Question 1 : (30 total points) Image data analysis with PCA

In this question we employ PCA to analyse image data

1.1 (3 points) Once you have applied the normalisation from Step 1 to Step 4 above, report the values of the first 4 elements for the first training sample in `Xtrn_nm`, i.e. `Xtrn_nm[0,:]` and the last training sample, i.e. `Xtrn_nm[-1,:]`.

The first 4 elements for the training sample in Xtrn_nm are:

$$[-3.13725490e-06, -2.26797386e-05, -1.17973856e-04, -4.07058824e-04]$$

The last 4 elements for the training sample in Xtrn_nm are:

$$[-3.13725490e-06, -2.26797386e-05, -1.17973856e-04, -4.07058824e-04]$$

1.2 (4 points) Using `Xtrn` and Euclidean distance measure, for each class, find the two closest samples and two furthest samples of that class to the mean vector of the class.

From the figure below, the sample images from each class in the second column and third column have much more similar shape with mean vector image, but their color are different from mean vector image. The samples from each classes in the fourth column and fifth column have very different shape comparing to mean vector image, but their color are closer to mean vector image than the two closest samples in each class.

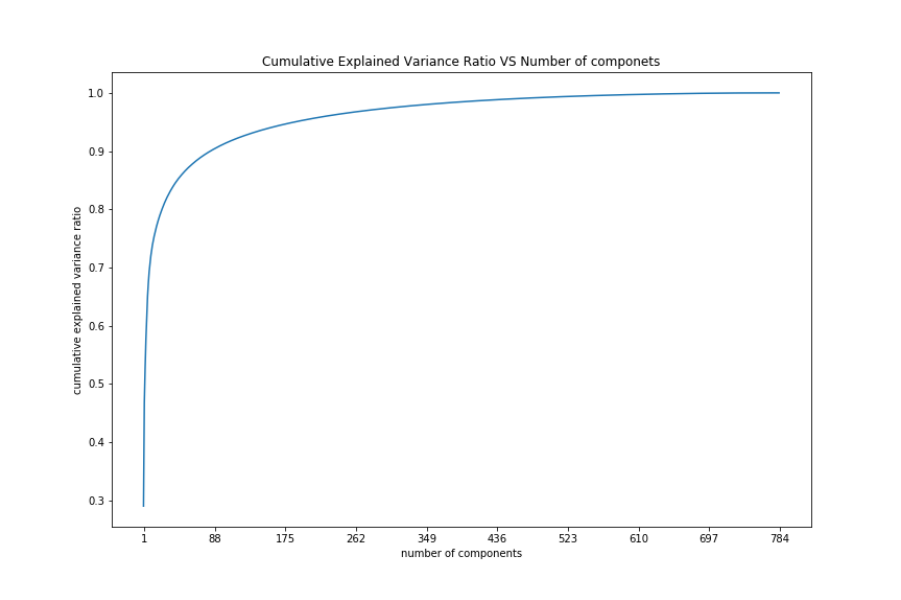
1.3 (3 points) Apply Principal Component Analysis (PCA) to the data of `Xtrn_nm` using sklearn.decomposition.PCA, and report the variances of projected data for the first five principal components in a table. Note that you should use `Xtrn_nm` instead of `Xtrn`.

Results are presented in the table below

1st PC	2nd PC	3rd PC	4th PC	5th PC
19.810	12.112	4.106	3.382	2.625

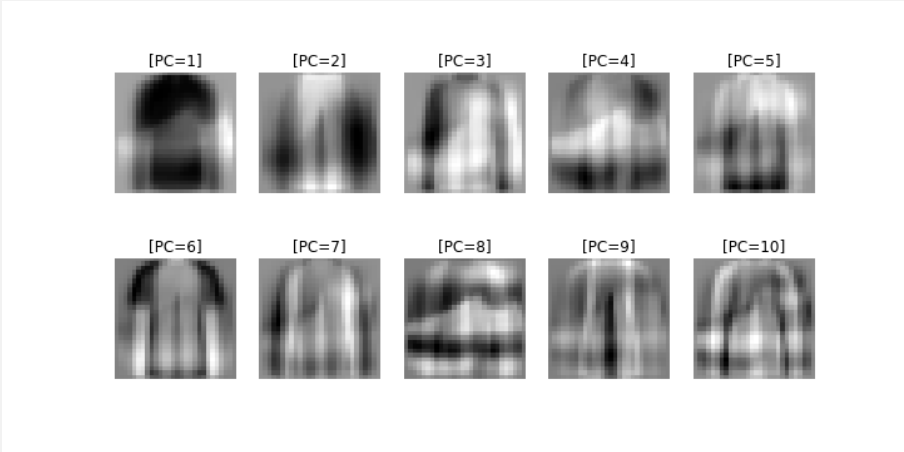
1.4 (3 points) Plot a graph of the cumulative explained variance ratio as a function of the number of principal components, K, where $1 \leq K \leq 784$. Discuss the result briefly.

From the figure below, the cumulative explained variance ratio increases very quickly in range of K=1 to K=88. Then cumulative explained variance ratio increases very slow. After K=380, Then cumulative explained variance ratio becomes constant. Which means first 88 components can explain the most variance. When number of principal components K is round 380, all the variance can be explained.



1.5 (4 points) Display the images of the first 10 principal components in a 2-by-5 grid, putting the image of 1st principal component on the top left corner, followed by the one of 2nd component to the right. Discuss your findings briefly.

From the figure below, each component applied on image very different. some images are with shadow of shoes, some are with shadow of pants. Different component can be explained by different algorithms of PCA applied. Each component as matrix to describe the original data(pixels). each pixel values are projected differently onto the new dimensions of the data for each respective matrix.



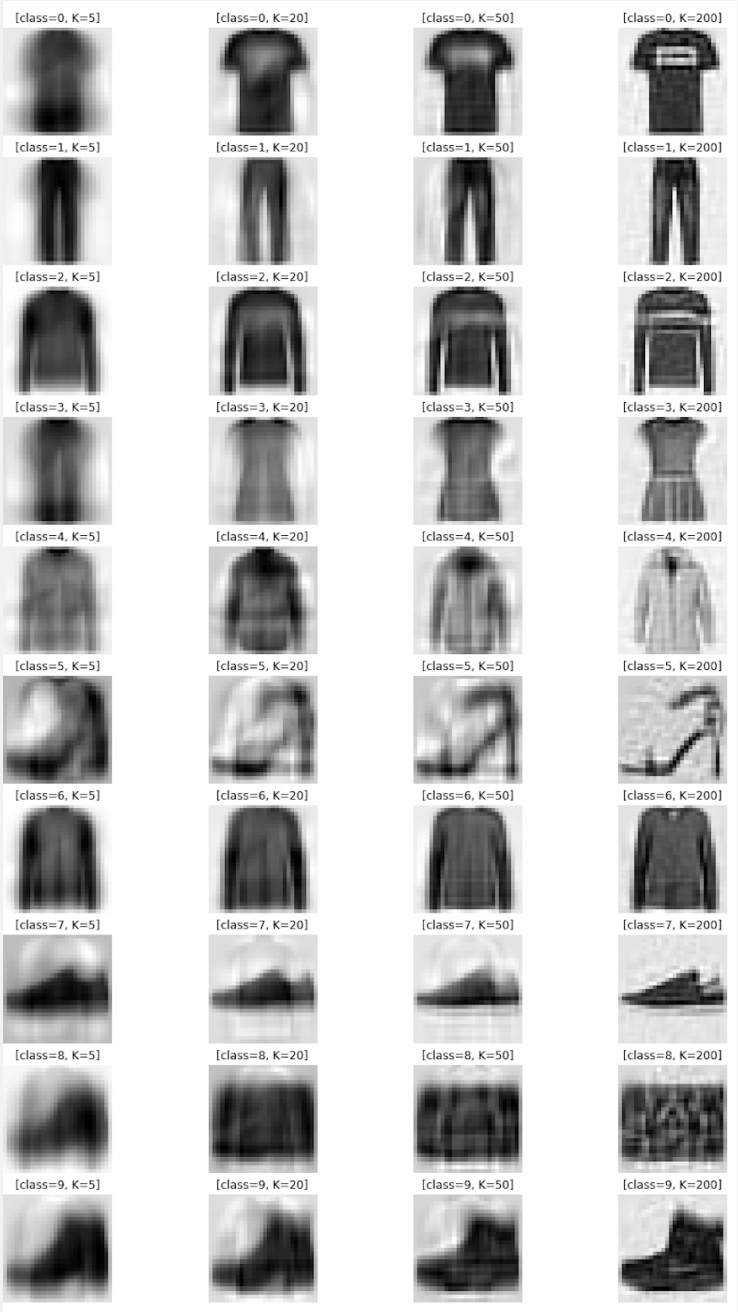
1.6 (5 points) Using `Xtrn_nm`, for each class and for each number of principal components $K = 5, 20, 50, 200$, apply dimensionality reduction with PCA to the first sample in the class, reconstruct the sample from the dimensionality-reduced sample, and report the Root Mean Square Error (RMSE) between the original sample in `Xtrn_nm` and reconstructed one.

Results are presented in the table below

	K=5	K=20	K=50	K=200
class_0	0.256	0.150	0.127	0.061
class_1	0.198	0.140	0.095	0.038
class_2	0.199	0.146	0.124	0.083
class_3	0.146	0.107	0.084	0.056
class_4	0.118	0.103	0.088	0.046
class_5	0.181	0.159	0.142	0.091
class_6	0.129	0.096	0.072	0.046
class_7	0.166	0.128	0.107	0.062
class_8	0.223	0.145	0.124	0.093
class_9	0.184	0.151	0.122	0.071

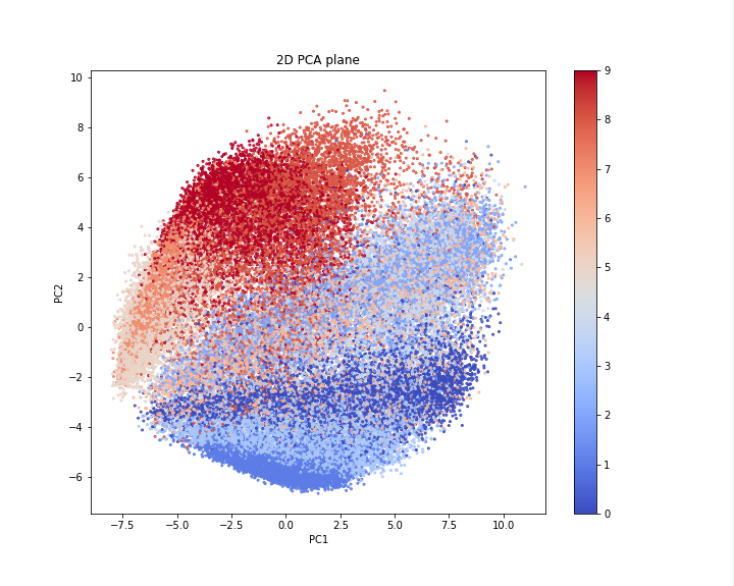
1.7 (4 points) Display the image for each of the reconstructed samples in a 10-by-4 grid, where each row corresponds to a class and each row column corresponds to a value of $K = 5,\ 20,\ 50,\ 200$.

From the figure below, in each class, when number of principal components increase, the more representative of the original image the reconstruction becomes. There is a sequential improvement of image when more principal components are used. The quality change of image for each class is very big from given number of principal components=5 to number of principal components=20. The quality change becomes smaller when the more principal components are used.



1.8 (4 points) Plot all the training samples (`Xtrn_nm`) on the two-dimensional PCA plane you obtained in Question 1.3, where each sample is represented as a small point with a colour specific to the class of the sample. Use the 'coolwarm' colormap for plotting.

From the figure below, in general, the separation is very bad for each class, although many warm color points occupy at upper left side and many cool color points are in bottom right side. There are some points are overlapping with another color points. There is no clear line can be regarded to separate warm and cool color points.



Question 2 : (25 total points) Logistic regression and SVM

In this question we will explore classification of image data with logistic regression and support vector machines (SVM) and visualisation of decision regions.

2.1 (3 points) Carry out a classification experiment with multinomial logistic regression, and report the classification accuracy and confusion matrix (in numbers rather than in graphical representation such as heatmap) for the test set.

The classification accuracy is 0.8401 and the confusion matrix is:

[[819, 3, 15, 50, 7, 4, 90, 1, 11, 0],
[5, 953, 4, 27, 5, 0, 3, 1, 2, 0],
[27, 4, 731, 11, 133, 0, 82, 2, 9, 1],
[31, 15, 14, 866, 33, 0, 37, 0, 4, 0],
[0, 3, 115, 38, 760, 2, 72, 0, 10, 0],
[2, 0, 0, 1, 0, 911, 0, 56, 10, 20],
[147, 3, 128, 46, 108, 0, 539, 0, 28, 1],
[0, 0, 0, 0, 0, 32, 0, 936, 1, 31],
[7, 1, 6, 11, 3, 7, 15, 5, 945, 0],
[0, 0, 0, 1, 0, 15, 1, 42, 0, 941]]

Question 2 continued on next page...

2.2 (3 points) Carry out a classification experiment with SVM classifiers, and report the mean accuracy and confusion matrix (in numbers) for the test set.

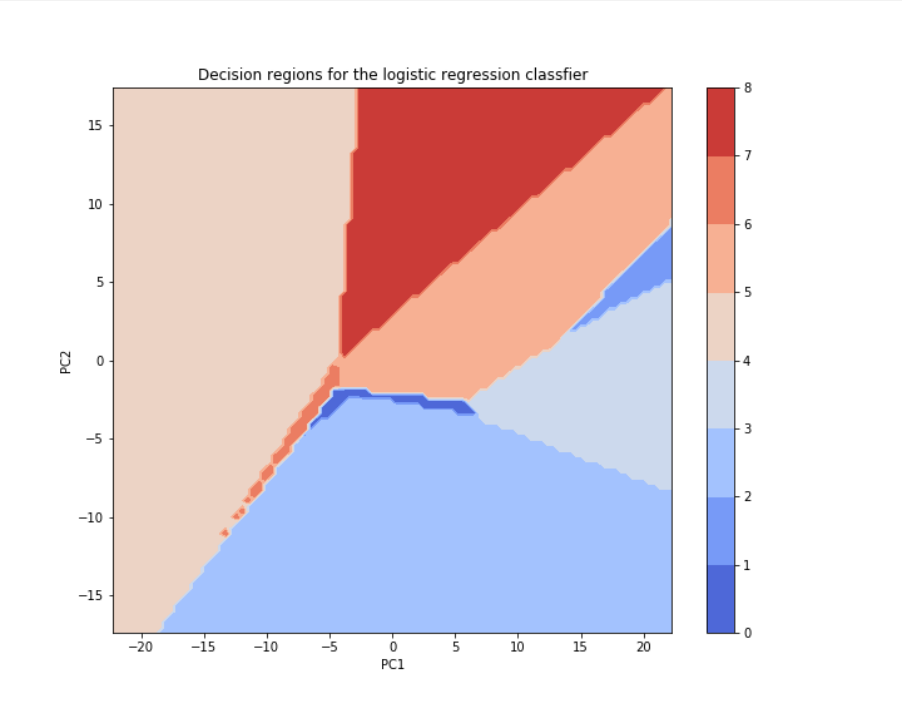
The mean accuracy is 0.8461 and the confusion matrix is:

[[845, 2, 8, 51, 4, 4, 72, 0, 14, 0],
[4, 951, 7, 31, 5, 0, 1, 0, 1, 0],
[15, 2, 748, 11, 137, 0, 79, 0, 8, 0],
[32, 6, 12, 881, 26, 0, 40, 0, 3, 0],
[1, 0, 98, 36, 775, 0, 86, 0, 4, 0],
[0, 0, 0, 1, 0, 914, 0, 57, 2, 26],
[185, 1, 122, 39, 95, 0, 533, 0, 25, 0],
[0, 0, 0, 0, 0, 34, 0, 925, 0, 41],
[3, 1, 8, 5, 2, 4, 13, 4, 959, 1],
[0, 0, 0, 0, 0, 22, 0, 47, 1, 930]]

Question 2 continued on next page...

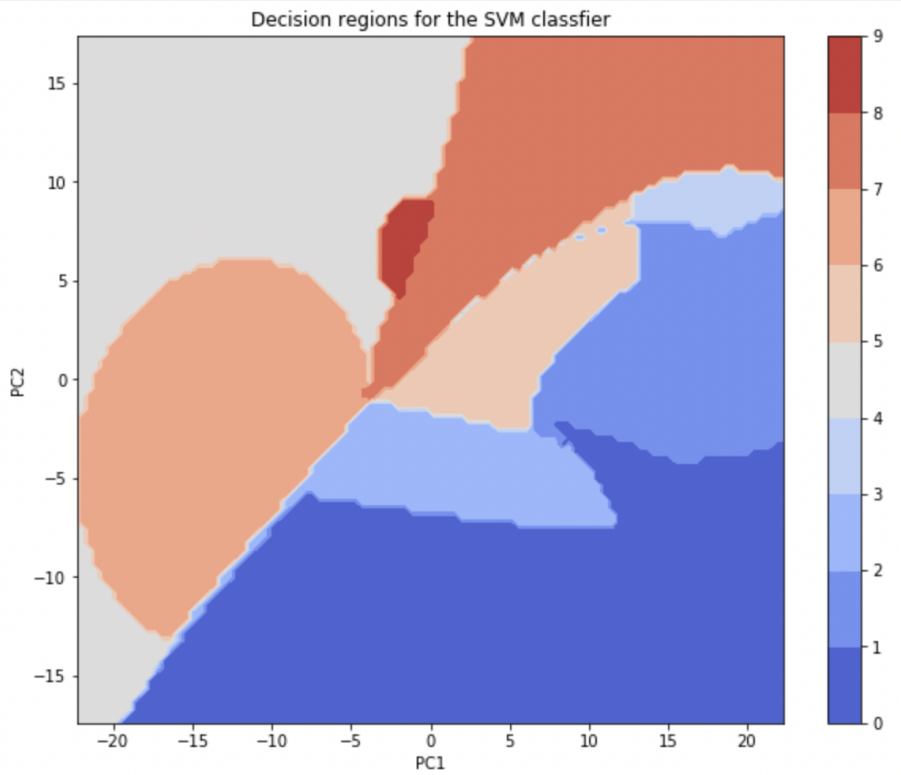
2.3 (6 points) We now want to visualise the decision regions for the logistic regression classifier we trained in Question 2.1.

From the figure below, some regions are small and some regions are very large. The decision boundaries are almost linear since the logistic regression can be considered as a generalised linear model. The prediction results are not including class9, it may be caused by only two principal components are not enough to visualise all classes.



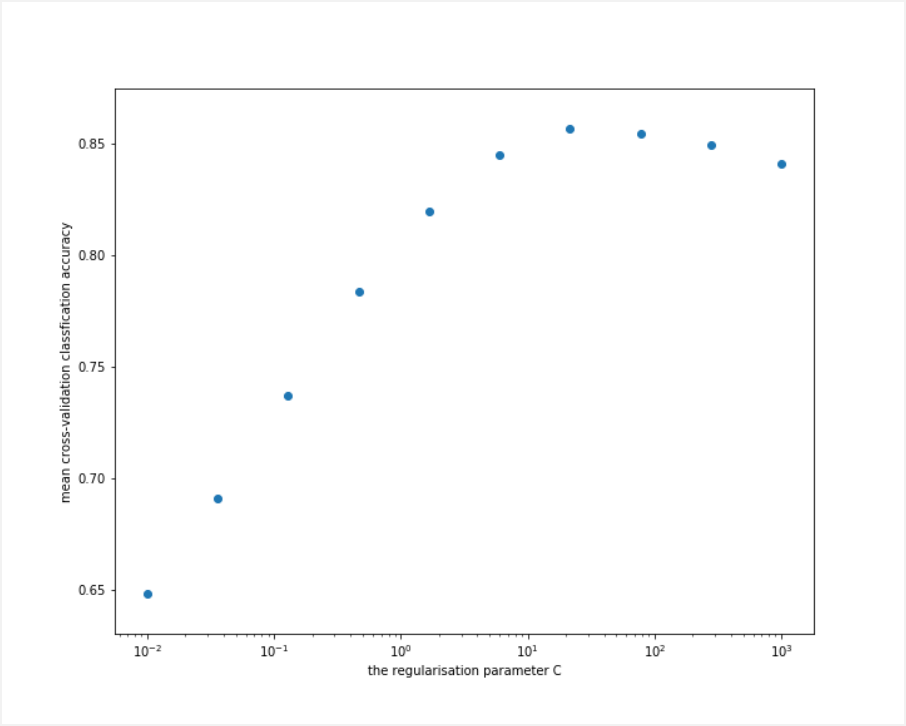
2.4 (4 points) Using the same method as the one above, plot the decision regions for the SVM classifier you trained in Question 2.2. Comparing the result with that you obtained in Question 2.3, discuss your findings briefly.

From the figure below, The decision boundaries is not linear since we use kernel = 'rbf' for SVM to get non-linear decision boundaries. However, the decision boundaries are linear when trained by logistic regression classifier. The prediction results include all classes by only two components in SVM classifier, but only 8 classes can be predicted by using logistic regression classifier.



2.5 (6 points) We used default parameters for the SVM in Question 2.2. We now want to tune the parameters by using cross-validation. To reduce the time for experiments, you pick up the first 1000 training samples from each class to create `Xsmall`, so that `Xsmall` contains 10,000 samples in total. Accordingly, you create labels, `Ysmall`.

The highest mean accuracy is 85.65% with C= 21.544



2.6 (3 points) Train the SVM classifier on the whole training set by using the optimal value of C you found in Question 2.5.

The classification accuracy on the training set 90.84%.
The classification accuracy on the training set 87.65%.

Question 3 : (20 total points) Clustering and Gaussian Mixture Models

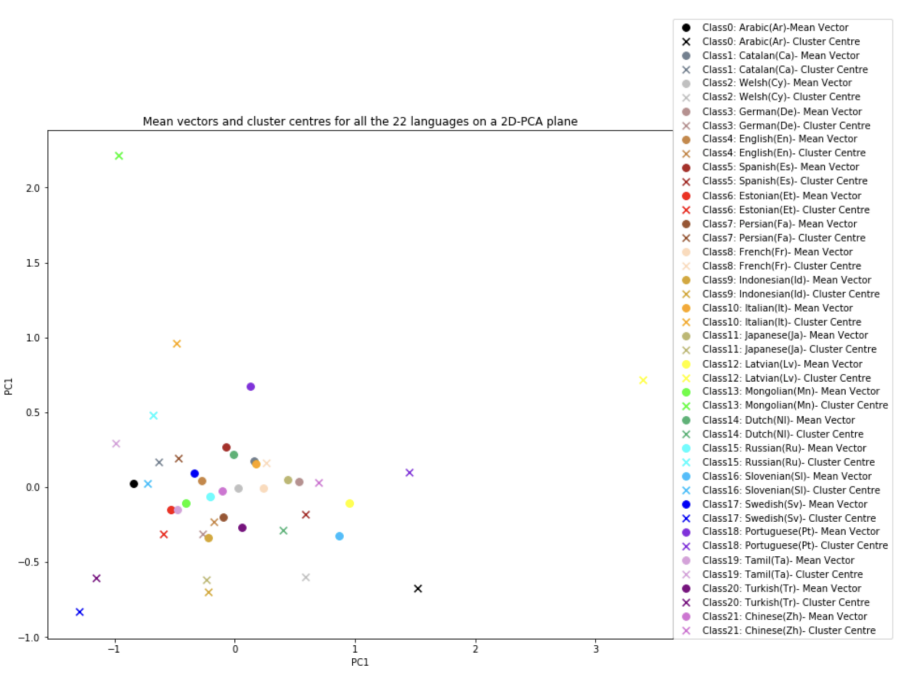
In this question we will explore K-means clustering, hierarchical clustering, and GMMs.

3.1 (3 points) Apply k-means clustering on `Xtrn` for $k = 22$, where we use sklearn.cluster.KMeans with the parameters `n_clusters=22` and `random_state=1`. Report the sum of squared distances of samples to their closest cluster centre, and the number of samples for each cluster.

The sum of squared distances of samples to their closest cluster centre is 38185.816.
The number of samples for each cluster is:
[class$_0$, 1018],
[$class_1$, 1125],
[$class_2$, 1191],
[$class_3$, 890],
[$class_4$, 1162],
[$class_5$, 1332],
[$class_6$, 839],
[$class_7$, 623],
[$class_8$, 1400],
[$class_9$, 838],
[$class_{10}$, 659],
[$class_{11}$, 1276],
[$class_{12}$, 121],
[$class_{13}$, 152],
[$class_{14}$, 950],
[$class_{15}$, 1971],
[$class_{16}$, 1251],
[$class_{17}$, 845],
[$class_{18}$, 896],
[$class_{19}$, 930],
[$class_{20}$, 1065],
[$class_{21}$, 1466]

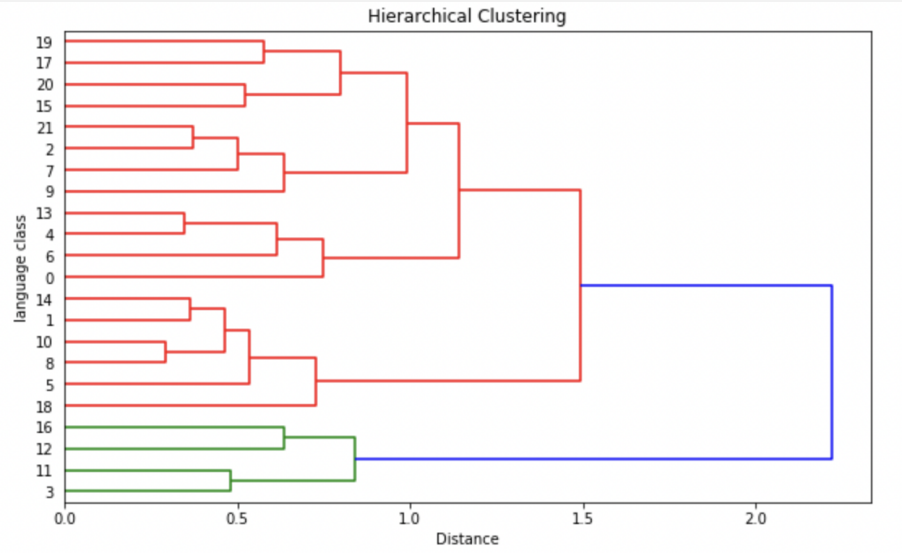
3.2 (3 points) Using the training set only, calculate the mean vector for each language, and plot the mean vectors of all the 22 languages on a 2D-PCA plane, where you apply PCA on the set of 22 mean vectors without applying standardisation. On the same figure, plot the cluster centres obtained in Question 3.1.

mean vectors and cluster centres are not similar. The set of 22 mean vectors is not applying standardisation which means large feature values will have larger influence on mean vectors. And cluster centres depend on how you initialise the cluster centres at first. Given different initialisation will have different cluster centres.



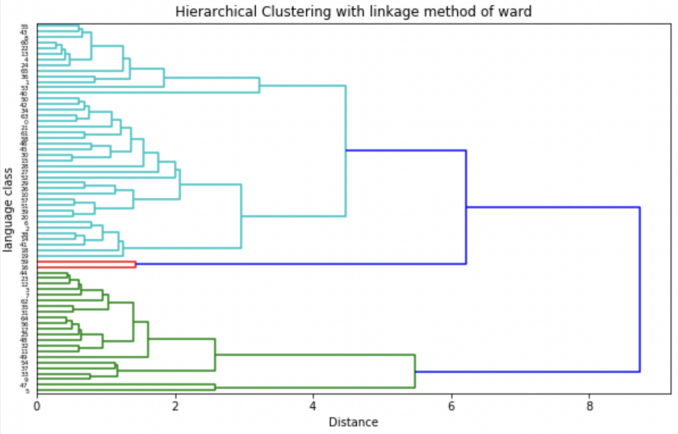
3.3 (3 points) We now apply hierarchical clustering on the training data set to see if there are any structures in the spoken languages.

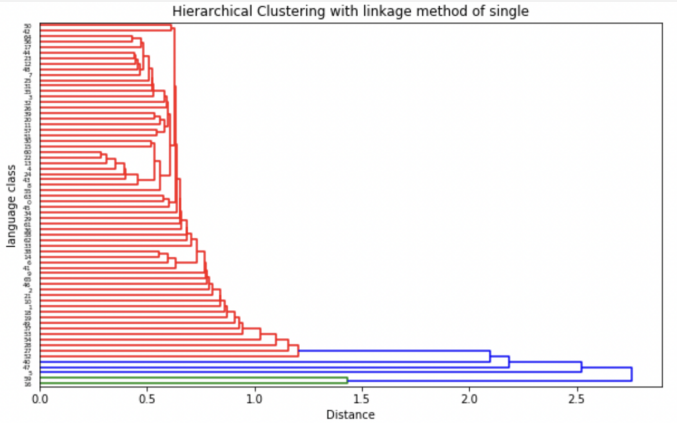
If we only use 22 mean vectors to apply hierarchical clustering, two clusters have automatically created. And the distance for every classes are very small.

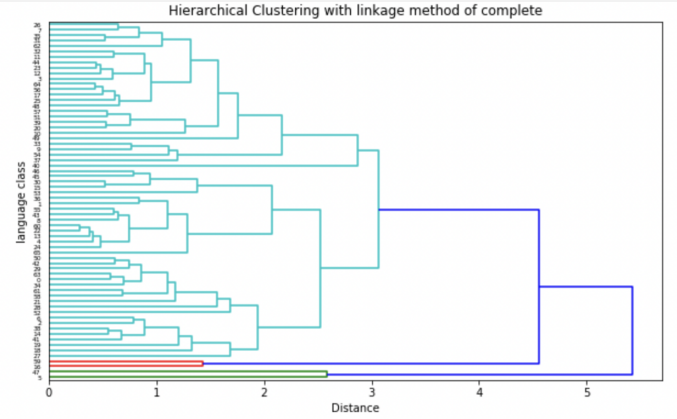


3.4 (5 points) We here extend the hierarchical clustering done in Question 3.3 by using multiple samples from each language.

The hierarchical clustering with linkage method of 'ward' has the largest maximum distance and behave like k-means. The hierarchical clustering with linkage method of 'single' has the smallest maximum distance.







3.5 (6 points) We now consider Gaussian mixture model (GMM), whose probability distribution function (pdf) is given as a linear combination of Gaussian or normal distributions, i.e.,

From the figure below, when applying diagonal covariance matrix, as number of mixture components K increases, the per-sample average log-likelihood on both training and testing data are both increasing and their trend is very similar; moreover, their per-sample average log-likelihood on each K is very similar as well. When applying full covariance matrix, as K increases, the per-sample average log-likelihood increases very quickly on training data; however, the per-sample average log-likelihood increases firstly and then decrease on testing data. It may because of overfitting of single training data.