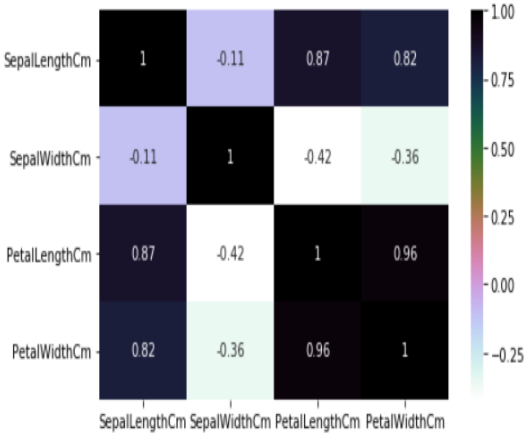
Unsupervised Classification methods from scratch

Ayush Chaurasia
IIT DELHI

1. Iris Dataset

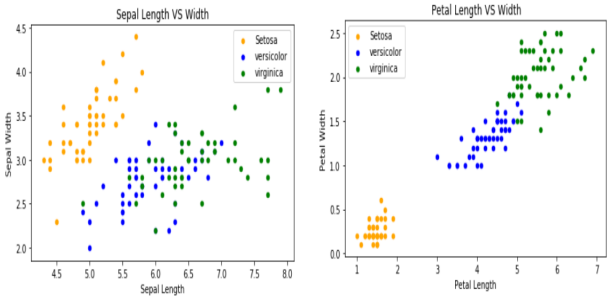
Objective is to predict the species of flower based on its characteristics. Data set contains 150 examples. Features in the data set:

- Sepal Length(in cm)
- Sepal Width(in cm)
- Petal Length(in cm)
- Petal Width(in cm)
- Species:Iris-Setosa,Iris-versicolor,Iris-virginica



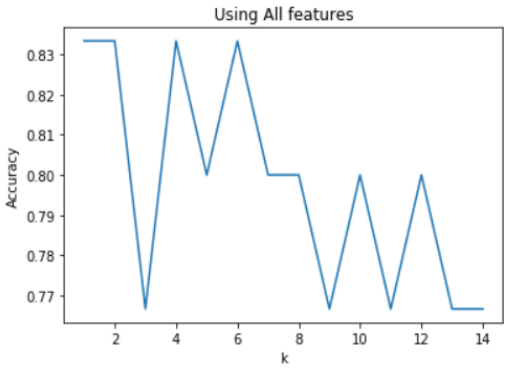
From the above correlation table it is observed that the Sepal Width and Length are not correlated whereas the Petal Width and Length are highly correlated.

1.1. Data Analysis



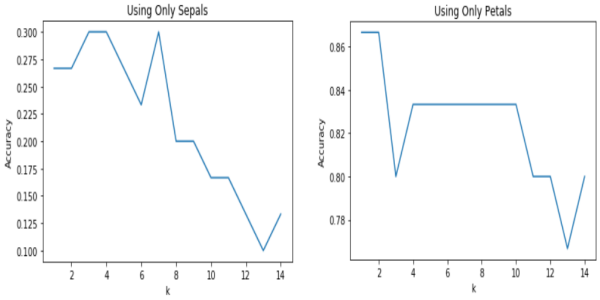
In the above figures it can be seen that sepal length vs width for each specie is not that clearly clustered compared petal length vs width for each specie. But since both the figures(petal and sepal) are still showing clustering indicates that we can apply KNN or K-means algorithm to them to predict the specie. Also, petal clustering is more good(distant) than sepal clustering, one might expect to have better results in clustering algorithms with only petal features taken into account rather than sepal features(as seen further).

1.2. K-Nearest Neighbour



Data is split in 80:20(train:test). In the above figure we used all the features and applied KNN algorithm. Highest accuracy have been obtained for 4

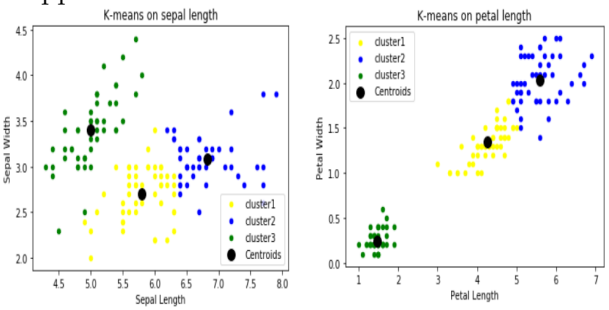
and 6 clusters with accuracy of 0.83 over test data.



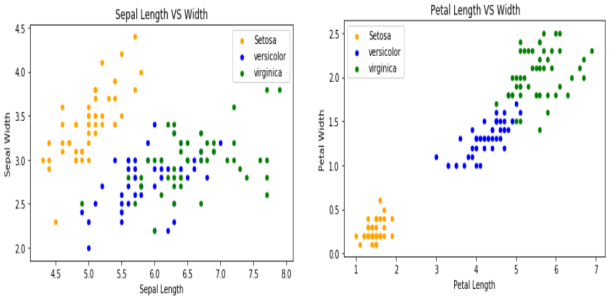
In the above figure on left only sepal features were used for KNN i.e. petal width and length were not considered. The highest accuracy achieved is only 0.3 (for 3 clusters). This is also expected as discussed in the data analysis section that clustering of sepal width vs length was not good and also they weren't highly correlated. In the above figure on the right only petal features are used for KNN and a high accuracy is obtained for 2(accuracy = 0.86) or 3 clusters(accuracy = 0.83)(higher accuracy than using all features). This is expected also for the similar reasons discussed in data analysis part.

1.3. K-Means

Since it is a unsupervised we use all the data points i.e. no train and test split. We separately apply K-means for 2 cases: first in which petal features are dropped and second in which sepal features are dropped.



In both the above figures K-means has quite well separated the clusters with each cluster given a particular color and black dots showing the centroid. These figures are quite good replica of the actual species figures shown below which shows that K-means works well and is doing good classification.



1.4. GMM

We implemented with the following variants:

- 2 stopping criteria work together. First is the no of iterations and second is the change in log-likelihood less than some threshold.
- 2 types of initialization for the clusters are used: random and K-means cluster used from sci-kit learn.

In general 2 results were observed by using different combination of parameters which was mainly due to different method of initialization(though number of gaussians were always used 3):

- When initialized with random clusters, one of the gaussian always diminished(coefficient was negligible) and henceforth 2 of the flower species were classified under the same cluster.
- When initialized with K-means, the results were very good with each cluster holding majorly one of the species.

```
{'Iris-virginica': 0, 'Iris-setosa': 47, 'Iris-versicolor': 0}
{'Iris-virginica': 0, 'Iris-setosa': 3, 'Iris-versicolor': 2}
{'Iris-virginica': 50, 'Iris-setosa': 0, 'Iris-versicolor': 48}
```

Above is the result of GMM for random cluster initialization where one cluster goes empty and one containing 2 species.

```
{'Iris-virginica': 50, 'Iris-setosa': 0, 'Iris-versicolor': 5}
{'Iris-virginica': 0, 'Iris-setosa': 50, 'Iris-versicolor': 0}
{'Iris-virginica': 0, 'Iris-setosa': 0, 'Iris-versicolor': 45}
```

Above is the result of GMM for K-means cluster initialization where very good results are obtained.

1.5. SVM

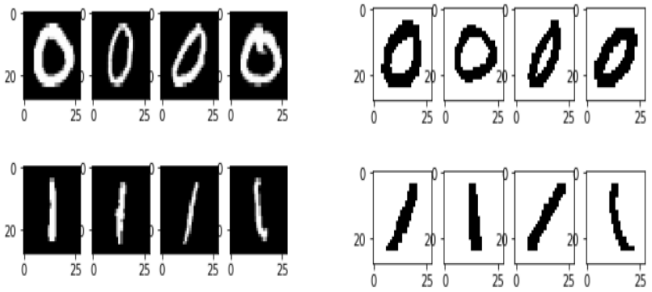
We used the sci-kit learn SVM which gave an accuracy of 86.6 %. Since SVM is primarily a binary classifier here it is used in a one-one fashion as was given in the sci-kit learn implementation.

2. Digit Recognizer

Goal is to correctly identify digits from a dataset of thousands of handwritten grayscale images of 28X28 pixel size. Since it was a very well known dataset we haven't given the dataset feature description.

2.1. Grayscale and Binary images

We used the dataset in 2 different ways one is the original image given in grayscale format with each pixel value from 0-255. The other is a binary image used with each pixel value either 0 or 1. It is made from grayscale images by changing all non-zero pixel values to 1.



The image above on the left is a grayscale image and the corresponding binary image is on the right.

2.2. SVM

SVM here was used in one-one fashion as in the implementation of sci-kit learn. Dataset was divided in 80 % train and 20 % test data. We used SVM on the following 3 data modifications:

2.2.1. Grayscale images

Accuracy was very less of about 11 % and it took about 35 minutes to train the model on the dataset.

2.2.2. Binary images

Accuracy rose to about 94 % and it took just about 5 minutes to train the model on the dataset.

2.3. PCA on grayscale images

Top 200 dimensions were taken after applying PCA. The accuracy was again a meagre 12 % but the train time(12 minutes) was better than the original dataset train time(35 minutes).

Key observations from the above are that grayscale images aren't giving good accuracy with SVM and henceforth PCA will also not give a good accuracy since it is just a dimensionalty reduction technique but the main advantage is in the training time which is a 3^{rd} of the original dataset training time with almost same accuracy. Also, binary images gave much better result indicating the fact that SVM makes a strict classification and henceforth gives good classification with images containing pixel values of only 0 or 1 rather than 0-255 values.

2.4. GMM

On applying GMM we either didn't get good results or did not get results at all. Using the complete dataset caused overflow problems or memory errors. On using a part of data gave inifinity or NAN in log-likelihood and hence we couldn't apply GMM to this problem.

2.5. KNN

KNN was taking a lot of time to train on the complete training dataset and hence we decided to train on some partial dataset. So we only took first 1000 samples to train on and it gave good accuracy results of about 82 % when using 10 clusters.

3. Problems

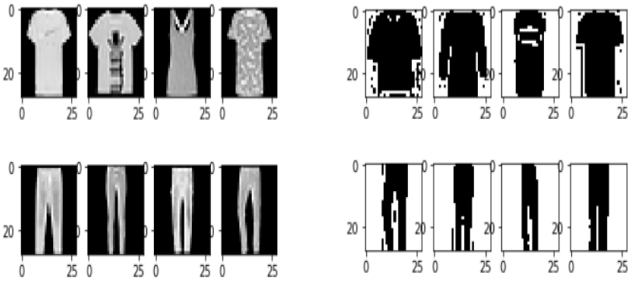
We faced a few problems mainly in image datasets. We found most of the image datasets in jpg format and were too high in complexity and size to handle. Even the digit recognizer problem gave memory exception in the GMM case and was taking a lot of time train in KNN. Since we weren't satisfied ourselves with the results in digit recognizer we tried another similar problem of fashion mnist dataset.

4. Fashion MNIST

Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. Goal is to identify the image with correct label. The labels are the following: 0:T-shirt/top, 1:Trouser, 2:Pullover, 3:Dress, 4:Coat, 5:Sandal, 6:Shirt, 7:Sneaker, 8:Bag, 9:Ankle boot.

4.1. Grayscale and Binary images

We used the dataset in 2 different ways one is the original image given in grayscale format with each pixel value from 0-255. The other is a binary image used with each pixel value either 0 or 1. It is made from grayscale images by changing all non-zero pixel values to 1.



The image above on the left is a grayscale image and the corresponding binary image is on the right.
Due to limitation of resources and large amount of time required to train model. We used only 5000 samples to train the SVM models.

4.2. SVM

SVM here was used in one-one fashion as in the implementation of sci-kit learn. Dataset was divided in 80 % train and 20 % test data. Then we took only 5000 of the train samples as the actual training data. We used SVM on the following 3 data modifications:

4.2.1. Grayscale images

Accuracy was very less of about 10 % and it took about 47 seconds to train the model on the dataset.

4.2.2. Binary images

Accuracy rose to about 80 % and it took just about 8 seconds to train the model on the dataset.

4.3. PCA on grayscale images

Top 200 dimensions were taken after applying PCA. The accuracy was again a meagre 10 % but the train time(14 seconds) was better than the original dataset train time(47 seconds).

Key observations from the above are that grayscale images aren't giving good accuracy with SVM and henceforth PCA will also not give a good accuracy since it is just a dimensionalty reduction technique but the main advantage is in the training time which is less than a 3^{rd} of the original dataset training time with almost same accuracy. Also, binary images gave much better result indicating the fact that SVM makes a strict classification and henceforth gives good classification with images containing pixel values of only 0 or 1 rather than 0-255 values.

4.4. GMM

On applying GMM we either didn't get good results or did not get results at all. Using the complete dataset caused overflow problems or memory errors. On using a part of data gave inifinity or NAN in log-likelihood and hence we couldn't apply GMM to this problem.
The same problem was observed with the digit recognizer and hence is a work to be explored in future.

4.5. KNN

KNN was taking a lot of time to train on the complete training dataset(even on 5000 samples) and hence we decided to train on some partial dataset. So we only took first 1000 samples to train on and the accurcary was about 70 %. Also a observation was that the error mainly were seen in the product number 2 and 4 which were pullover and coat respectively.

5. References

- https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95
- https://www.kaggle.com/c/digit-recognizer/data
- https://www.kaggle.com/sabasiddiqi/svm-classification-parameter-selection-0-968
- https://www.geeksforgeeks.org/multiclass-classification-using-scikit-learn/
- https://www.kaggle.com/s00100624/digit-image-clustering-via-autoencoder-kmeans
- https://www.kaggle.com/zalando-research/fashionmnist