Clustering and SVM using scikit

Part 1: Image Classification using clustering

Scope of this project:

The scope of this problem statement pertains to the clustering and SVM using scikit-learn"

Objectives of this project

- Learn to process images using scikit-learn
- Use clustering to perform image classification by grouping the images and identifying the image label
- Use SVM from scikit-learn

Problem Statement:

K-means clustering and Dimensionality reduction (also known as Principal Component Analysis) are powerful techniques in the category of unsupervised learning. They can be applied broadly and agnostic to the data type. For instance, one can use clustering to process speech vectors. They can also be used to perform text or image classification. In this problem, we perform clustering on Fashion MNIST dataset.

The broad approach to building this is as below:

- Obtain a sample of the Fashion MNIST dataset using the web service
- Create a representation for each image
- Use clustering to segment the list of images
- Inspect the clusters, identify the labels these clusters seem to represent
- Test this with a new image

Detailed Steps:

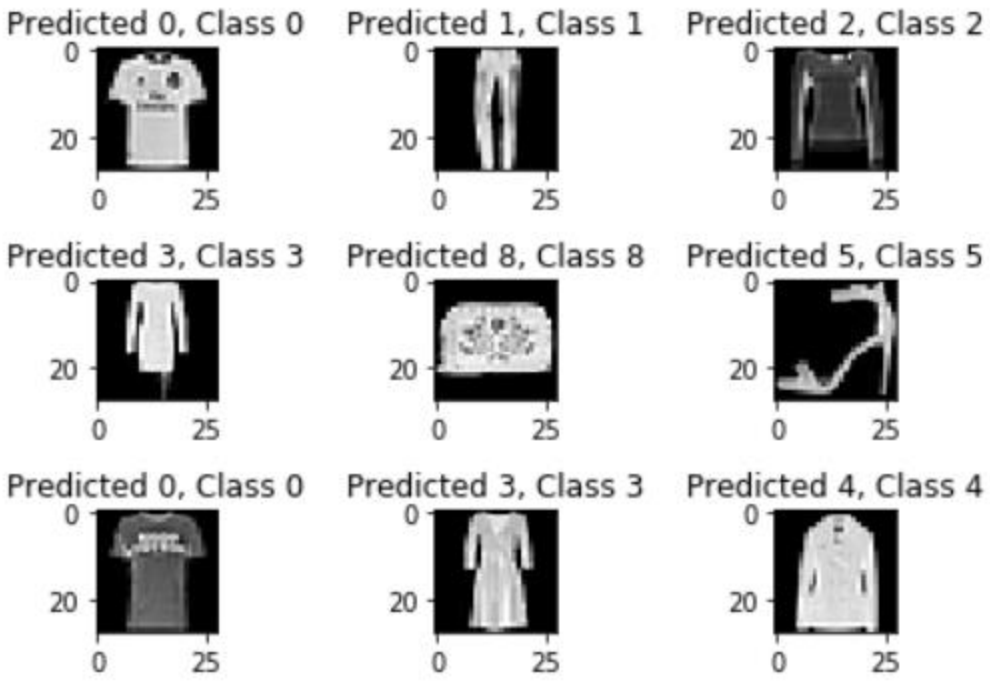
1. Obtain a sample of 60000 records of dataset. Please refer the starter code example_fashion_mnist.py that is provided to you.
2. Each training example of the dataset is a tuple (X, Y), where X is an image and Y is the corresponding label. The image is a Python list having 28 rows and 28 columns. You should flatten this in to a single vector. You can use numpy to reshape this.
3. For 60000 images, you will get as many vectors and the shape of your dataset will be: (60000, 784). Your label values will be (60000, 1), where each label is an integer. The mapping of the label integer to its corresponding category is shown in the screenshot below:

```
# Create dictionary of target classes
label_dict = {
 0: "T-shirt/top",
 1: "Trouser",
 2: "Pullover",
 3: "Dress",
 4: "Coat",
 5: "Sandal",
 6: "Shirt",
 7: "Sneaker",
 8: "Bag",
 9: "Ankle boot"
}
```

4. Implement a K-means clustering model that maps an input in to one of the 10 clusters.
5. Visualize the input images and the corresponding labels. You can refer to the starter code for visualization with matplotlib
6. Test the above by obtaining another 25 samples from web service, plotting them along with your predicted cluster number.

Exercise Part 2:

1. By inspection map the each cluster number to the label it represents. When the clustering algorithm assigns cluster number for each image, it need not assign number 0 for the images that are labelled with 0 and so on. Hence, assuming that our algorithm did a good job, we can manually inspect each cluster and map it to the actual label.
2. As we do have the target labels available as a part of the dataset, we now can compute the accuracy. Use about 5000 test cases obtained from web service and get the cluster prediction. Compare this with the target label. Report the accuracy
3. Implement a Linear SVM based classifier in scikit, train it with 60000 records that you fetched before and perform the classification for the same 5000 tests. Compare the accuracies.
4. An example visualization is shown in the screenshot below:



Deliverables

- Code that implements clustering for Fashion MNIST images, SVM implementation
- Report that discusses your results, approach and screenshots of images with cluster assignments