sk/aUPVCoXdzG6R9I+S2iT9s7svTz2+XWN1vd1YZJMAEjZ7d26t7
rfxZtYm6VuSbpU0Q9IiM5tR7/MBSdrj7m+6+ylJ6yXL6ctAGUrEvbJkn4+6Pd92bJfY2adZtZjZj2
ndbLA5gAU0fCj8e7e5e4d7t4xSmMavTkAOYqEfb+kn+kqYN+n5IItA9CCioT9T9VUnTzewaMxst6u6YuSNpbTFoCy
1T305u5nzGyJpO9JapO01t3/rd7i7Lcl1H5q0NVNvnPt6WrH9z2seT9WjYswN
BEHYgCMIOBEHYgSAIOxAEYQeCIOxAEIyzo5Czn/tEsp4aS19+1avJdY/1n03Wl3bdk6xP1o+S9WjYswNBE
HYgCMIOBEHYgSAIOxAEYQeCIOxAEIyzIyl1Prok/cOjjybrHx89c/YMoFHYz2y
vpiKSzks64e0cZTQEoXxl79s+5+9slPA+ABuIzOxBE0bC7Y2RYz6xqzqAWbWaWY9ZtZjZj2
Z/t7vvN7EpJm8zsv9z9xcEPcPcuSV2SdJmN94LXSlpEFnGg5ohRRCy+vxM7uTt2Nh1nNXgWu4nEYXfk3l5GexLGQCZE5yjD6lz+f6QV9lT6n8d21q+ZQw+cS9XRFMoXnTC+uq1+1wSavA9WZrk62ep7bDZ0fXr+4S
u6WdKOshoDUK4ib+MnSnrGzM49Gz49z3fd/d9L6QqlsVGjk/W2q65zd0NeMfAXzanNFyk9xYNn7LgTRzpDFP1yaVkhJjCP73hfkdQlH8lu
Qy1R12d39T0rUl9gKggRh6A4Ig7EAQhB0IAhOcb3AHb49fannHz787WS9v8afyBmlL/f8L9+Y
n1u7/KlXkuuiXOzZgSAIOxAEYQeCIOxAEIQdCIKwA0EQdiAIxtkvAG0fyr8k80f+emdDt33dyqXJ+tSnOE21
VbBnB4Ig7EAQhB0IAgrADQRB2IAjCDgRB2IEgGGe/AOz56ozc2oap9J+9ao3kvX02e5oJvb
sQBCEHQiCsANBEHYgCMIOBEHYgSAIOxAE4+zngV/+2+2aeT9a13PZKopqdU/unQ/U/unpE8n6yXnHkvX+I0eS
Omnt2M1tjZofMbZePNbJOZ7c5uxzW2TQBFDedEd/rN/unLf2f21/kBeAkNsLv7i5LZxVr3xhMHR
4vqS12f21khaU3BIwNz3QuAQtpWlBhEuqmsnyHY+c22IbmH4cbp9n7fDOs6mR6EO3eSubfxfvvwcdyTCjK2IoGNvcSG1bI8YMkS3Wd3mlYd8oaXF2f7GkDeW0A6BRarcOt/dt9Yt9MKpRu2lbMMWL+Kje/UdLWu/4wuLv6q5cTwwGwKnj0KXRn5IKsL4R5jl7CJrwbLsA22ZfiM0KXoJjIm8a+Z+Z2xSU

+a2d2S3pJ0ZyObjK7vY7mfkiRJF9no3FqtcfQ//6t7kvX2Iz9J1nH+qBl2d1+UU7qx5F4ANBBflwWCIOxAEI
QdCIKwA0EQdiAITnFtAccWXp+sdy9Ykaz366Lc2vNHP5pct/1ZhtaiYM8OBEHYgSAIOxAEYQeCIOxAEIQd
CIKwA0Ewzt4C2v7iYLI+ZWT+OLok9as/t7burU8m171ce5L1RrKRNf782tqKbaA//9RgP32q2HOfh9izA0E
QdiAIwg4EQdiBIAg7EARhB4Ig7EAQjLM3gY1Jz4Qz7bK3k/URsmT9hm1/lFsbd9vu5Lq1tI1LT9B7dPb0ZP
1/P5M/Vn77TT9OrvvAlZuT9Vqvy7PHLsut/dMdn0+u27/t9WT9fMSeHQiCsANBEHYgCMIOBEHYgSAIOx
AEYQeCYJy9Cd6dNzNZXzX128l6/tnqA85umJBbG/nh08l133jgQ8n633c8k6wvHPt8sp4aC+9Xeirqom67
+J3c2v3L0tu+akHZ3VSv5p7dzNaY2SEz2zFo2TIz1tE0BRw3kb/7ikW4ZY/oi7z8x+niu3LQBlq
xl2d39RUl8TegHQQEUO0C0xs+3Z2/zcL1CbWaeZ9ZhZz2mdLLA5AEXUG/aVkqZJmimpV9JDeQ909y537
3D3jlFKnxACoHHqCru7H3T3s+7eL2m1pFnltgWgbHWF3cwmDfp1oaQdeY8F0BpqjrOb2ROS5kiaYGb7J
N0vaY6ZzZTkkvZK+nIDezzv9S062tDnP9ueP5bd9630tdd3/f5jhba9ZP/sZH3L6vR3DFJGLvxFsv7ytU/W/d
z3/u6mZH2dptT93K2qZtjdfdEQi4v9hQBoOr4uCwRB2IEgCDsQBGEHgiDsQBCc4toEl1yU/ppwrUsibzqe
nrJ58r+9mVuz29PPXXWvbK/o+kqz/7LPpU0WvOJG+XHTK/0z7dPoB16bLqX/bsu//YXLd6Xol/eTnIfbsQB
CEHQiCsANBEHYgCMIOBEHYgSAIOxAE4+xNsGTaC8l6rUsqr9o/J1Ml+9g9D2Ml8g1z/zJDH/+z8S0lf8eDPkuuuuzo9HfRxP
5Ws37n79tyaz8ufzlmS+o829vLfjbLZu3XY+4Y8kZ89OxAEYQeCIOxAEIQdCIKwA0EQdiAIwg4EwTh7E4xo
b0/Wp/xHelrlVVN+mKy/krgs/bWT9/ckazvnZfufc1tq3Nrn2lPf3+g1nn+v/edJcn6NX9T7
XcMqlBonN3MpprZC2b2upntNLOl2fLLZu3XY+4Y8kZ89OxAEYQeCIOxAEIQdCIKwA0EQdiAIwg4EwTh7E4xo
yV1Z78DaFE1w+7uve6+Nbt/RNIuSZMlkLGtUkgOI+0DXozOxqSddJ2ixporv3ZqUDkibmrN
MpqVOS2nVxvX0CKGjYR+PN7BJJT0u6x90PFG+IY+muHuXu3e4e8co5Ff5+7
fyxYfNLNJWX2SpEONaRFAPGWq+jTczk/SYpF3u/vCg0kZJi7PHiyU9V//2AJSlltf6YtqH+qNLO4
z9v9zaU+tXJtc91n82Wb90RHrq4otsdLKesvqdqcn6gz+6NVn/6Io3kvX0vyye4Xxmv0HSXZJeM7Nzg7Jf0
0DInzSzuyW9JenOxrQIoAw1w1w+7uL0m5s9rH+4YMcJ7i67JAEIQdCIKwA0EQdiAIwg4EwSmu54G235mW
rB+48crc2sh5byfXnrsnsLmwLiUNgLADURB2IEgCDsQBGEHgiDsQBCEHgiDsQBCDsQBGEHgiDsQBCEHQiCsANBEHYgCMIOBEHYgSAI
OxAEYQeCqBl2M5tqZi+Y2etm9vO2kG0ZLmmK3mmtj2Y2etmttPMlmbLl5nZfjmV7jMbv3PbPVcX3Xy7AOo1ml923tmlkraY2aaS9WY9ZtZ2Wic9Wis9oi7r2hc
ewDKMpz52Xsl9Wb3j5jZLkmTG90YgHJ9oM/sZna1pYOskbc4WLDaoSkbc4WLTGz7Wa2xszG5azTaWY9ZtZzWicLNQu
gfsMOu5ldIulpSfe4+2FJKyVNlzRTAyv+t4ZaZ2zLWFVmXzFGlmqUwxmz6X8PpTtjJJrVjz/VzYz6WY9WtaGWsQLl2GJaFhBAmZhBlE7+VcLlpGwAOPjHFGJGDIDfJcRSvjUYUgoWjdCEvJkr/q
bd+yWtljSrcW0CKGo4R+NN0mOSdrn7w4OWTwzv4Z5Z7NSMJQdSwCaasmIFioCyDOdo/A2S7L4pL0mTcSq3Dl0PUy1MzPZ9Tc1lIm5spyS
XtlfTlhnQIoBTDORr/kX/qShrkP9XPntAGkEHBEHYgSAIOxAEYQeCIOxAEIQdCIKwA0EQdiAIwg4EQdiBI
Ag7EARhB4Ig7EAQhB0Iwty9eRtrykYW66MYtMG0am+t2pdGvfmYeb9fcvfGKrQ1LCl/b+NmPe
7eUVkDCa3aWv6v2JdFbvZrG2/jgSAIOxBTq9V7dgBNQtiBIAg7EARhB4Ig7EAQhB0IgrADQRB2IAjCDgRB2IEgCDsQBGEHgiDsQBCEHQiCsANBEHYgCMIOBEHYgSAIOxAEYQeCIOxAEIQdCIKwA0EQdiAI
oJu5ndYmZvmNkeM7uvih7ymNleLprrJsk7ZITPbMcDMpj6a6WZBlX0WDmWWS6ZeqlnyS6p87L8NK1X4aTtTpmbuQazf06MmZmmt0k1TU++Ahv8gNzPhnNpsp9P6LiVSRU31S46nNp3Oqr
pa9d1dOfN/0zu5m1SfqppJsk7ZP0qqRF4N+oFk8VLvZYWZ7Tz/VcDCzz0p6V5J33P33VbJQOJzGcQC54XQSFV93CXs3rX7cT8zZt3XS1rNrm45f/+IuXYsdwQFuvhYsR+cxvGAFk4ZEz2oZoC8AJUMGKwq4l
j3P2rLdLbMknvVj2NdzZb0aTB04xLWiDpT1T5ogOm/0Nl34XGvuQSt7UpeyYfuVsm+JDinBjppdU9XVyYs8+8JD3/T3U9JWi9pfgV9tx1fFW33s
Wz5e0Nru/VgN/LE2X01tLcPded9a3T8i6dw045W+dom+mqKKsE+W9PNbNBv+9Ta8337pJ+YGZbzKyz6K6a7z6m
aGMNHde7P7ByRNrLKZdScxruZ3jPNeMu8dvVMf14UB+jeb7a7f0LSrZK+ZK+kr1dbUk+8BmslcZOhzWNd7
MMMc34r1T52tU7/XlRVYR9v6Spg36fki8Tg36fVpvIeaZlwt8NpVO
f15FWF/VdJ0M7vGzGGKkjRX08T5mNjY7cCIzGyvpZrXeVNQbNXSbJS3O3i+W9GyFvVzKp+yzs
f/tzdm/4jaa4Gjsj/t6SvV9FDTl8flvSf2c/OqnuT9lQG3tad1sCxjbslXSG3/8AsCZ
V1NtsDbxF3y5pW/Yzt+rXLtFXU143vi4LBMEOBEHYgSAIOxAEYQeCIOxAEIQdCIKwA0EQdiBIAg7EARhB4Ig7EAQhB0I4v8BJ99UDMnd
G+wAAAAASUVORK5CYII=\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

```
    "metadata": {},
    "output_type": "display_data"
   }
  ],
  "source": [
   "plt.imshow(x_train[6000])     #ploting the index=image"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "colab": {
    "base_uri": "https://localhost:8080/"
   },
   "id": "FXsFBkr73awY",
   "outputId": "d756fb7e-c167-454c-8455-de62c4be0596"
  },
  "outputs": [
   {
    "data": {
     "text/plain": [
      "0"
     ]
    },
    "execution_count": 12,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
```

```
    "np.argmax(y_train[6000])"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "X5uXzeX83fGz"
   },
   "source": [
    "Reshaping Dataset"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "eXT4spCl3hMg"
   },
   "outputs": [],
   "source": [
    "#Reshaping to format which CNN expects (batch, height, width, channels)\n",
    "x_train=x_train.reshape (60000, 28, 28, 1).astype('float32')\n",
    "x_test=x_test.reshape (10000, 28, 28, 1).astype ('float32')"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "lPZ-5k_d3muT"
   },
   "source": [
```

```
    "Applying One Hot Encoding"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "6DZXVKJW3ni8"
   },
   "outputs": [],
   "source": [
    "number_of_classes = 10  #storing the no of classes in a variable"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "Z-PimWtG3rK1"
   },
   "outputs": [],
   "source": [
    "y_train = np_utils.to_categorical (y_train, number_of_classes) #converts the output in binary format\n",
    "y_test = np_utils.to_categorical (y_test, number_of_classes)"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "vtQbVscm3yy4"
```

```
  },
  "source": [
    "Add CNN Layers"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "id": "GqtjDvPm30ex"
  },
  "outputs": [],
  "source": [
    "#create model\n",
    "model=Sequential ()"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "id": "8yEPHzeS35Xk"
  },
  "outputs": [],
  "source": [
    "#adding modeL Layer\n",
    "model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation='relu'))\n",
    "model.add(Conv2D(32, (3, 3), activation = 'relu'))"
  ]
 },
 {
```

```
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "id": "AgY3Vovx37QU"
  },
  "outputs": [],
  "source": [
    "#flatten the dimension of the image\n",
    "model.add(Flatten())"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "id": "XrsIoODR4BK6"
  },
  "outputs": [],
  "source": [
    "#output layer with 10 neurons\n",
    "model.add(Dense(number_of_classes,activation = 'softmax'))"
  ]
 },
 {
  "cell_type": "markdown",
  "metadata": {
    "id": "vqndtKCH4IPl"
  },
  "source": [
    "Compiling the model"
  ]
```

```
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "id": "GZv1gJHo4JfJ"
 },
 "outputs": [],
 "source": [
  "#Compile model\n",
  "model.compile(loss= 'categorical_crossentropy', optimizer=\"Adam\", metrics=['accuracy'])"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "id": "Pm4PobP74Pnn"
 },
 "outputs": [],
 "source": [
  "x_train = np.asarray(x_train)\n",
  "y_train = np.asarray(y_train)"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "e-QxweTG4TKU"
 },
 "source": [
```

```
    "Train the model"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "o5GnPJfw4T7Z",
    "outputId": "bce15ea9-d799-4a9d-cb39-7b5386c4f095"
   },
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Epoch 1/5\n",
      "1875/1875 [==============================] - 181s 96ms/step - loss: 0.2721 - accuracy: 0.9494 - val_loss: 0.0847 - val_accuracy: 0.9749\n",
      "Epoch 2/5\n",
      "1875/1875 [==============================] - 174s 93ms/step - loss: 0.0688 - accuracy: 0.9793 - val_loss: 0.0935 - val_accuracy: 0.9713\n",
      "Epoch 3/5\n",
      "1875/1875 [==============================] - 181s 96ms/step - loss: 0.0441 - accuracy: 0.9864 - val_loss: 0.0781 - val_accuracy: 0.9777\n",
      "Epoch 4/5\n",
      "1875/1875 [==============================] - 179s 95ms/step - loss: 0.0328 - accuracy: 0.9899 - val_loss: 0.0893 - val_accuracy: 0.9769\n",
      "Epoch 5/5\n",
      "1875/1875 [==============================] - 181s 96ms/step - loss: 0.0278 - accuracy: 0.9920 - val_loss: 0.0975 - val_accuracy: 0.9796\n"
```

```
     ]
    },
    {
     "data": {
      "text/plain": [
       "<keras.callbacks.History at 0x7fa9d5f699d0>"
      ]
     },
     "execution_count": 22,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#fit the model\n",
    "model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=5, batch_size=32)"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "5pwwJyNEPLbq"
   },
   "source": [
    "Observing the metrics"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
```

```
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "xQ90P4HIPMg4",
    "outputId": "1f16739b-1e9d-47bb-be64-e8f9ea45e38f"
   },
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Metrics (Test loss &Test Accuracy) : \n",
      "[0.09751885384321213, 0.9796000123023987]\n"
     ]
    }
   ],
   "source": [
    "# Final evaluation of the model\n",
    "metrics = model.evaluate(x_test, y_test, verbose=0)\n",
    "print(\"Metrics (Test loss &Test Accuracy) : \")\n",
    "print(metrics)"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "_P4RZ60TPgdQ"
   },
   "source": [
    "Test The Model"
   ]
```

```
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "SPXzYL6TPhnu",
  "outputId": "568b9976-9d7f-433c-a9f2-48783d154b26"
 },
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "1/1 [==============================] - 0s 130ms/step\n",
    "[[1.67997398e-12 6.91265110e-17 9.30522148e-09 1.37389069e-08\n",
    "  5.37760243e-05 4.41636274e-07 1.10186464e-13 5.36197012e-06\n",
    "  5.49552851e-07 9.99939799e-01]]\n"
   ]
  }
 ],
 "source": [
  "prediction=model.predict(x_test[6000:6001])\n",
  "print(prediction)"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
```

```
 "metadata": {
  "id": "MfNpkgicPudX"
 },
 "outputs": [],
 "source": [
  "plt.imshow(x_test[6000])"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "FiET6KRePunj",
  "outputId": "1f66a8f4-8e5c-4e8c-cfba-168a885c9f83"
 },
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "[9]\n"
   ]
  }
 ],
 "source": [
  "import numpy as np\n",
  "print(np.argmax(prediction, axis=1)) #printing our Labels from first 4 images"
 ]
```

```
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "pIV2UYC5P2gv",
  "outputId": "59b74137-b352-478e-8715-bb1aa3f39ca7"
 },
 "outputs": [
  {
   "data": {
    "text/plain": [
     "9"
    ]
   },
   "execution_count": 28,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "np.argmax(y_test[6000:6001]) #printing the actual labels"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "aBXoWG83P72w"
```

```
  },
  "source": [
    "Save The model"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "id": "KLmxEYIiP8sU"
  },
  "outputs": [],
  "source": [
    "# Save the model\n",
    "model.save('models/mnistCNN.h5')"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "Er8o33VIP_m-",
    "outputId": "a07858dc-4d7e-4be0-d512-e2cf8d6cf2e5"
  },
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
```

```
      "text": [
        "/content/models\n"
      ]
    }
  ],
  "source": [
    "cd models"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "beTQP3D1QJDz",
    "outputId": "5c416348-80d5-4bb4-bb87-4e2c0b1500d1"
  },
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "mnistCNN.h5\n"
      ]
    }
  ],
  "source": [
    "!tar -zcvf handwritten-digit-recognition-model_new.tgz mnistCNN.h5"
  ]
```

```
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "ypUQT_-KQN0q",
    "outputId": "a5d24ba1-2a4e-46e9-98b1-64892cbe1e2a"
   },
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/\n",
      "Collecting watson-machine-learning-client\n",
      "  Downloading watson_machine_learning_client-1.0.391-py3-none-any.whl (538 kB)\n",
      "\u001b[K     |████████████████████████████████| 538 kB 5.1 MB/s \n",
      "\u001b[?25hRequirement already satisfied: tabulate in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (0.8.10)\n",
      "Collecting ibm-cos-sdk\n",
      "  Downloading ibm-cos-sdk-2.12.0.tar.gz (55 kB)\n",
      "\u001b[K     |████████████████████████████████| 55 kB 1.9 MB/s \n",
      "\u001b[?25hRequirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (4.64.1)\n",
      "Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (2022.9.24)\n",
      "Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (1.3.5)\n",
      "Collecting lomond\n",
```

" Downloading lomond-0.3.3-py2.py3-none-any.whl (35 kB)\n",

"Collecting boto3\n",

" Downloading boto3-1.26.3-py3-none-any.whl (132 kB)\n",

"\u001b[K || 132 kB 23.6 MB/s \n",

"\u001b[?25hRequirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (2.23.0)\n",

"Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages (from watson-machine-learning-client) (1.24.3)\n",

"Collecting botocore<1.30.0,>=1.29.3\n",

" Downloading botocore-1.29.3-py3-none-any.whl (9.8 MB)\n",

"\u001b[K || 9.8 MB 33.8 MB/s \n",

"\u001b[?25hCollecting s3transfer<0.7.0,>=0.6.0\n",

" Downloading s3transfer-0.6.0-py3-none-any.whl (79 kB)\n",

"\u001b[K || 79 kB 4.3 MB/s \n",

"\u001b[?25hCollecting jmespath<2.0.0,>=0.7.1\n",

" Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)\n",

"Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/python3.7/dist-packages (from botocore<1.30.0,>=1.29.3->boto3->watson-machine-learning-client) (2.8.2)\n",

"Collecting urllib3\n",

" Downloading urllib3-1.26.12-py2.py3-none-any.whl (140 kB)\n",

"\u001b[K || 140 kB 10.6 MB/s \n",

"\u001b[?25hRequirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.30.0,>=1.29.3->boto3->watson-machine-learning-client) (1.15.0)\n",

"Collecting ibm-cos-sdk-core==2.12.0\n",

" Downloading ibm-cos-sdk-core-2.12.0.tar.gz (956 kB)\n",

"\u001b[K || 956 kB 59.3 MB/s \n",

"\u001b[?25hCollecting ibm-cos-sdk-s3transfer==2.12.0\n",

" Downloading ibm-cos-sdk-s3transfer-2.12.0.tar.gz (135 kB)\n",

"\u001b[K || 135 kB 25.5 MB/s \n",

"\u001b[?25hCollecting jmespath<2.0.0,>=0.7.1\n",

" Downloading jmespath-0.10.0-py2.py3-none-any.whl (24 kB)\n",

"Collecting requests\n",

" Downloading requests-2.28.1-py3-none-any.whl (62 kB)\n",

"\u001b[K |████████████████████████████████| 62 kB 331 kB/s \n",

"\u001b[?25hRequirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->watson-machine-learning-client) (2.10)\n",

"Requirement already satisfied: charset-normalizer<3,>=2 in /usr/local/lib/python3.7/dist-packages (from requests->watson-machine-learning-client) (2.1.1)\n",

"Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas->watson-machine-learning-client) (2022.5)\n",

"Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-packages (from pandas->watson-machine-learning-client) (1.21.6)\n",

"Building wheels for collected packages: ibm-cos-sdk, ibm-cos-sdk-core, ibm-cos-sdk-s3transfer\n",

" Building wheel for ibm-cos-sdk (setup.py) ... \u001b[?25l\u001b[?25hdone\n",

" Created wheel for ibm-cos-sdk: filename=ibm_cos_sdk-2.12.0-py3-none-any.whl size=73930 sha256=8b6b81d80325b3bc916e20c39e55bf1ce1a8aca356cd4c96ede13f8f77846306\n",

" Stored in directory: /root/.cache/pip/wheels/ec/94/29/2b57327cf00664b6614304f7958abd29d77ea0e5bbece2ea57\n",

" Building wheel for ibm-cos-sdk-core (setup.py) ... \u001b[?25l\u001b[?25hdone\n",

" Created wheel for ibm-cos-sdk-core: filename=ibm_cos_sdk_core-2.12.0-py3-none-any.whl size=562962 sha256=e20fa3d232ff5fd914dc839e0e5893c76ef26f6c87cfc032d134632755ce8403\n",

" Stored in directory: /root/.cache/pip/wheels/64/56/fb/5cd6f4f40406c828a5289b95b2752a4d142a9afb359244ed8d\n",

" Building wheel for ibm-cos-sdk-s3transfer (setup.py) ... \u001b[?25l\u001b[?25hdone\n",

" Created wheel for ibm-cos-sdk-s3transfer: filename=ibm_cos_sdk_s3transfer-2.12.0-py3-none-any.whl size=89778 sha256=e4baf912d28ff427b346f9efe07a9dd0e1f4f95d42f2f81efbcf5a5bb89633f5\n",

" Stored in directory: /root/.cache/pip/wheels/57/79/6a/ffe3370ed7ebc00604f9f76766e1e0348dcdcad2b2e32df9e1\n",

"Successfully built ibm-cos-sdk ibm-cos-sdk-core ibm-cos-sdk-s3transfer\n",

"Installing collected packages: urllib3, requests, jmespath, ibm-cos-sdk-core, botocore, s3transfer, ibm-cos-sdk-s3transfer, lomond, ibm-cos-sdk, boto3, watson-machine-learning-client\n",

" Attempting uninstall: urllib3\n",

"   Found existing installation: urllib3 1.24.3\n",

"   Uninstalling urllib3-1.24.3:\n",

"   Successfully uninstalled urllib3-1.24.3\n",

```
        "  Attempting uninstall: requests\n",
        "    Found existing installation: requests 2.23.0\n",
        "    Uninstalling requests-2.23.0:\n",
        "      Successfully uninstalled requests-2.23.0\n",
        "Successfully installed boto3-1.26.3 botocore-1.29.3 ibm-cos-sdk-2.12.0 ibm-cos-sdk-core-2.12.0 ibm-cos-sdk-s3transfer-2.12.0 jmespath-0.10.0 lomond-0.3.3 requests-2.28.1 s3transfer-0.6.0 urllib3-1.26.12 watson-machine-learning-client-1.0.391\n"
       ]
      }
     ],
     "source": [
      "!pip install watson-machine-learning-client --upgrade"
     ]
    },
    {
     "cell_type": "markdown",
     "source": [
      "Cloud deploy"
     ],
     "metadata": {
      "id": "DFqAO13R4lsf"
     }
    },
    {
     "cell_type": "code",
     "source": [
      "from ibm_watson_machine_learning import APIClient\n",
      "credentials ={\n",
      "    \"url\":\"https://jp-tok.ml.cloud.ibm.com\",\n",
      "    \"apikey\":\"BHyalu2c7JN6n9cnvAVULvSKRYFVLMQ_m51toZ9Yk0nS\"\n",
      "}\n",
      "client = APIClient(credentials)\n",
```

```
      "client"
     ],
     "metadata": {
      "id": "LL2d3Xxi4nzy"
     },
     "execution_count": null,
     "outputs": []
    },
    {
     "cell_type": "code",
     "source": [
      "client.spaces.get_details()"
     ],
     "metadata": {
      "id": "x0YTN8JC4y8i"
     },
     "execution_count": null,
     "outputs": []
    },
    {
     "cell_type": "code",
     "source": [
      "{'resources': [{'entity': {'compute': [{'crn': 'crn:v1:bluemix:public:pm-20:jp-
tok:a/53f9f6400d0d44889534e8abcd2dfe39:0f4376b6-c944-4b27-b23e-48b54d8f4bbd::',\n",
      "      'guid': '0f4376b6-c944-4b27-b23e-48b54d8f4bbd',\n",
      "      'name': 'Watson Machine Learning-sp',\n",
      "      'type': 'machine_learning'}],\n",
      "    'description': '',\n",
      "    'name': 'digitrecognition',\n",
      "    'scope': {'bss_account_id': '53f9f6400d0d44889534e8abcd2dfe39'},\n",
      "    'stage': {'production': False},\n",
```

```
"    'status': {'state': 'active'},\n",
"    'storage': {'properties': {'bucket_name': '63888f6f-d1ef-475c-a8d8-a2e4957bb673',\n",
"     'bucket_region': 'jp-tok-standard',\n",
"     'credentials': {'admin': {'access_key_id': '834b3358ebb945fb9ebbb4020cd2bf0e',\n",
"      'api_key': '2JONUuuPfYzZzPGzTp1J7dwwjNTpkOsyxdW5gx_vml3m',\n",
"      'secret_access_key': '1ed5b29fdd6c65b48ca72963b6177133ce51a7b23acdcaa5',\n",
"      'service_id': 'ServiceId-a2495f73-f36b-4fa1-9991-976f110c1a4f'},\n",
"     'editor': {'access_key_id': 'b56d445c54794369b2a4e0115e166605',\n",
"      'api_key': 'wcwCBLp8z4xpgnsEDeUCOZquAovrWhXu2wcF9Kz5Vhpe',\n",
"      'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-
storage:global:a/53f9f6400d0d44889534e8abcd2dfe39:d8fa8aee-cd61-4757-9543-
a61f55971074::',\n",
"      'secret_access_key': '84b0b128f52e57c025e6517604a06212b8d19f0b349eeea3',\n",
"      'service_id': 'ServiceId-4e1f87ab-27bc-4654-b6ea-667a8640c7e0'},\n",
"     'viewer': {'access_key_id': '558109e942fb4b1eb020c881f04d8588',\n",
"      'api_key': 'zWS-VZ_d9GfkDt1XnCmWoOA6liYXNnGtrPwJt2fI0UI5',\n",
"      'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-
storage:global:a/53f9f6400d0d44889534e8abcd2dfe39:d8fa8aee-cd61-4757-9543-
a61f55971074::',\n",
"      'secret_access_key': '3e2d27ab9d4041707cfa721daa638d1ad57f42ab8df94c09',\n",
"      'service_id': 'ServiceId-93177c88-86e2-470d-b5bf-3aed99d093a8'}},\n",
"    'endpoint_url': 'https://s3.jp-tok.cloud-object-storage.appdomain.cloud',\n",
"    'guid': 'd8fa8aee-cd61-4757-9543-a61f55971074',\n",
"    'resource_crn': 'crn:v1:bluemix:public:cloud-object-
storage:global:a/53f9f6400d0d44889534e8abcd2dfe39:d8fa8aee-cd61-4757-9543-
a61f55971074::'},\n",
"    'type': 'bmcos_object_storage'}},\n",
"   'metadata': {'created_at': '2022-10-31T10:33:07.575Z',\n",
"    'creator_id': 'IBMid-667000CZ2Y',\n",
"    'id': 'aa24227a-9f01-493f-90e6-1b6132057fc6',\n",
"    'updated_at': '2022-10-31T10:33:25.148Z',\n",
"    'url': '/v2/spaces/aa24227a-9f01-493f-90e6-1b6132057fc6'}}]}"
  ],
```

```
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "Ep-e3MEG5BGa",
    "outputId": "079fa6cd-8ef3-40c4-ea56-5d3506aa8e39"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "{'resources': [{'entity': {'compute': [{'crn': 'crn:v1:bluemix:public:pm-20:jp-tok:a/53f9f6400d0d44889534e8abcd2dfe39:0f4376b6-c944-4b27-b23e-48b54d8f4bbd::',\n",
          "     'guid': '0f4376b6-c944-4b27-b23e-48b54d8f4bbd',\n",
          "     'name': 'Watson Machine Learning-sp',\n",
          "     'type': 'machine_learning'}],\n",
          "   'description': '',\n",
          "   'name': 'digitrecognition',\n",
          "   'scope': {'bss_account_id': '53f9f6400d0d44889534e8abcd2dfe39'},\n",
          "   'stage': {'production': False},\n",
          "   'status': {'state': 'active'},\n",
          "   'storage': {'properties': {'bucket_name': '63888f6f-d1ef-475c-a8d8-a2e4957bb673',\n",
          "     'bucket_region': 'jp-tok-standard',\n",
          "     'credentials': {'admin': {'access_key_id': '834b3358ebb945fb9ebbb4020cd2bf0e',\n",
          "       'api_key': '2JONUuuPfYzZzPGzTp1J7dwwjNTpkOsyxdW5gx_vml3m',\n",
          "       'secret_access_key': '1ed5b29fdd6c65b48ca72963b6177133ce51a7b23acdcaa5',\n",
          "       'service_id': 'ServiceId-a2495f73-f36b-4fa1-9991-976f110c1a4f'},\n",
          "      'editor': {'access_key_id': 'b56d445c54794369b2a4e0115e166605',\n",
          "       'api_key': 'wcwCBLp8z4xpgnsEDeUCOZquAovrWhXu2wcF9Kz5Vhpe',\n",
```

```
"       'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/53f9f6400d0d44889534e8abcd2dfe39:d8fa8aee-cd61-4757-9543-a61f55971074::',\n",
"       'secret_access_key': '84b0b128f52e57c025e6517604a06212b8d19f0b349eeea3',\n",
"       'service_id': 'ServiceId-4e1f87ab-27bc-4654-b6ea-667a8640c7e0'},\n",
"      'viewer': {'access_key_id': '558109e942fb4b1eb020c881f04d8588',\n",
"       'api_key': 'zWS-VZ_d9GfkDt1XnCmWoOA6liYXNnGtrPwJt2fI0UI5',\n",
"       'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/53f9f6400d0d44889534e8abcd2dfe39:d8fa8aee-cd61-4757-9543-a61f55971074::',\n",
"       'secret_access_key': '3e2d27ab9d4041707cfa721daa638d1ad57f42ab8df94c09',\n",
"       'service_id': 'ServiceId-93177c88-86e2-470d-b5bf-3aed99d093a8'}},\n",
"     'endpoint_url': 'https://s3.jp-tok.cloud-object-storage.appdomain.cloud',\n",
"     'guid': 'd8fa8aee-cd61-4757-9543-a61f55971074',\n",
"     'resource_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/53f9f6400d0d44889534e8abcd2dfe39:d8fa8aee-cd61-4757-9543-a61f55971074::'},\n",
"     'type': 'bmcos_object_storage'}},\n",
"   'metadata': {'created_at': '2022-10-31T10:33:07.575Z',\n",
"    'creator_id': 'IBMid-667000CZ2Y',\n",
"    'id': 'aa24227a-9f01-493f-90e6-1b6132057fc6',\n",
"    'updated_at': '2022-10-31T10:33:25.148Z',\n",
"    'url': '/v2/spaces/aa24227a-9f01-493f-90e6-1b6132057fc6'}}]}"
        ]
       },
       "metadata": {},
       "execution_count": 5
      }
     ]
    },
    {
     "cell_type": "code",
     "source": [
```

```
    "def guid_from_space_name(client,deploy):\n",
    "  space = client.spaces.get_details()\n",
    "  return (next(item for item in space['resources'] if item['entity']['name']==deploy)['metadata']['id'])"
   ],
   "metadata": {
    "id": "maexT8Xi5czb"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "space_uid = guid_from_space_name(client,'digitrecognition')\n",
    "print(\"Space UID = \" + space_uid)"
   ],
   "metadata": {
    "id": "Fz2uEcor5fGl"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "markdown",
   "source": [
    "Space UID = aa24227a-9f01-493f-90e6-1b6132057fc6"
   ],
   "metadata": {
    "id": "3zRvX7A95nrU"
   }
```

```
},
{
 "cell_type": "code",
 "source": [
  "client.set.default_space(space_uid)"
 ],
 "metadata": {
  "id": "PZE3NQbW6M84"
 },
 "execution_count": null,
 "outputs": []
},
{
 "cell_type": "markdown",
 "source": [
  "'SUCCESS'"
 ],
 "metadata": {
  "id": "L3ktbOe86Xt_"
 }
},
{
 "cell_type": "code",
 "source": [
  "client.software_specifications.list(limit=100)"
 ],
 "metadata": {
  "id": "_gxq-gC46ex0"
 },
 "execution_count": null,
 "outputs": []
```

```
},
{
 "cell_type": "code",
 "source": [
  "------------------------------  ----------------------------------  ----\n",
  "NAME                        ASSET_ID                          TYPE\n",
  "default_py3.6               0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base\n",
  "kernel-spark3.2-scala2.12       020d69ce-7ac1-5e68-ac1a-31189867356a  base\n",
  "pytorch-onnx_1.3-py3.7-edt      069ea134-3346-5748-b513-49120e15d288  base\n",
  "scikit-learn_0.20-py3.6         09c5a1d0-9c1e-4473-a344-eb7b665ff687  base\n",
  "spark-mllib_3.0-scala_2.12      09f4cff0-90a7-5899-b9ed-1ef348aebdee  base\n",
  "pytorch-onnx_rt22.1-py3.9       0b848dd4-e681-5599-be41-b5f6fccc6471  base\n",
  "ai-function_0.1-py3.6           0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base\n",
  "shiny-r3.6                  0e6e79df-875e-4f24-8ae9-62dcc2148306  base\n",
  "tensorflow_2.4-py3.7-horovod    1092590a-307d-563d-9b62-4eb7d64b3f22  base\n",
  "pytorch_1.1-py3.6             10ac12d6-6b30-4ccd-8392-3e922c096a92  base\n",
  "tensorflow_1.15-py3.6-ddl       111e41b3-de2d-5422-a4d6-bf776828c4b7  base\n",
  "runtime-22.1-py3.9            12b83a17-24d8-5082-900f-0ab31fbfd3cb  base\n",
  "scikit-learn_0.22-py3.6         154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base\n",
  "default_r3.6                1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base\n",
  "pytorch-onnx_1.3-py3.6          1bc6029a-cc97-56da-b8e0-39c3880dbbe7  base\n",
  "kernel-spark3.3-r3.6          1c9e5454-f216-59dd-a20e-474a5cdf5988  base\n",
  "pytorch-onnx_rt22.1-py3.9-edt   1d362186-7ad5-5b59-8b6c-9d0880bde37f  base\n",
  "tensorflow_2.1-py3.6          1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base\n",
  "spark-mllib_3.2             20047f72-0a98-58c7-9ff5-a77b012eb8f5  base\n",
  "tensorflow_2.4-py3.8-horovod    217c16f6-178f-56bf-824a-b19f20564c49  base\n",
  "runtime-22.1-py3.9-cuda         26215f05-08c3-5a41-a1b0-da66306ce658  base\n",
  "do_py3.8                    295addb5-9ef9-547e-9bf4-92ae3563e720  base\n",
  "autoai-ts_3.8-py3.8           2aa0c932-798f-5ae9-abd6-15e0c2402fb5  base\n",
  "tensorflow_1.15-py3.6          2b73a275-7cbf-420b-a912-eae7f436e0bc  base\n",
  "kernel-spark3.3-py3.9          2b7961e2-e3b1-5a8c-a491-482c8368839a  base\n",
```

"pytorch_1.2-py3.6 2c8ef57d-2687-4b7d-acce-01f94976dac1 base\n",

"spark-mllib_2.3 2e51f700-bca0-4b0d-88dc-5c6791338875 base\n",

"pytorch-onnx_1.1-py3.6-edt 32983cea-3f32-4400-8965-dde874a8d67e base\n",

"spark-mllib_3.0-py37 36507ebe-8770-55ba-ab2a-eafe787600e9 base\n",

"spark-mllib_2.4 390d21f8-e58b-4fac-9c55-d7ceda621326 base\n",

"xgboost_0.82-py3.6 39e31acd-5f30-41dc-ae44-60233c80306e base\n",

"pytorch-onnx_1.2-py3.6-edt 40589d0e-7019-4e28-8daa-fb03b6f4fe12 base\n",

"default_r36py38 41c247d3-45f8-5a71-b065-8580229facf0 base\n",

"autoai-ts_rt22.1-py3.9 4269d26e-07ba-5d40-8f66-2d495b0c71f7 base\n",

"autoai-obm_3.0 42b92e18-d9ab-567f-988a-4240ba1ed5f7 base\n",

"pmml-3.0_4.3 493bcb95-16f1-5bc5-bee8-81b8af80e9c7 base\n",

"spark-mllib_2.4-r_3.6 49403dff-92e9-4c87-a3d7-a42d0021c095 base\n",

"xgboost_0.90-py3.6 4ff8d6c2-1343-4c18-85e1-689c965304d3 base\n",

"pytorch-onnx_1.1-py3.6 50f95b2a-bc16-43bb-bc94-b0bed208c60b base\n",

"autoai-ts_3.9-py3.8 52c57136-80fa-572e-8728-a5e7cbb42cde base\n",

"spark-mllib_2.4-scala_2.11 55a70f99-7320-4be5-9fb9-9edb5a443af5 base\n",

"spark-mllib_3.0 5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9 base\n",

"autoai-obm_2.0 5c2e37fa-80b8-5e77-840f-d912469614ee base\n",

"spss-modeler_18.1 5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b base\n",

"cuda-py3.8 5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e base\n",

"autoai-kb_3.1-py3.7 632d4b22-10aa-5180-88f0-f52dfb6444d7 base\n",

"pytorch-onnx_1.7-py3.8 634d3cdc-b562-5bf9-a2d4-ea90a478456b base\n",

"spark-mllib_2.3-r_3.6 6586b9e3-ccd6-4f92-900f-0f8cb2bd6f0c base\n",

"tensorflow_2.4-py3.7 65e171d7-72d1-55d9-8ebb-f813d620c9bb base\n",

"spss-modeler_18.2 687eddc9-028a-4117-b9dd-e57b36f1efa5 base\n",

"pytorch-onnx_1.2-py3.6 692a6a4d-2c4d-45ff-a1ed-b167ee55469a base\n",

"spark-mllib_2.3-scala_2.11 7963efe5-bbec-417e-92cf-0574e21b4e8d base\n",

"spark-mllib_2.4-py37 7abc992b-b685-532b-a122-a396a3cdbaab base\n",

"caffe_1.0-py3.6 7bb3dbe2-da6e-4145-918d-b6d84aa93b6b base\n",

"pytorch-onnx_1.7-py3.7 812c6631-42b7-5613-982b-02098e6c909c base\n",

"cuda-py3.6 82c79ece-4d12-40e6-8787-a7b9e0f62770 base\n",

"tensorflow_1.15-py3.6-horovod 8964680e-d5e4-5bb8-919b-8342c6c0dfd8 base\n",

"hybrid_0.1 8c1a58c6-62b5-4dc4-987a-df751c2756b6 base\n",

"pytorch-onnx_1.3-py3.7 8d5d8a87-a912-54cf-81ec-3914adaa988d base\n",

"caffe-ibm_1.0-py3.6 8d863266-7927-4d1e-97d7-56a7f4c0a19b base\n",

"spss-modeler_17.1 902d0051-84bd-4af6-ab6b-8f6aa6fdeabb base\n",

"do_12.10 9100fd72-8159-4eb9-8a0b-a87e12eefa36 base\n",

"do_py3.7 9447fa8b-2051-4d24-9eef-5acb0e3c59f8 base\n",

"spark-mllib_3.0-r_3.6 94bb6052-c837-589d-83f1-f4142f219e32 base\n",

"cuda-py3.7-opence 94e9652b-7f2d-59d5-ba5a-23a414ea488f base\n",

"nlp-py3.8 96e60351-99d4-5a1c-9cc0-473ac1b5a864 base\n",

"cuda-py3.7 9a44990c-1aa1-4c7d-baf8-c4099011741c base\n",

"hybrid_0.2 9b3f9040-9cee-4ead-8d7a-780600f542f7 base\n",

"spark-mllib_3.0-py38 9f7a8fc1-4d3c-5e65-ab90-41fa8de2d418 base\n",

"autoai-kb_3.3-py3.7 a545cca3-02df-5c61-9e88-998b09dc79af base\n",

"spark-mllib_3.0-py39 a6082a27-5acc-5163-b02c-6b96916eb5e0 base\n",

"runtime-22.1-py3.9-do a7e7dbf1-1d03-5544-994d-e5ec845ce99a base\n",

"default_py3.8 ab9e1b80-f2ce-592c-a7d2-4f2344f77194 base\n",

"tensorflow_rt22.1-py3.9 acd9c798-6974-5d2f-a657-ce06e986df4d base\n",

"kernel-spark3.2-py3.9 ad7033ee-794e-58cf-812e-a95f4b64b207 base\n",

"autoai-obm_2.0 with Spark 3.0 af10f35f-69fa-5d66-9bf5-acb58434263a base\n",

"default_py3.7_opence c2057dd4-f42c-5f77-a02f-72bdbd3282c9 base\n",

"tensorflow_2.1-py3.7 c4032338-2a40-500a-beef-b01ab2667e27 base\n",

"do_py3.7_opence cc8f8976-b74a-551a-bb66-6377f8d865b4 base\n",

"spark-mllib_3.3 d11f2434-4fc7-58b7-8a62-755da64fdaf8 base\n",

"autoai-kb_3.0-py3.6 d139f196-e04b-5d8b-9140-9a10ca1fa91a base\n",

"spark-mllib_3.0-py36 d82546d5-dd78-5fbb-9131-2ec309bc56ed base\n",

"autoai-kb_3.4-py3.8 da9b39c3-758c-5a4f-9cfd-457dd4d8c395 base\n",

"kernel-spark3.2-r3.6 db2fe4d6-d641-5d05-9972-73c654c60e0a base\n",

"autoai-kb_rt22.1-py3.9 db6afe93-665f-5910-b117-d879897404d9 base\n",

"tensorflow_rt22.1-py3.9-horovod dda170cc-ca67-5da7-9b7a-cf84c6987fae base\n",

"autoai-ts_1.0-py3.7 deef04f0-0c42-5147-9711-89f9904299db base\n",

```
    "tensorflow_2.1-py3.7-horovod    e384fce5-fdd1-53f8-bc71-11326c9c635f  base\n",
    "default_py3.7                e4429883-c883-42b6-87a8-f419d64088cd  base\n",
    "do_22.1                      e51999ba-6452-5f1f-8287-17228b88b652  base\n",
    "autoai-obm_3.2               eae86aab-da30-5229-a6a6-1d0d4e368983  base\n",
    "do_20.1                      f686cdd9-7904-5f9d-a732-01b0d6b10dc5  base\n",
    "scikit-learn_0.19-py3.6      f963fa9d-4bb7-5652-9c5d-8d9289ef6ad9  base\n",
    "tensorflow_2.4-py3.8         fe185c44-9a99-5425-986b-59bd1d2eda46  base\n",
    "------------------------------  ------------------------------------  ----"
   ],
   "metadata": {
    "id": "fZRCAfN8ZXLu"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "software_space_uid = client.software_specifications.get_uid_by_name('tensorflow_rt22.1-py3.9')\n",
    "software_space_uid"
   ],
   "metadata": {
    "id": "E4vi-b5UZe1h"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
```

```
    "'acd9c798-6974-5d2f-a657-ce06e986df4d'"
   ],
   "metadata": {
    "id": "Dd_szjfCZiDN"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "model_details = client.repository.store_model(model='handwritten-digit-recognition-model_new.tgz',meta_props={\n",
    "    client.repository.ModelMetaNames.NAME:\"CNN Digit recognition model\",\n",
    "    client.repository.ModelMetaNames.TYPE:\"tensorflow_2.7\",\n",
    "    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid\n",
    "})"
   ],
   "metadata": {
    "id": "HWiNGTEVZrOA"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "model_details"
   ],
   "metadata": {
    "id": "rO2X9OXxZu9y"
```

```
  },
  "execution_count": null,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "{'entity': {'hybrid_pipeline_software_specs': [],\n",
   "  'software_spec': {'id': 'acd9c798-6974-5d2f-a657-ce06e986df4d',\n",
   "   'name': 'tensorflow_rt22.1-py3.9'},\n",
   "  'type': 'tensorflow_2.7'},\n",
   " 'metadata': {'created_at': '2022-11-01T10:15:40.847Z',\n",
   "  'id': '97d463b1-45ee-47f7-b8af-aed338794ce1',\n",
   "  'modified_at': '2022-11-01T10:15:44.197Z',\n",
   "  'name': 'CNN Digit recognition model',\n",
   "  'owner': 'IBMid-667000CZ2Y',\n",
   "  'resource_key': '84636ddb-9fa8-47e4-8fa4-3c36731e2fe6',\n",
   "  'space_id': 'aa24227a-9f01-493f-90e6-1b6132057fc6'},\n",
   " 'system': {'warnings': []}}"
  ],
  "metadata": {
   "id": "yZ_WFrB3Zx1c"
  },
  "execution_count": null,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "model_id = client.repository.get_model_id(model_details)\n",
   "model_id"
```

```
  ],
  "metadata": {
    "id": "J_8uc-IwZ1TV"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "'97d463b1-45ee-47f7-b8af-aed338794ce1'"
  ],
  "metadata": {
    "id": "iet0ENFsZ-4j"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "client.repository.download(model_id,'DigitRecog_IBM_model.tar.gz')"
  ],
  "metadata": {
    "id": "nZoJKpAoaFJz"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "markdown",
```

```
  "source": [
    "Successfully saved model content to file: 'DigitRecog_IBM_model.tar.gz'"
  ],
  "metadata": {
    "id": "TvnPhcmaaIeS"
  }
 },
 {
  "cell_type": "code",
  "source": [
    "'/home/wsuser/work/models/DigitRecog_IBM_model.tar.gz'"
  ],
  "metadata": {
    "id": "xctL0364aLxv"
  },
  "execution_count": null,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
    "ls"
  ],
  "metadata": {
    "id": "m5v-cUpzaPAH"
  },
  "execution_count": null,
  "outputs": []
 },
 {
  "cell_type": "markdown",
```

```
  "source": [
    "DigitRecog_IBM_model.tar.gz                mnistCNN.h5\n",
    "handwritten-digit-recognition-model_new.tgz"
  ],
  "metadata": {
    "id": "CdkKGqqYaU1J"
  }
 },
 {
  "cell_type": "markdown",
  "source": [
    "TEST MODEL"
  ],
  "metadata": {
    "id": "RK-LBmAEaZWS"
  }
 },
 {
  "cell_type": "code",
  "source": [
    "from tensorflow.keras.models import load_model\n",
    "from keras.preprocessing import image\n",
    "from PIL import Image\n",
    "import numpy as np"
  ],
  "metadata": {
    "id": "2xlcufFaaaag"
  },
  "execution_count": 3,
  "outputs": []
 },
```

```
{
 "cell_type": "code",
 "source": [
  "model = load_model(\"mnistCNN.h5\")"
 ],
 "metadata": {
  "id": "ZdklCqf-aeSu"
 },
 "execution_count": null,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
  "import os, types\n",
  "import pandas as pd\n",
  "from botocore.client import Config\n",
  "import ibm_boto3\n",
  "\n",
  "def __iter__(self): return 0\n",
  "\n",
  "# @hidden_cell\n",
  "# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.\n",
  "# You might want to remove those credentials before you share the notebook.\n",
  "cos_client = ibm_boto3.client(service_name='s3',\n",
  "    ibm_api_key_id='is_QZGPyU8oxZr3W-td-LCHXS3QPMaWArILi18FdSyGT',\n",
  "    ibm_auth_endpoint=\"https://iam.cloud.ibm.com/oidc/token\",\n",
  "    config=Config(signature_version='oauth'),\n",
  "    endpoint_url='https://s3.private.ap.cloud-object-storage.appdomain.cloud')\n",
  "\n",
```

```
    "bucket = 'handwrittenimagerecognition-donotdelete-pr-8tlrnykut46vpi'\n",
    "object_key = 'mnist-dataset-1024x424 (2).png'\n",
    "\n",
    "streaming_body_1 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']\n",
    "\n",
    "# Your data file was loaded into a botocore.response.StreamingBody object.\n",
    "# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.\n",
    "# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/\n",
    "# pandas documentation: http://pandas.pydata.org/"
   ],
   "metadata": {
    "id": "IDM4wX3zalMv"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "img = Image.open(streaming_body_1).convert(\"L\") # convert image to monochrome\n",
    "img = img.resize( (28,28) ) # resizing of input image"
   ],
   "metadata": {
    "id": "VgXWnCl1apiZ"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "code",
```

```
   "source": [
    "img"
   ],
   "metadata": {
    "id": "BgnROzPNaqq6"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "markdown",
   "source": [

"![download.png](
BIAAABUUlEQVR4nGNgGGSAEcFklrW+ftfCjeHH2vP/ISIsCEmhovB7V+zUGBi4H77FMCTlzf//v3+f2v3+T
RYbupz6tv9///4/bao28ftNT0ZUOcbKH////X9fw8AgPvdTBROqJP+8/9/+fmgVZmBgSHh5VhJFjsn18Zcp
ByYJMDAwMMQ8/5SCYq7swv837J3EGRgYGBgSXv6eyYLsFTV3hkPXX0HY9x4IQOyE2cwnzvDiPZR9aP4
nVMn/DC9u/IPZcforK4rkP4ZHV/7CJFl+HvuHJMmqwPD2C9x1f67uRJYUcPl/7jFMjtP91DNkY38+YxThgk
m6Rv/4jexNxuj/b6pEWVhZWQXcZlzerge1G+rYd1+Ei7SeMDH8k7URO9VymQEFiBWe+f3/37//f748Xm
0MCzt4GLKppVmz/Gf8uufg1Zf/0SUZGJgYGRgYGP79Z6A1AABiMn1gl3jf8wAAAABJRU5ErkJggg==)"

   ],
   "metadata": {
    "id": "BNcPegZLa-qU"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "im2arr = np.array(img) #converting to image\n",
    "im2arr = im2arr.reshape(1, 28, 28, 1) #reshaping according to our requirement"
   ],
   "metadata": {
    "id": "V46Cig7cbSkF"
```

```
  },
  "execution_count": null,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "pred = model.predict(im2arr)\n",
   "print(pred)"
  ],
  "metadata": {
   "id": "K6H1lSQ0bTa2"
  },
  "execution_count": null,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "[[1.0000000e+00 5.3912803e-17 3.9648812e-11 2.0051219e-16 5.1053910e-18\n",
   "  2.9315760e-12 7.0849349e-13 2.0999634e-16 2.9204243e-09 7.4729778e-11]]"
  ],
  "metadata": {
   "id": "I1aaLRfcbXOx"
  },
  "execution_count": null,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
```

```
      "print(np.argmax(pred, axis=1)) #printing our Labels"
    ],
    "metadata": {
      "id": "U15fDU9rbeav"
    },
    "execution_count": null,
    "outputs": []
  },
  {
    "cell_type": "markdown",
    "source": [
      "[0]"
    ],
    "metadata": {
      "id": "ECcT3gupbhxn"
    }
  }
],
"metadata": {
  "colab": {
    "provenance": []
  },
  "kernelspec": {
    "display_name": "Python 3",
    "name": "python3"
  },
  "language_info": {
    "name": "python"
  }
},
"nbformat": 4,
```

```
  "nbformat_minor": 0
}
```