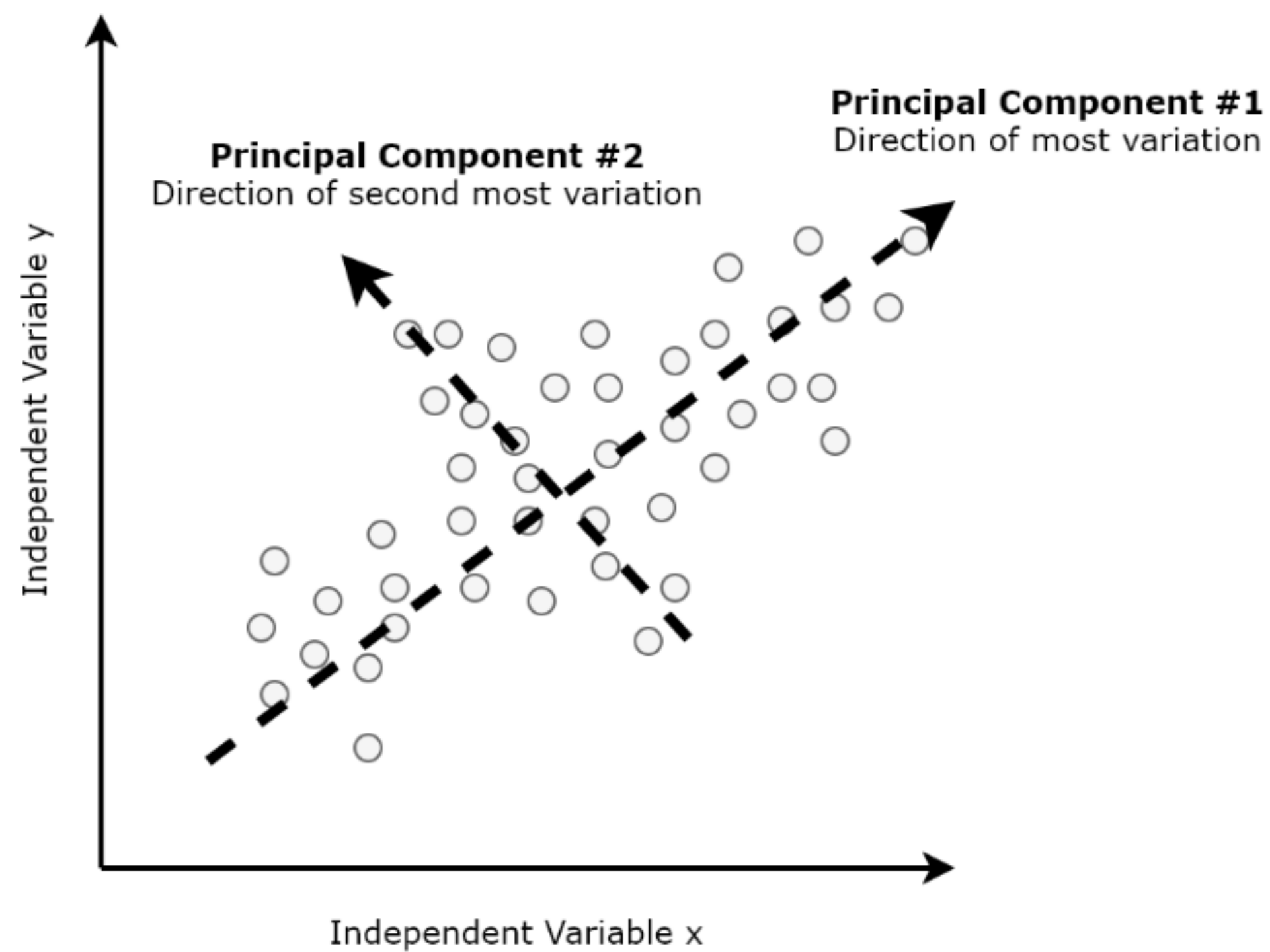


Practice 7

Principal Component Analysis

Problem

- **PCA** : Use RowMatrix in mllib
 - Use predefined function in **pyspark.mllib.linalg.distributed.RowMatrix**



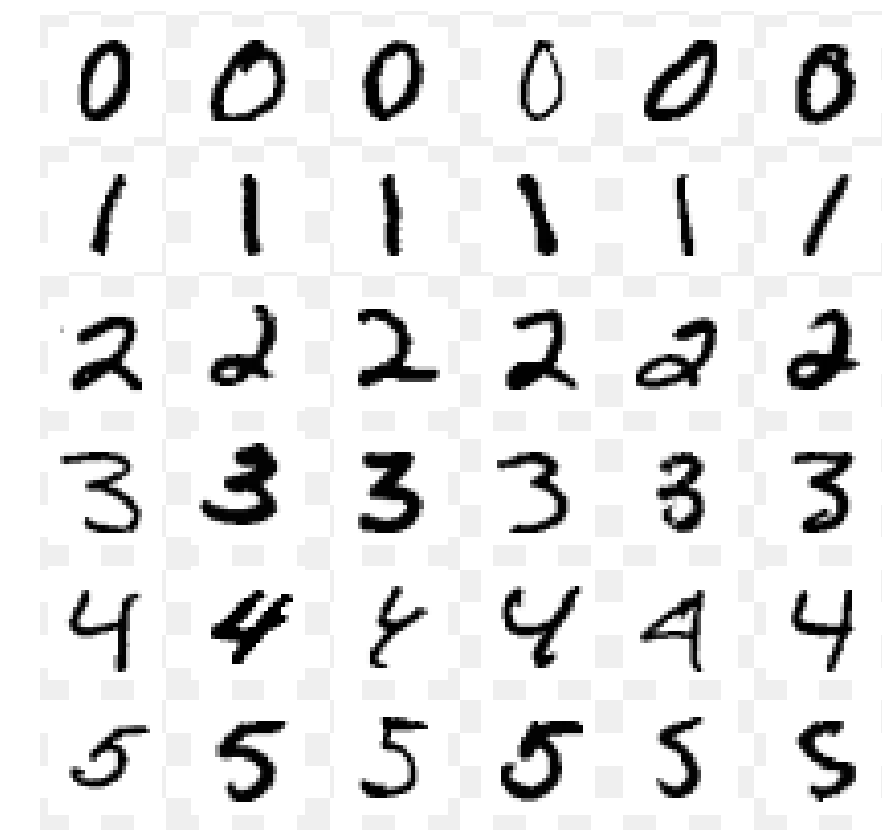
Dataset for PCA

- **MNIST** : Recognition of handwritten digits
 - There are 10 handwritten digits(0~9) in bitmap format.

- 784 Features (pixel values)

1. Pixel 1
2. Pixel 2
...
...
...
63. Pixel 63
784. Pixel 784

* The MNIST Database :



- You can download the dataset using ***sklearn.datasets.fetch_openml*** library

Practice 7

1. Use predefined classes in *pyspark.mllib.linalg.distributed.RowMatrix* for PCA

- Row matrix makes the RDD data be row-oriented distributed matrix
- It has many sub-functions, and you need to use *computePrincipalComponents* to get principal component of Row matrix.

- *For example,*

```
pc_rdd = mat.computePrincipalComponents(16)
```

- In this example, mat means transformed Row Matrix of MNIST dataset

2. Reduce the number of the dataset features from **784** to **16**

Practice 7

3. Visualize principal components after implementing PCA on MNIST dataset.

- You need to visualize the principal component of MNIST dataset in **28x28** bitmap.
- Print out **first 16 pictures in 2x8** matrix.
- For example,

```
image_shape = (28,28)
fig,axes = plt.subplots(2, 8, figsize=(15,12),subplot_kw = {'xticks': (), 'yticks': ()})
for i, (component, ax) in enumerate(zip(pct, axes.ravel())):
    ax.imshow(component.reshape(image_shape), cmap='gray_r')
```

Practice 7

4. You need to use predefined arguments we suggest.

- Number of data points: 10,000

Use first ten thousands(10,000) data points as datasets

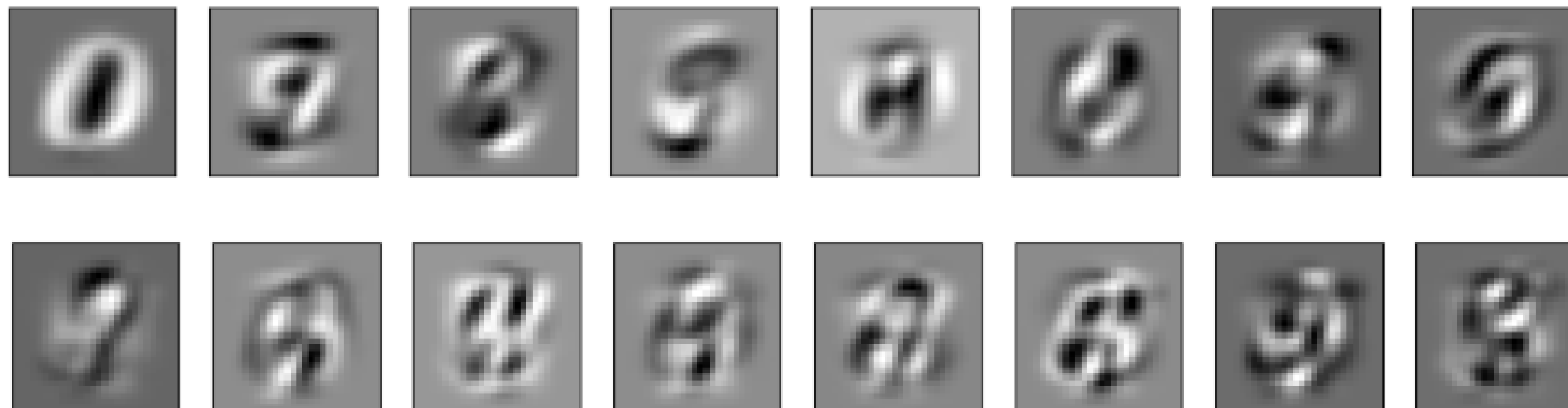
- Number of partitions: 300

You can split data when you make it RDDs.

For example, “ *RDD = sc.parallelize(Data, numPartition)* ”

Submission

1. You have to submit “**result.png**” file on iCampus.
2. In your **result.png** file, there must be figures of principal components of MNIST dataset.
3. Deadline: *May 28th 23:59 P.M.*
4. Your **result.png** file must be like following



Solution

➤ load MNIST dataset and libraries for PCA

```
import numpy as np
import matplotlib.pyplot as plt
from pyspark import SparkConf, SparkContext
from pyspark.sql import SQLContext
from pyspark.mllib.linalg.distributed import RowMatrix
from sklearn.datasets import fetch_openml
```

Library for visualizing principal components

Row matrix makes the RDD data be row-oriented distributed matrix

Load MNIST dataset

➤ Set Spark configuration

```
conf = SparkConf()
conf.set("spark.master", "local")
sc = SparkContext(conf=conf)
sqlContext = SQLContext(sc)
```

Configure master as local

Solution

➤ Preprocess MNIST dataset

```
mnist = fetch_openml('mnist_784')
```

Use 10,000 data points to get principal components of MNIST dataset

```
data = mnist.data[:10000]
```

```
rdd = sc.parallelize(data.tolist(), 300)
```

Configure numpartition to 300

```
rdd.cache()
```

```
mat = RowMatrix(rdd)
```

➤ Do PCA

Find 16 main features among 784 features

```
pc_rdd = mat.computePrincipalComponents(16)
```

```
pc = pc_rdd.toArray()
```

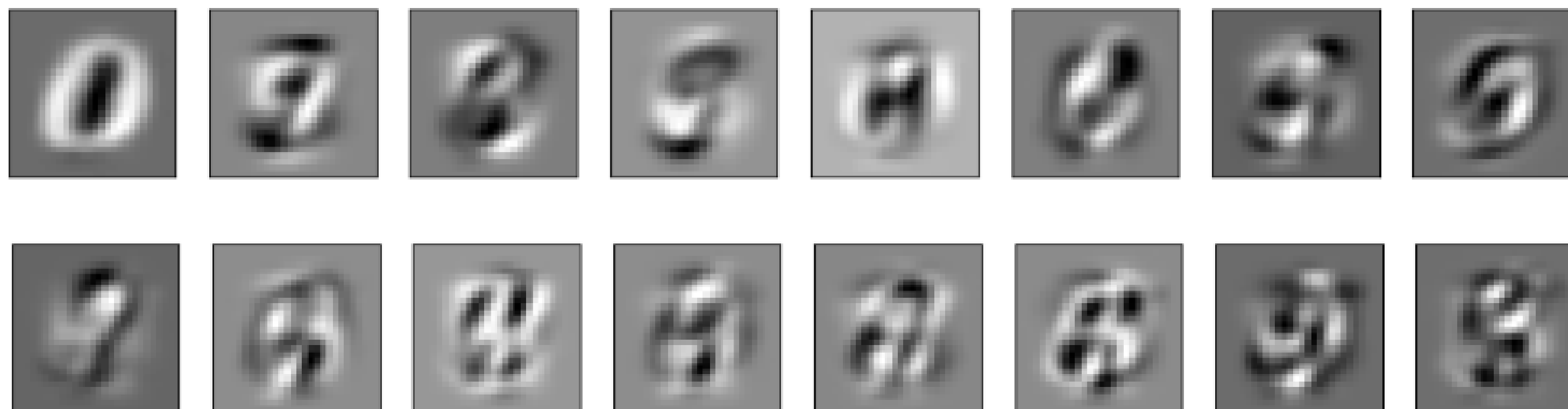
```
pct = np.transpose(pc)
```

Solution

➤ Visualize principal components

```
image_shape = (28,28)
fig,axes = plt.subplots(2, 8, figsize=(15,12),subplot_kw = {'xticks': (), 'yticks': ()})
for i, (component, ax) in enumerate(zip(pct, axes.ravel())):
    ax.imshow(component.reshape(image_shape), cmap='gray_r')
plt.savefig('result.png')
sc.stop()
```

➤ Result



Key features of handwritten digits image