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 from the first sample are:
`[-3.14e-06 -2.27e-05 -1.18e-04 -4.07e-04]` rounded to 2 decimal places.

The first 4 elements from the last sample are:
`[-3.14e-06 -2.27e-05 -1.18e-04 -4.07e-04]` rounded to 2 decimal places.

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.

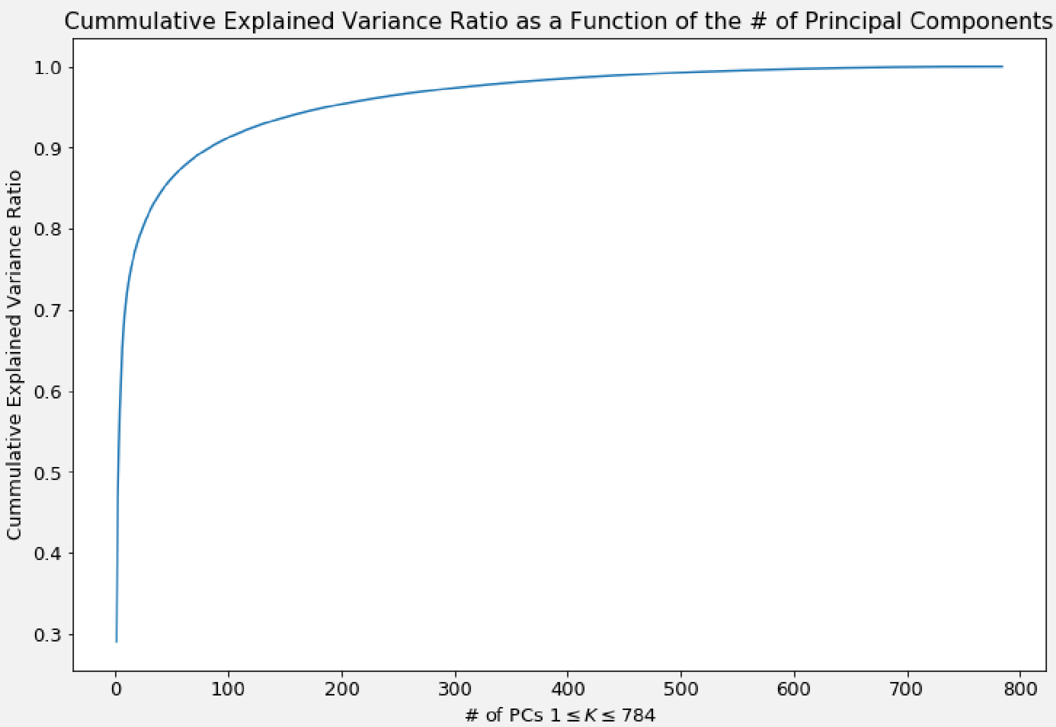
The closest sample to the mean vector holds the highest resemblance of shape, the furthest - although in the same class - has a completely different shape.

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`.

The Explained Variances of the First 5 Principal Components of `Xtrn_nm`

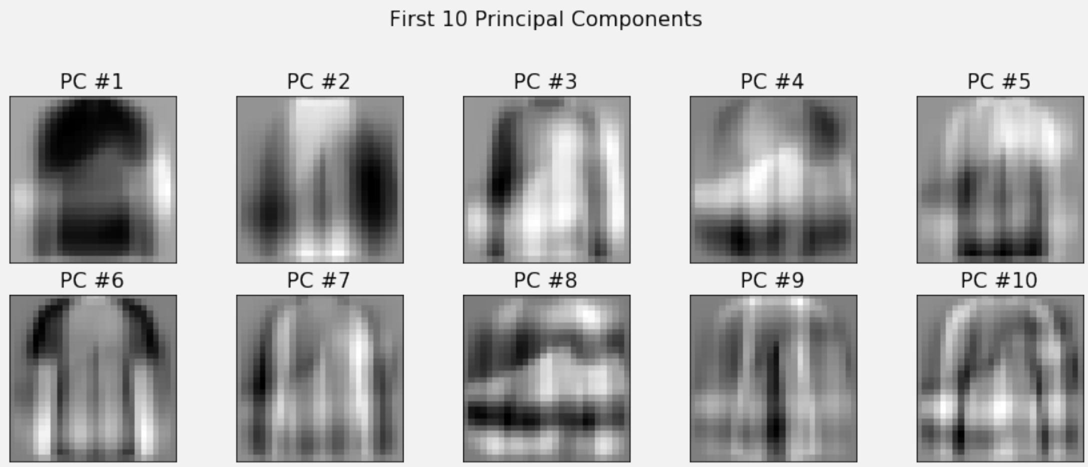
	PC #1	*PC #2*	*PC #3*	*PC #4*	*PC #5*
Variance	~ 19.81	~ 12.11	~ 4.11	~ 3.38	~ 2.62

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.



There is a logarithmic relationship between the cumulative explained variance ratio and the number of principal components, K, as K increases. To reach at least 90% cumulative explained variance, $K = 84$.

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.



White areas are the classes the PC is trying to classify, dark areas are where an image has no correlation with the class. We can see that in PC #1 it is trying to classify a trainer, and the darker areas are a long-sleeved shirt, in PC #2 it's trying to classify trousers, #3 boots, etc. It gets trickier to identify everything as the number of components increases.
We could score an image on how many darker points are produced to decide if it fits the class - the lower the better (i.e. lighter pixels carry a positive weight and darker ones a negative one).

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.

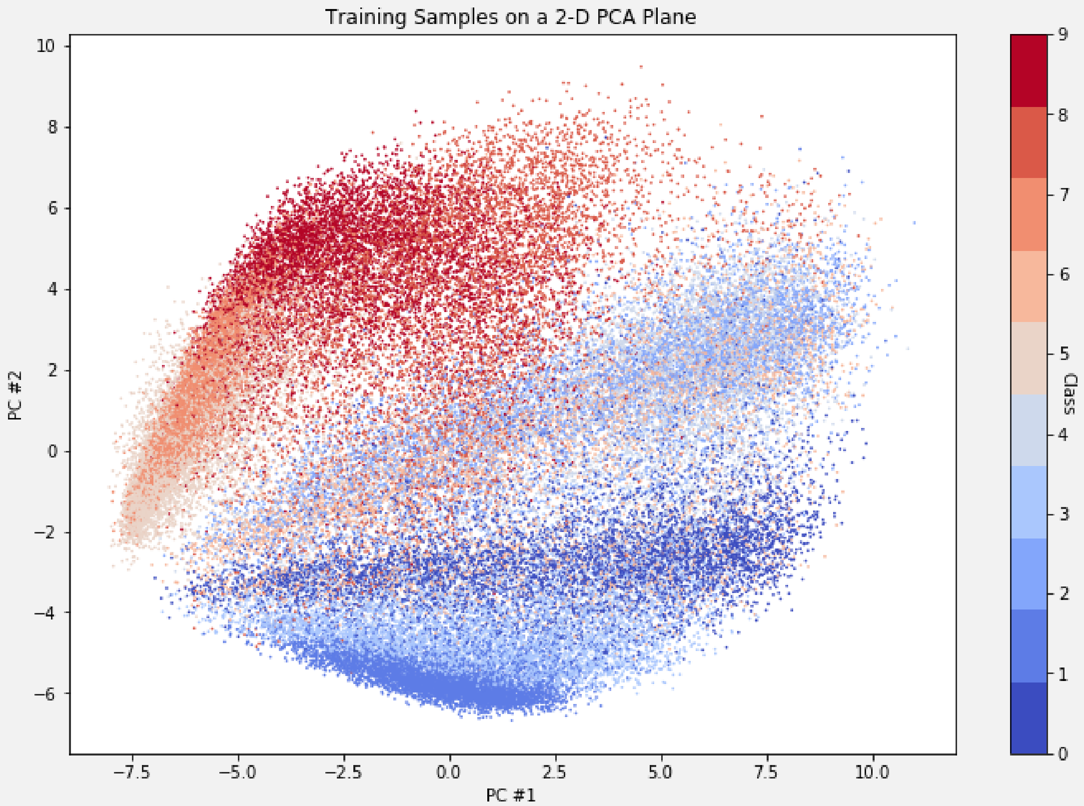
RMSE Between the Original Sample in `Xtrn_nm` and a Reconstructed One from K Number of Principal Components

	RMSE			
	$K = 5$	$K = 20$	$K = 50$	$K = 200$
Class 0	~ 0.256	~ 0.150	~ 0.128	~ 0.060
Class 1	~ 0.198	~ 0.140	~ 0.095	~ 0.345
Class 2	~ 0.199	~ 0.146	~ 0.123	~ 0.078
Class 3	~ 0.146	~ 0.107	~ 0.083	~ 0.056
Class 4	~ 0.118	~ 0.103	~ 0.088	~ 0.046
Class 5	~ 0.181	~ 0.159	~ 0.143	~ 0.090
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	~ 0.092
Class 9	~ 0.184	~ 0.151	~ 0.122	~ 0.073

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 $K = 5,\ 20,\ 50,\ 200$.

It is clear to see that as we increase the number of components that we gather more info, and thus we can produce a much sharper image. We have added the mean vectors to their respected classes for clarity since, because we are not using all components, there will be errors. As we don't have to reconstruct the mean vector we aren't introducing new errors.

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.



PC #1 vs. #2 represents a large variability in the data, so the closer points are to each other the more similar they are. As we can see classes appear to be clustered together, which is what we would intuitively expect to happen. We do however see many outliers to their respected classes.

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 accuracy score: 84.01%

The confusion matrix:

$$\begin{bmatrix} 819 & 3 & 15 & 50 & 7 & 4 & 90 & 1 & 11 & 0 \\ 5 & 953 & 4 & 27 & 5 & 0 & 3 & 1 & 2 & 0 \\ 27 & 4 & 731 & 11 & 133 & 0 & 82 & 2 & 9 & 1 \\ 31 & 15 & 14 & 866 & 33 & 0 & 37 & 0 & 4 & 0 \\ 0 & 3 & 115 & 38 & 760 & 2 & 72 & 0 & 10 & 0 \\ 2 & 0 & 0 & 1 & 0 & 911 & 0 & 56 & 10 & 20 \\ 147 & 3 & 128 & 46 & 108 & 0 & 539 & 0 & 28 & 1 \\ 0 & 0 & 0 & 0 & 0 & 32 & 0 & 936 & 1 & 31 \\ 7 & 1 & 6 & 11 & 3 & 7 & 15 & 5 & 945 & 0 \\ 0 & 0 & 0 & 1 & 0 & 15 & 1 & 42 & 0 & 941 \end{bmatrix}$$

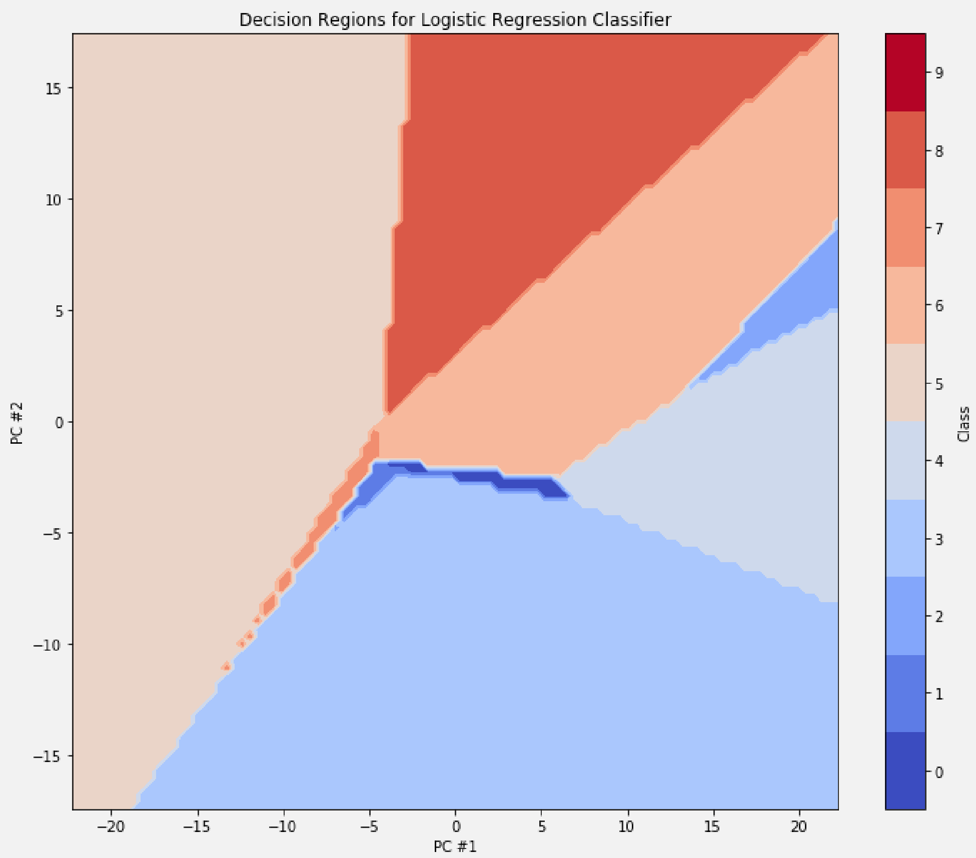
2.2 (3 points) Carry out a classification experiment with SVM classifiers, and report the mean accuracy and confusion matrix (in numbers) for the test set.

The accuracy score: 84.61%

The confusion matrix:

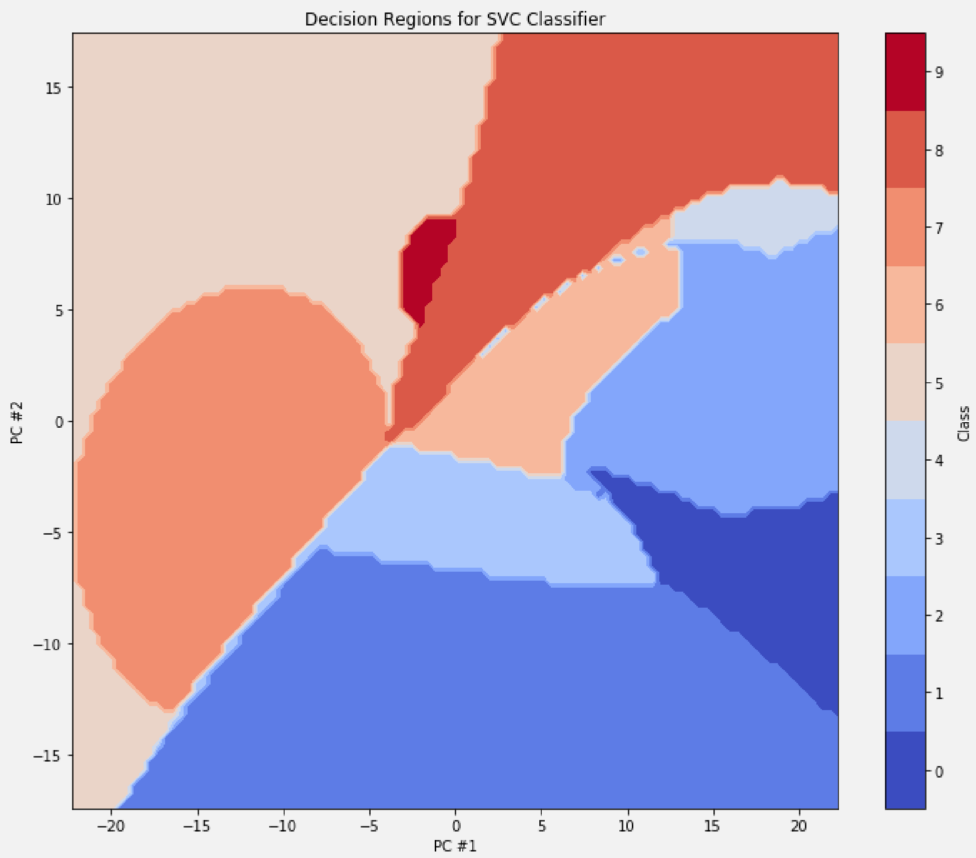
$$\begin{bmatrix}
845 & 2 & 8 & 51 & 4 & 4 & 72 & 0 & 14 & 0 \\
4 & 951 & 7 & 31 & 5 & 0 & 1 & 0 & 1 & 0 \\
15 & 2 & 748 & 11 & 137 & 0 & 79 & 0 & 8 & 0 \\
32 & 6 & 12 & 881 & 26 & 0 & 40 & 0 & 3 & 0 \\
1 & 0 & 98 & 36 & 775 & 0 & 86 & 0 & 4 & 0 \\
0 & 0 & 0 & 1 & 0 & 914 & 0 & 57 & 2 & 26 \\
185 & 1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

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



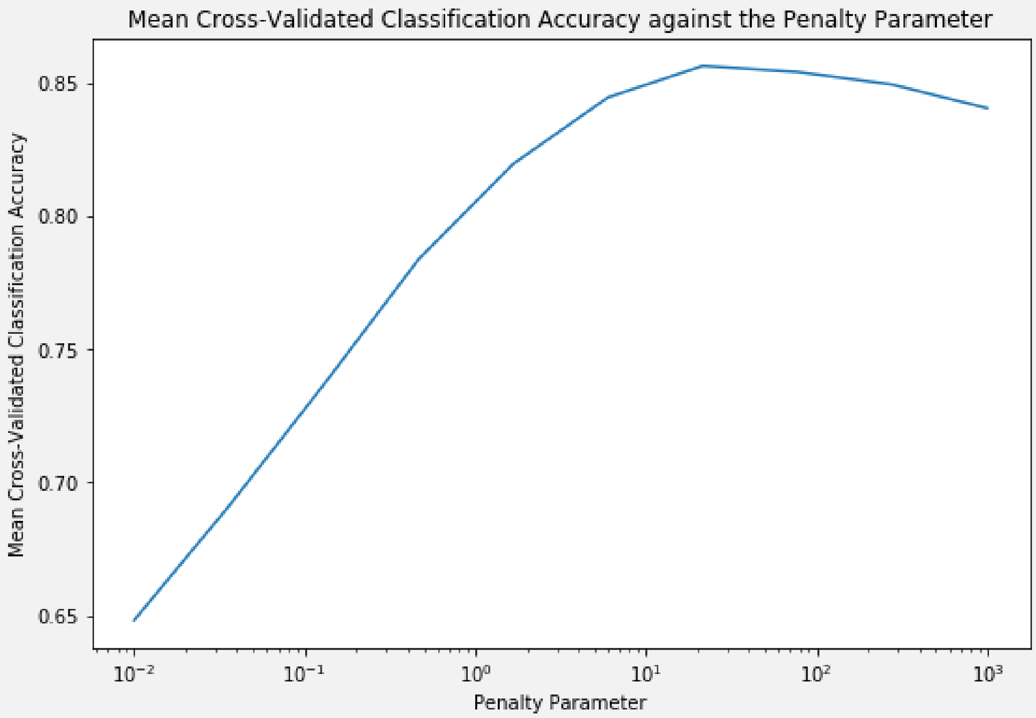
We can see that only classes 0 - 8 have actually been assigned a region. Linear regression predicts a continuous output and finds a linear curve solution, hence every boundary is a straight line. We are missing Class 9 from dimensionality reduction.

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.



Unlike logistic regression, we have drawn a region for every class. SVC supports both linear and non-linear solutions, hence we see many curved decision boundaries. In general, regions for each class have been drawn somewhere else as well.

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`.



Highest score: $\sim 85.65\%$

The value C, the penalty parameter, that obtained it: ~ 21.54 (log-scale)

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.

Classification accuracy on the training set: $\sim 90.84\%$

Classification accuracy on the test 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.

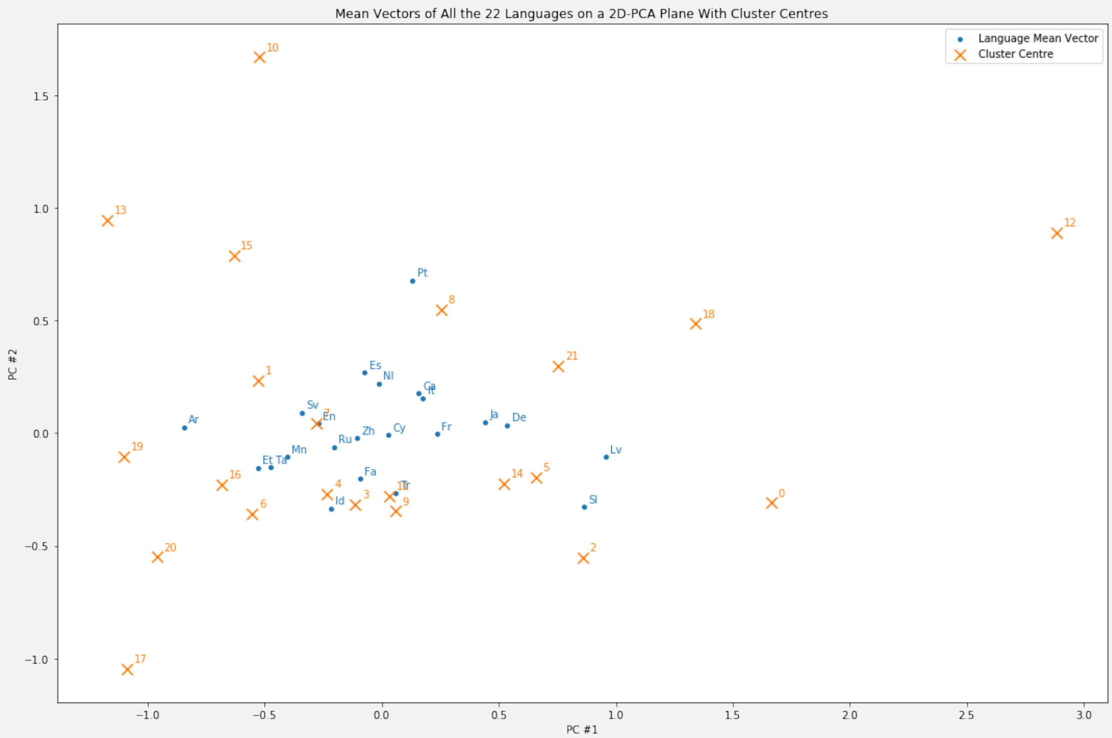
Sum of squared distances of samples to their closest cluster centre: ~ 38185.82

Number of samples for each cluster

Cluster Number	*# of Samples*
0	1018
1	1125
2	1191
3	890
4	1162
5	1332
6	839
7	623
8	1400
9	838
10	659
11	1276
12	121
13	152
14	950
15	1971
16	1251
17	845
18	896
19	930
20	1065
21	1466

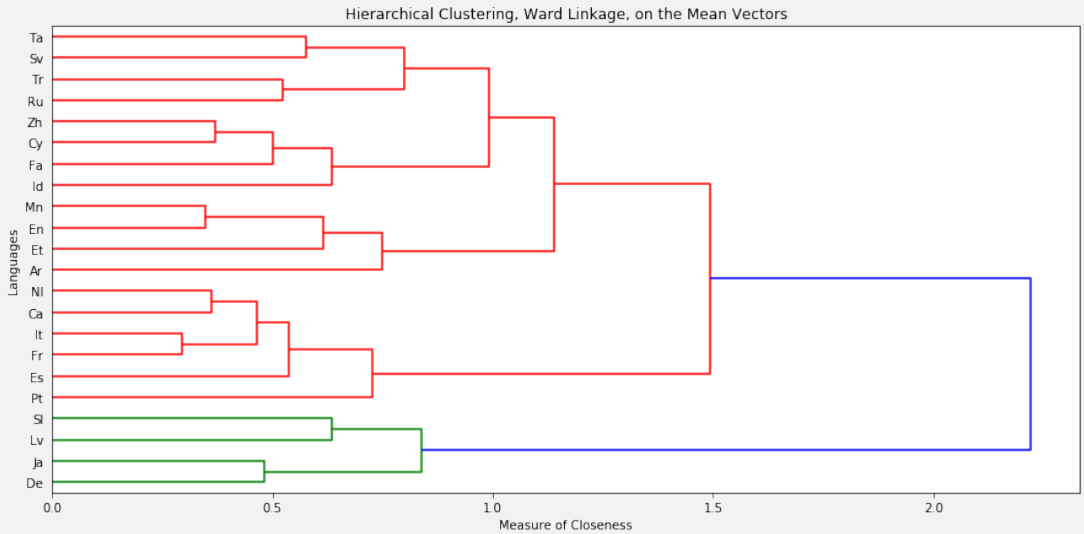
Question 3 continued on next page...

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.



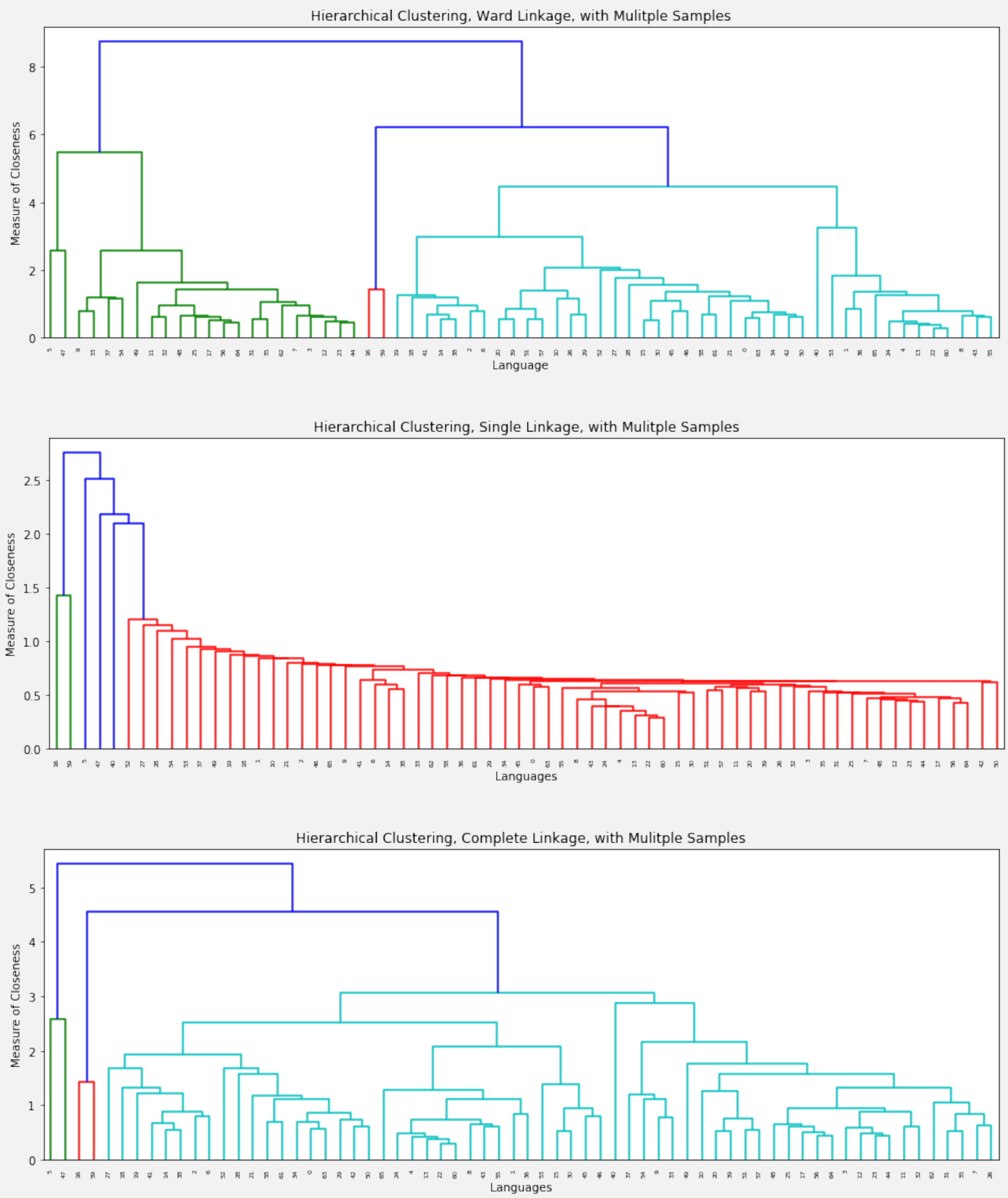
We do see a similarity in shape of plots for the mean vectors and centres, where the centres are more dispersed.

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.



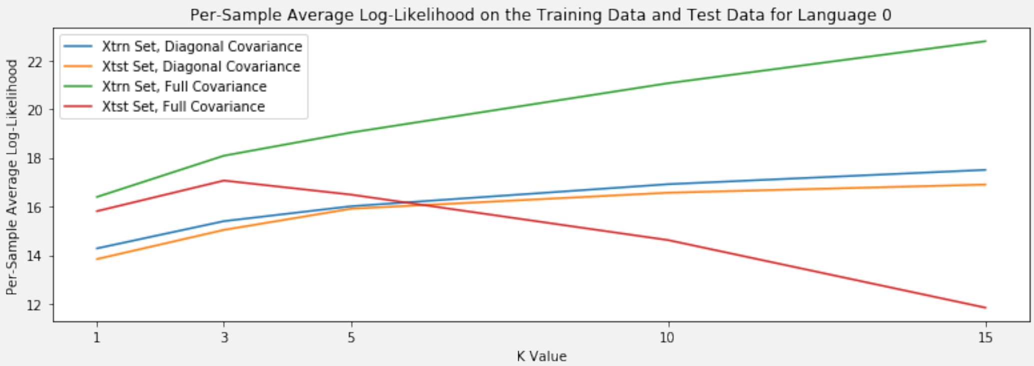
Here we can see we start with 8 new clusters where the closest languages have been linked, which can see along the x-axis. The distance measured is the error sum of squares between the mean vectors of the languages.
If we were to set the threshold, say between 1.5 and 2.0, we would have 2 final clusters, one containing 17 languages and the other 4. These clusters could be interpreted as predictions of families of languages or origin.

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



The 3 methods all differ in terms of maximum measure of closeness - *Single* has the lowest at just over 2.5 and *Complete* has the highest at just over 8. Pearson's correlation coefficient tells us that there is a very low degree of correlation between *Ward* and *Single* at only 0.19, a high degree of correlation between *Single* and *Complete* at 0.72 and also between *Complete* and *Ward* at 0.53.

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.,



	Per-Sample Average Log-Likelihood			
	Diagonal Covariance		*Full Covariance*	
	Training Set	*Test Set*	*Training Set*	*Test Set*
$K = 1$	~ 14.28	~ 13.84	~ 16.39	~ 15.81
$K = 3$	~ 15.40	~ 15.04	~ 18.09	~ 17.07
$K = 5$	~ 16.01	~ 15.91	~ 19.04	~ 16.49
$K = 10$	~ 16.92	~ 16.57	~ 21.06	~ 14.62
$K = 15$	~ 17.50	~ 16.90	~ 22.79	~ 11.85

The average likelihood start higher when a full covariance matrix is used, however after 3 components, the *Test Set* begins to over-fit and between $k = 3$ and $K = 15$ it dropped by 5.22. Apart from that, the other 3 sets of data gradually increase as K increases - this is a good thing.