

```
!unzip Input_Data.zip -d ./content
```

➞ File "["<ipython-input-10-84d4835b431d>"](#), line 3
 `pip install --upgrade scikit-learn`

SyntaxError: invalid syntax

```
!pip install matplotlib chainer scikit-learn scipy numpy
```

➞ Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (3.10.0)
Requirement already satisfied: chainer in /usr/local/lib/python3.11/dist-packages (7.8.1)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (1.6.1)
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-packages (1.13.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (1.23.5)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (3.2.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages (from chainer) (75.1.0)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.11/dist-packages (from chainer) (4.12.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from chainer) (3.17.0)
Requirement already satisfied: protobuf>=3.0.0 in /usr/local/lib/python3.11/dist-packages (from chainer) (4.25.6)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.11/dist-packages (from chainer) (1.17.0)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (3.5.0)

```
import matplotlib
import sklearn
import scipy
import numpy

print("All libraries installed successfully!")
```

⇒ All libraries installed successfully!

```
#InputData.py
```

```
import binascii

# Input Images
def get_images(filename, bol=False, length=10000):
    # Parameters -
    # 1. filename - FORMAT: filepath/filename
    # 2. bol - (default -False)-- get images for full length or not
    # 3. length of input images (default=10000)
    length = length*784
    with open(filename,'rb') as f:
        byte_=f.read()
        i = 16
        data = []
        while True:
            byte = byte_[i:i+1]
            if len(byte) == 0:
                break
            if i == length+16 and bol==False:
                break
            val = int.from_bytes(byte,byteorder='big', signed=False)
            data.append(val/255)
            i=i+1
    return data
```

```

# Input Lables
def get_labels(filename):
    # Parameters -
    # 1. filename - FORMAT: filepath/filename
    with open(filename,'rb') as f:
        byte_=f.read()
        i = 8
        data = []
        while True:
            byte = byte_[i:i+1]
            if len(byte) == 0:
                break
            hexadecimal = binascii.hexlify(byte)
            decimal = int(hexadecimal, 16)
            data.append(decimal)
            i = i+1
    return data

```

✓ RBF kernel

#Training——set (RBF)

```

from input_data import get_labels, get_images
from sklearn import svm
import pickle
import numpy as np

train_data = get_images('Input_Data/train-images-idx3-ubyte/train-images.idx3-ubyte', length=50000)
train_labels = get_labels('Input_Data/train-labels-idx1-ubyte/train-labels.idx1-ubyte')

clf = svm.SVC(kernel='rbf')
train_data = np.asarray(train_data[: (50000*784)]).reshape(50000, 784)

```

```
clf.fit(train_data, train_labels[:50000])
```

```
# save the model to disk
filename = 'finalized_model_50000_f_rbf.sav'
pickle.dump(clf, open(filename, 'wb'))
print("Succeed!")
```

⇒ Succeed!

```
#test_model.py
```

```
from input_data import get_labels, get_images
import pickle
import numpy as np
```

```
filename = 'finalized_model_50000_f_rbf.sav'
```

```
# load the model from disk
clf = pickle.load(open(filename, 'rb'))
```

```
test_data=get_images('Input_Data/t10k-images-idx3-ubyte/t10k-images.idx3-ubyte', True) # True: for full length
test_labels=get_labels('Input_Data/t10k-labels-idx1-ubyte/t10k-labels.idx1-ubyte')
```

```
test_data = np.array(test_data).reshape(10000, 784)
result = clf.score(test_data, test_labels)
print("Accuracy: ", result)
```

⇒ Accuracy: 0.9785

✓ Linear Kernel

```
#Training——set (RBF)

from input_data import get_labels, get_images
from sklearn import svm
import pickle
import numpy as np

train_data = get_images('Input_Data/train-images-idx3-ubyte/train-images.idx3-ubyte',
train_labels = get_labels('Input_Data/train-labels-idx1-ubyte/train-labels.idx1-ubyte')

clf = svm.SVC(kernel='linear')
train_data = np.asarray(train_data[: (50000*784)]).reshape(50000, 784)

clf.fit(train_data, train_labels[:50000])

# save the model to disk
filename = 'finalized_model_50000_f_linear.sav'
pickle.dump(clf, open(filename, 'wb'))
print("Succeed!")
```

⇒ Succeed!

```
#test_model.py

from input_data import get_labels, get_images
import pickle
import numpy as np

filename = 'finalized_model_50000_f_linear.sav'
```

```
# load the model from disk
clf = pickle.load(open(filename, 'rb'))

test_data=get_images('Input_Data/t10k-images-idx3-ubyte/t10k-images.idx3-ubyte',True)
test_labels=get_labels('Input_Data/t10k-labels-idx1-ubyte/t10k-labels.idx1-ubyte')

test_data = np.array(test_data).reshape(10000, 784)
result = clf.score(test_data, test_labels)
print("Accuracy: ",result)
```

 Accuracy: 0.9388