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

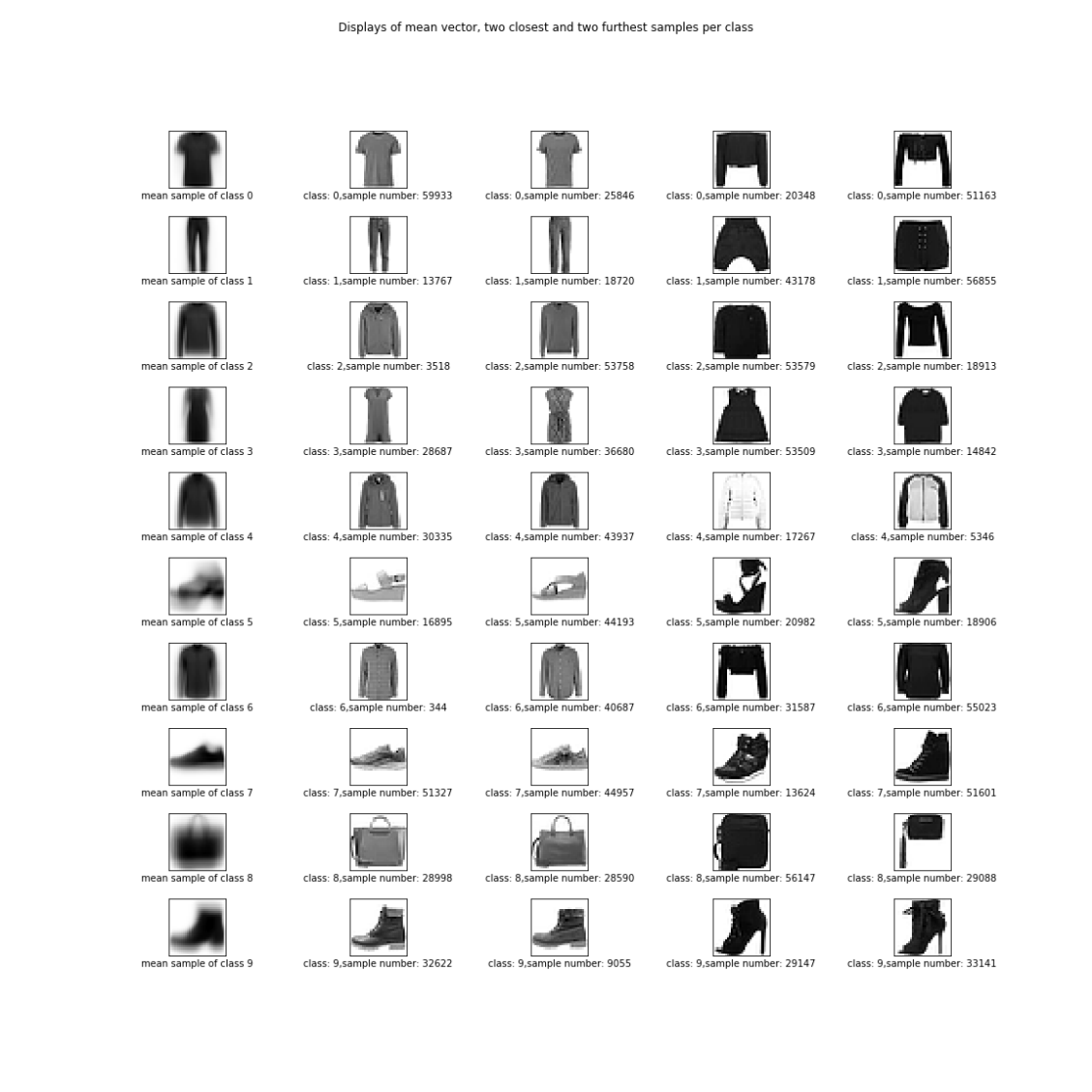
In this question we employ PCA to analyse image data

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,:]`.

First four elements for the first training sample
-3.1373×10^{-6}, -2.2680×10^{-5}, -1.1797×10^{-4}, -4.0706×10^{-4}
First four elements for the last training sample
-3.1373×10^{-6}, -2.2680×10^{-5}, -1.1797×10^{-4}, -4.0706×10^{-4}

Question 1 continued on next page...

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.



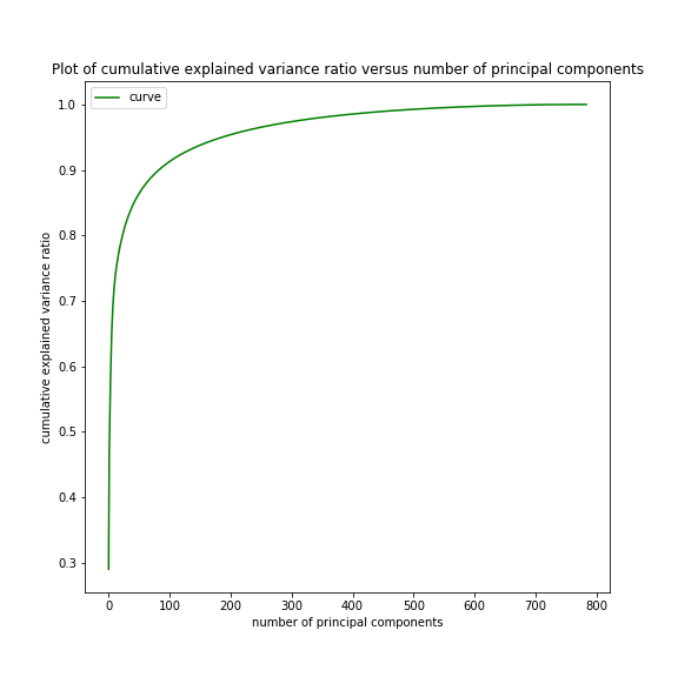
The two nearest samples of a class are representative of what most of the typical elements of the class look like while the two furthest samples are representative of what a few anomalous elements of that class look like. The value of each feature in the mean sample of a given class is the average value of that particular feature over all the elements of that class. This explains why the image of the mean sample looks blurred.

1.3 (3 points) Apply Principal Component Analysis (PCA) to the data of `Xtrn_nm` using sklearn.decomposition.PCA, and find the cumulative explained variance.

Cumulative explained variance : 68.2174
Variances of first five principal components:
19.8098, 12.1122, 4.1062, 3.3818, 2.6248

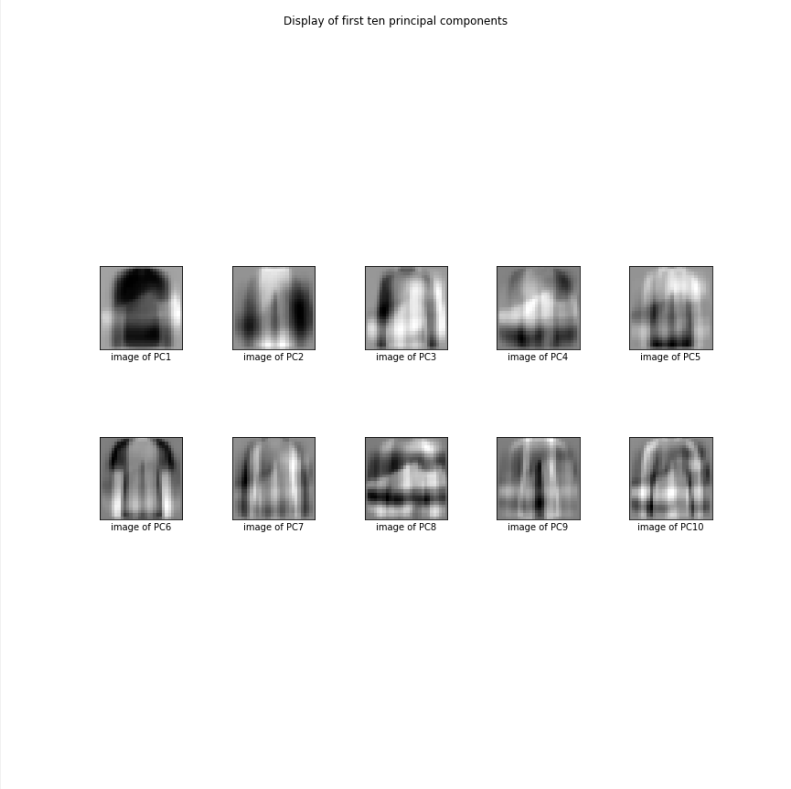
Question 1 continued on next page...

1.4 (3 points) Plot a graph of the cumulative explained variance ratio. Discuss the result briefly.



The amount of variance contained by kth principal component decreases as k increases. As such, the cumulative variance increases rapidly for the first 100 principal components. Later on its rate of increase slows down to just above zero as the remaining principal components contribute very little to the cumulative variance.

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.



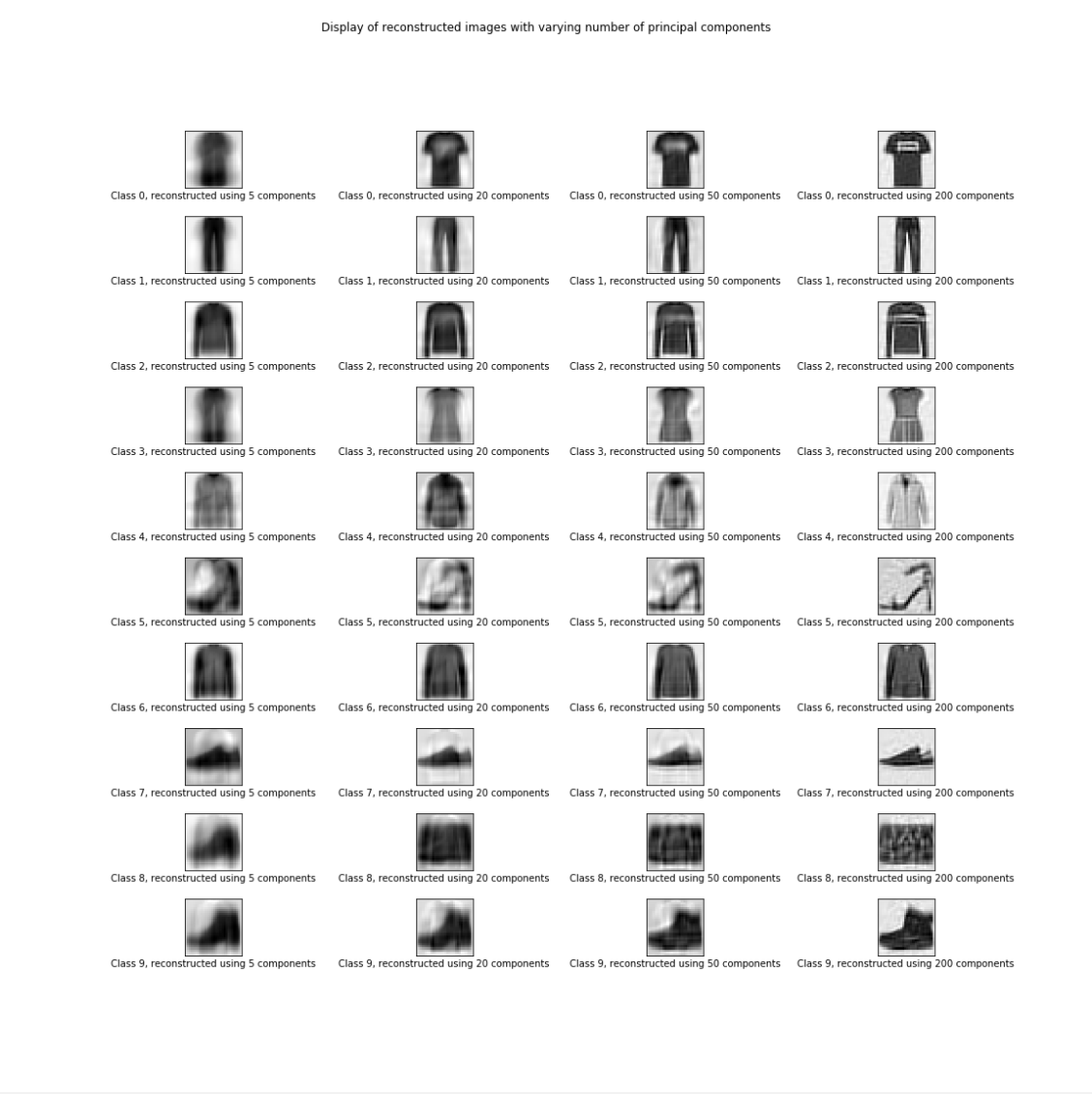
Each principal component contains the characteristic features present in some or all of the classes. For the first six components, these characteristic features are striking but as the number of the component increases, these distinctive features become blurred i.e. less noticeable.

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.

Class	K=5	K=20	K=50	K=200
0	0.2561	0.1500	0.1276	0.0601
1	0.1980	0.1404	0.09508	0.0354
2	0.1987	0.1456	0.1234	0.0783
3	0.1457	0.1073	0.0834	0.0560
4	0.1182	0.1026	0.0879	0.0464
5	0.1811	0.1587	0.1428	0.0901
6	0.1295	0.0958	0.0723	0.0458
7	0.1656	0.1278	0.1066	0.0622
8	0.2234	0.1450	0.1236	0.0922
9	0.1835	0.1511	0.1219	0.0728

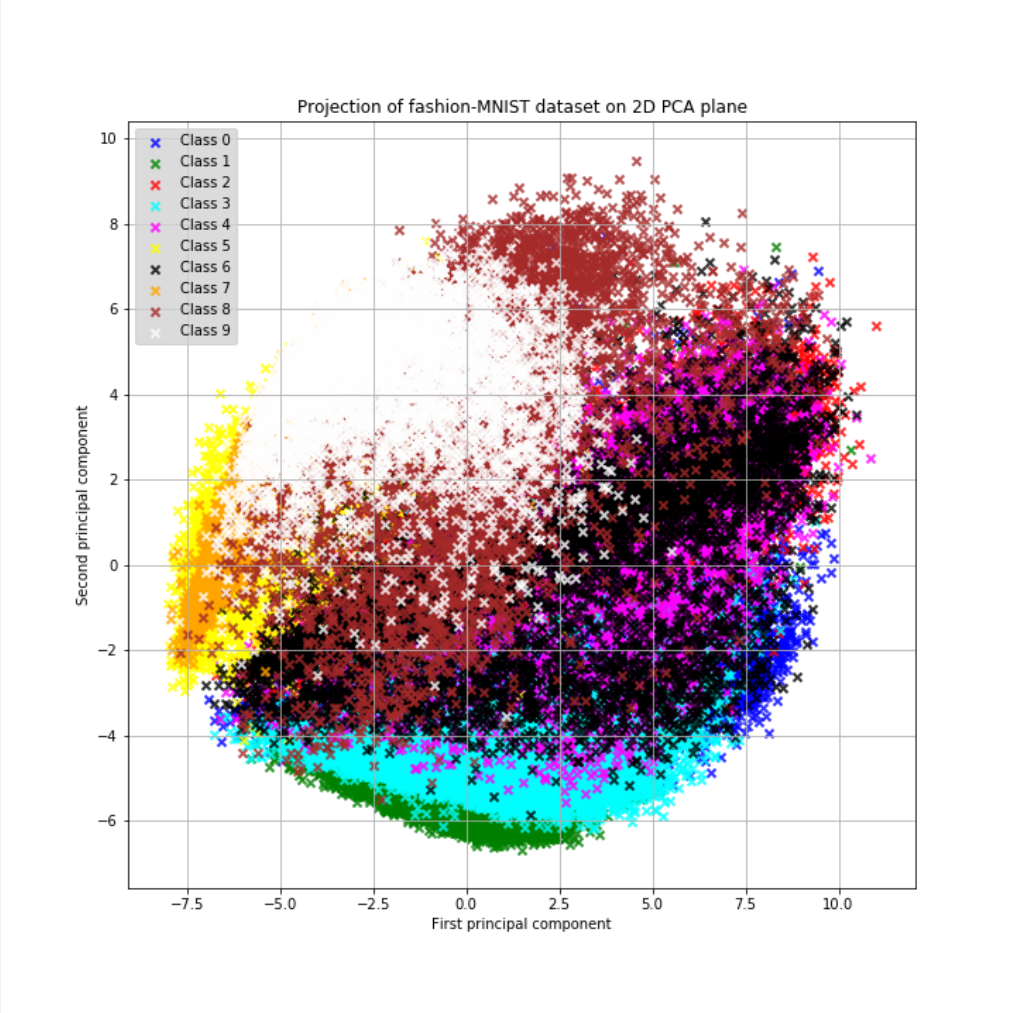
Question 1 continued on next page...

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



As the number of components increases, the quality of the reconstructed image increases. Each additional principal component increases the amount of information needed to distinguish one sample in the dataset from another.

1.8 (4 points) Plot all the test 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.



The classes are inseparable when projected on the 2D PCA plane since there are only two principal components involved. In general, projections on lower dimensions lead to loss of information. If we were to increase the number of principal components, we would get clearer boundaries between the different 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.

Classification accuracy : 0.8398

Confusion matrix :

$$\begin{pmatrix} 819 & 4 & 15 & 50 & 7 & 4 & 88 & 1 & 12 & 0 \\ 5 & 953 & 4 & 27 & 5 & 0 & 3 & 1 & 2 & 0 \\ 27 & 4 & 731 & 11 & 133 & 0 & 82 & 2 & 9 & 1 \\ 29 & 17 & 14 & 866 & 33 & 0 & 37 & 0 & 4 & 0 \\ 1 & 4 & 115 & 39 & 759 & 2 & 71 & 0 & 9 & 0 \\ 2 & 0 & 0 & 1 & 0 & 912 & 0 & 56 & 9 & 20 \\ 149 & 3 & 126 & 47 & 108 & 1 & 537 & 0 & 28 & 1 \\ 0 & 0 & 0 & 0 & 0 & 32 & 0 & 936 & 1 & 31 \\ 7 & 1 & 6 & 11 & 3 & 7 & 15 & 5 & 944 & 1 \\ 0 & 0 & 0 & 1 & 0 & 15 & 1 & 42 & 0 & 941 \end{pmatrix}$$

Question 2 continued on next page...

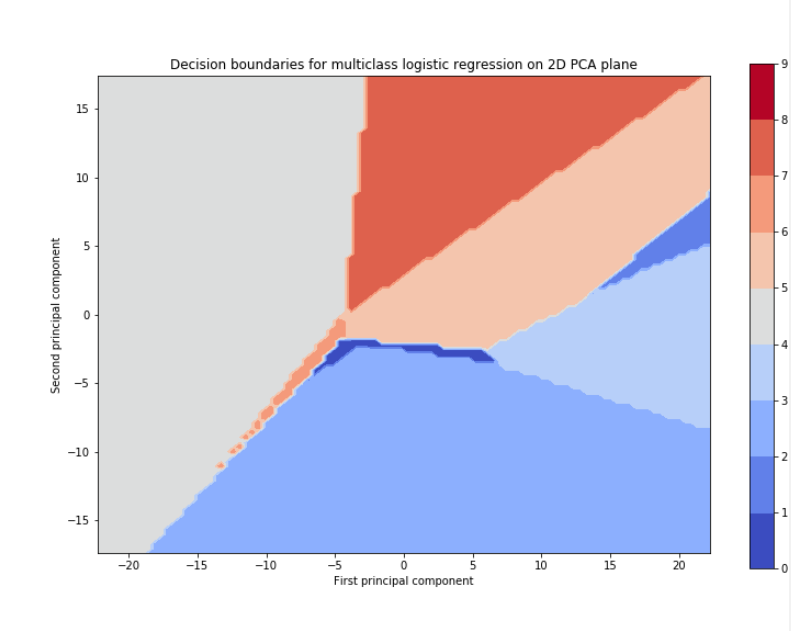
2.2 (3 points) Carry out a classification experiment with SVM classifiers, and report the mean accuracy and confusion matrix (in numbers) for the test set.

Mean accuracy : 0.8462

Confusion matrix :

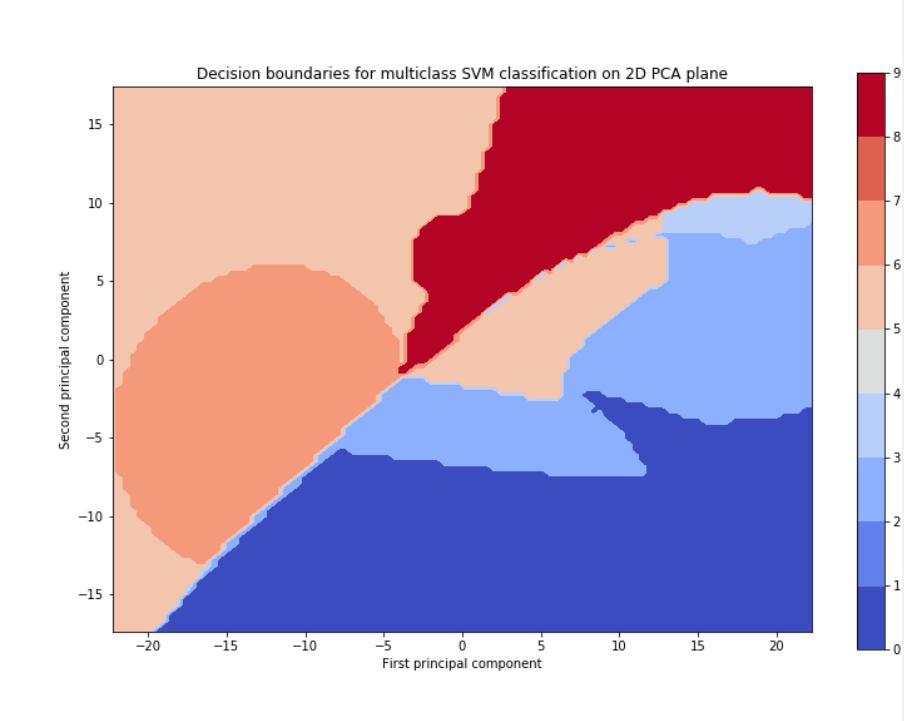
$$\begin{pmatrix} 845 & 2 & 8 & 51 & 4 & 4 & 72 & 0 & 14 & 0 \\ 4 & 951 & 7 & 31 & 5 & 0 & 1 & 0 & 1 & 0 \\ 16 & 2 & 751 & 11 & 136 & 0 & 76 & 0 & 8 & 0 \\ 32 & 6 & 12 & 883 & 26 & 0 & 38 & 0 & 3 & 0 \\ 1 & 0 & 98 & 38 & 773 & 0 & 86 & 0 & 4 & 0 \\ 0 & 0 & 0 & 1 & 0 & 911 & 0 & 60 & 2 & 26 \\ 185 & 1 & 122 & 39 & 95 & 0 & 533 & 0 & 25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 34 & 0 & 926 & 0 & 40 \\ 3 & 1 & 8 & 5 & 2 & 4 & 13 & 4 & 959 & 1 \\ 0 & 0 & 0 & 0 & 0 & 22 & 0 & 47 & 1 & 930 \end{pmatrix}$$

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



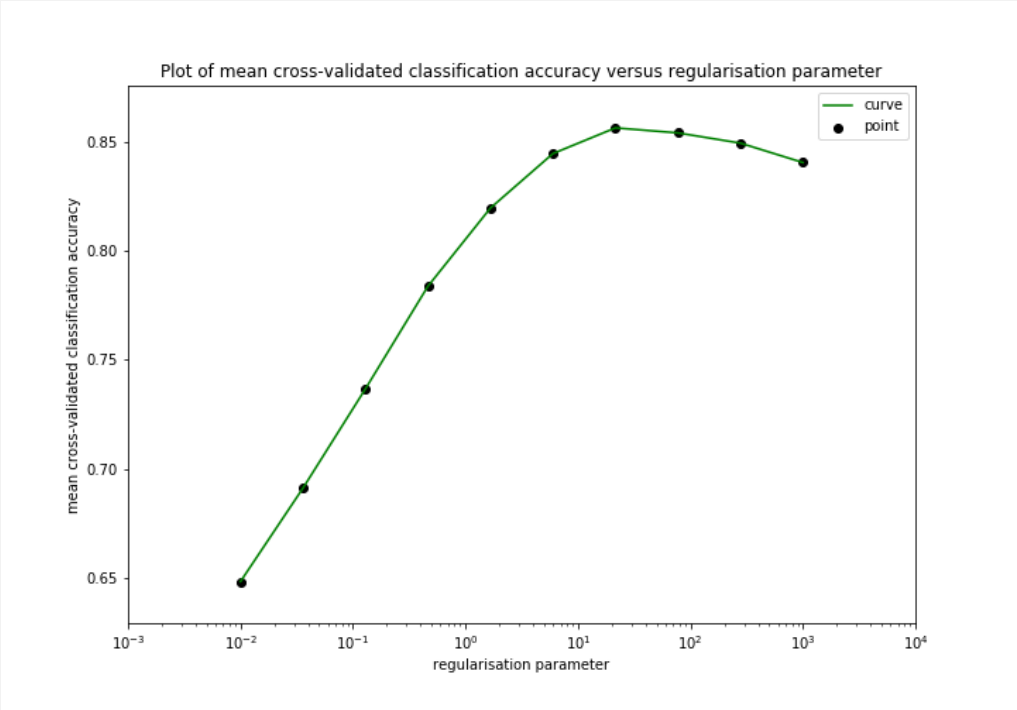
Elements of class 9 are not present among the points. When we transform a data point z on the plane to its corresponding point x in the original space, it is very unlikely that x will be a point in the dataset. This is due to loss of information as a result of the 2D representation. So the points we obtain in the original space are not representative of all the classes. As a linear solver was used, the decision boundaries for the classes are approximately linear. Due to the large size of the dataset, it takes a considerable amount of time to build the model and predict the classes of the testing data.

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.



This time around, elements of class 8 are absent. Class 1 has significantly more elements when the SVM is used. In general, the 'average' position where most classes are located is similar for both the Logistic Regression and SVM classifiers. As radial basis functions are used, the decision boundaries are non-linear. Due to the large size of the dataset, it takes a considerable amount of time to build the model and predict the classes of the testing data.

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 accuracy score : 85.65%
Value of C which yielded highest accuracy : 21.5443

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 training set : 90.84%
Classification accuracy on testing set : 87.70%

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

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

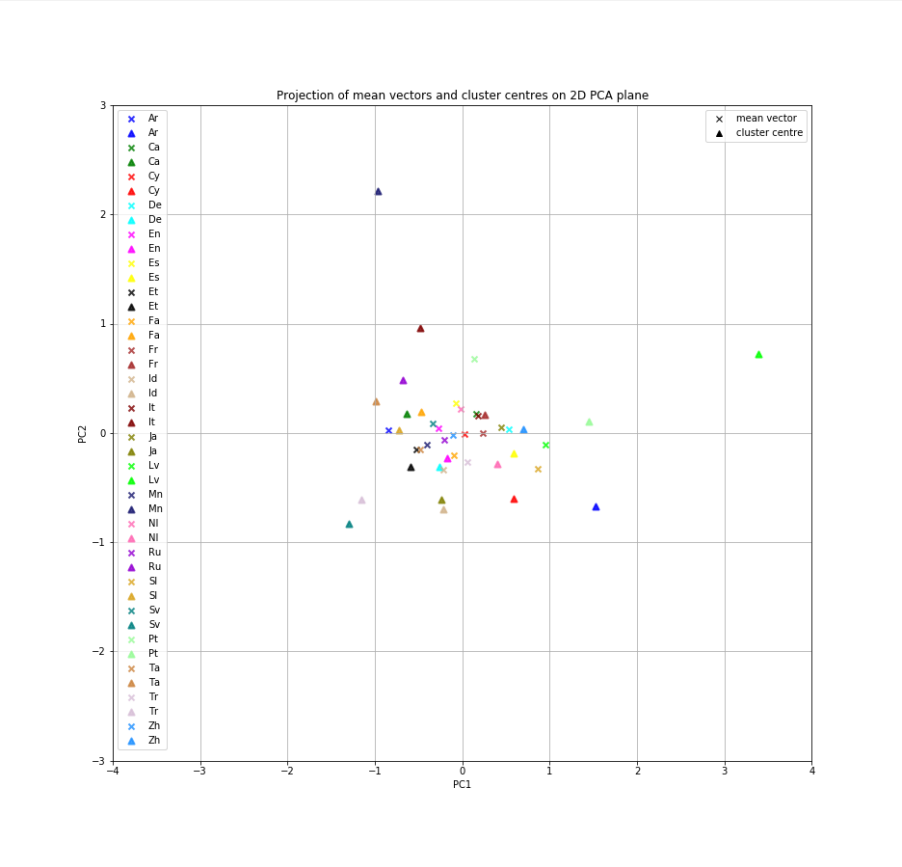
3.1 (3 points) Apply k-means clustering on `Xtrn` for $k = 22$, where we use sklearn.cluster.KMeans with the parameters `n_clusters=22` and `random_state=1`. Report the sum of squared distances of samples to their closest cluster centre, and the number of samples for each cluster.

Inertia : 38185.817

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

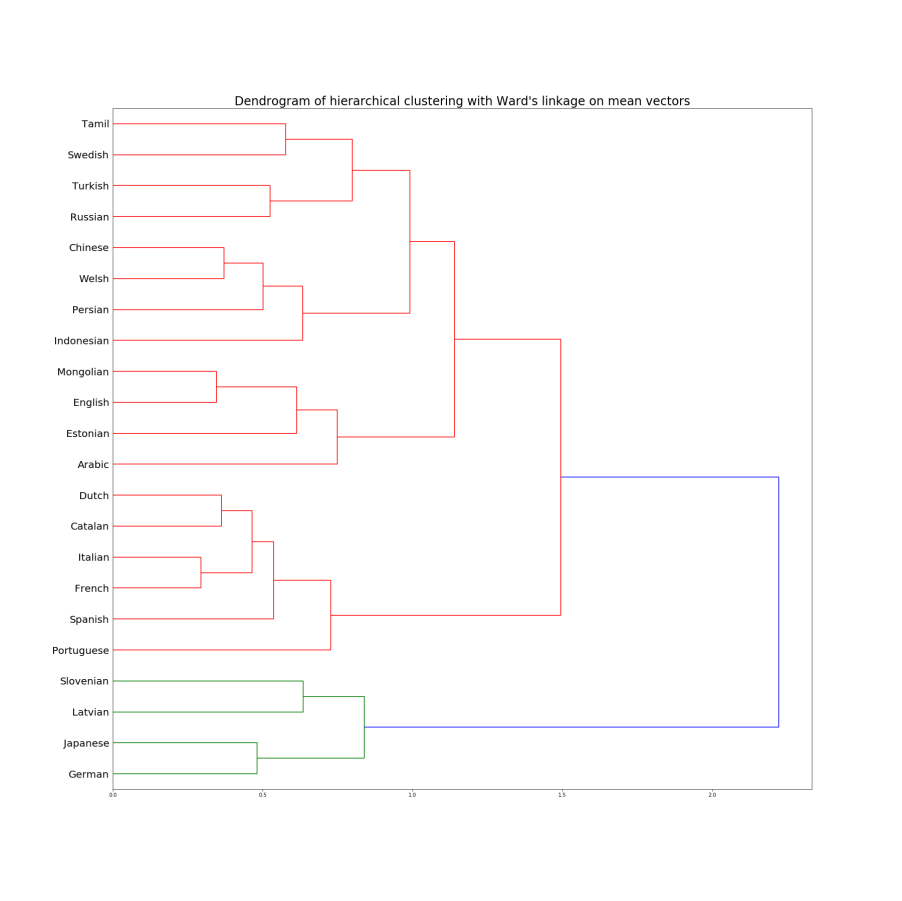
Question 3 continued on next page...

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.



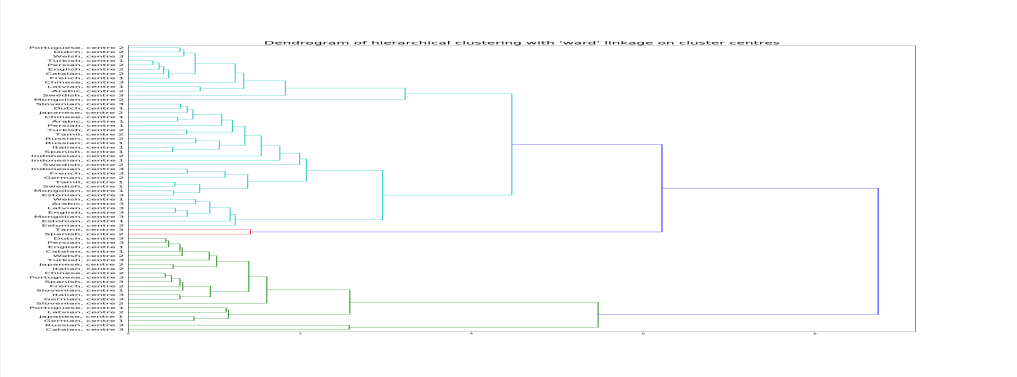
Mean vectors and cluster centres not similar for most of the languages. We expect almost all the cluster centres to be close to the mean vectors but due to the extreme simplification applied to the data this is not the case.

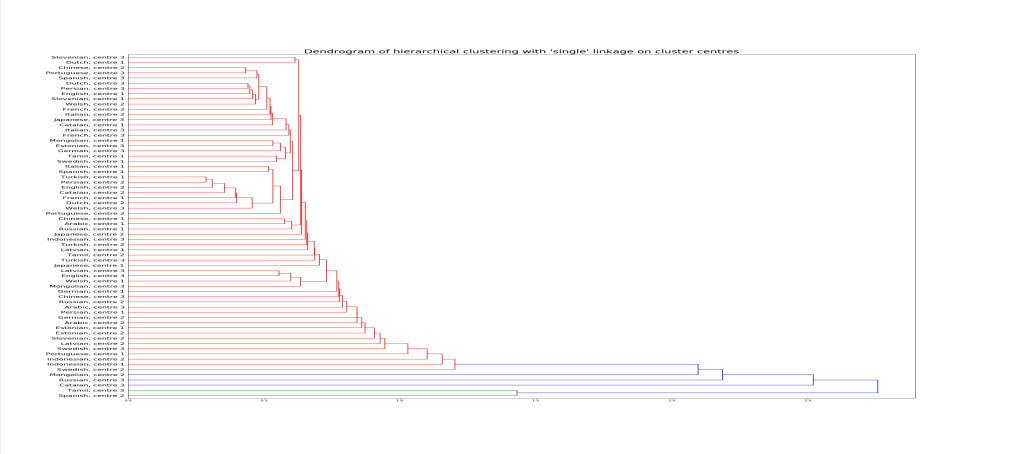
3.3 (3 points) We now apply hierarchical clustering on the training data set to see if there are any structures in the spoken languages.

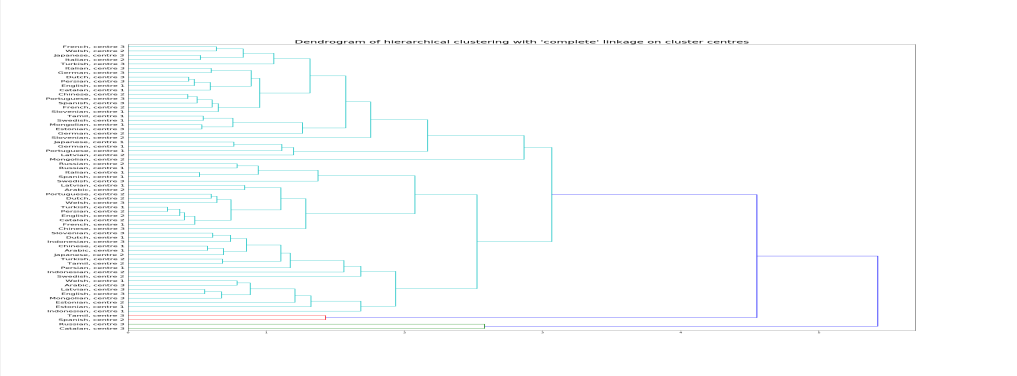


We notice that languages belonging to geographically close regions are in the same hierarchical cluster most of the time.

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



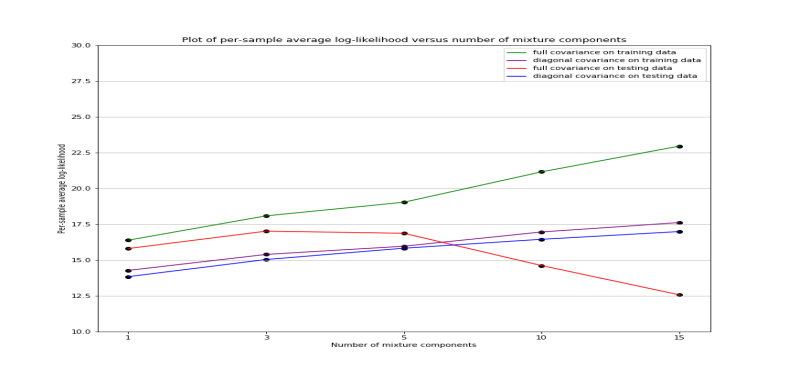




From the dendrograms we see that the different cluster centres for a given language usually belong to different hierarchical clusters. This means that the way some words of a given language are pronounced can be similar to the way some words of another language are pronounced. This could be because words belonging to different languages may have the same region of origin.

Question 3 continued on next page...

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.,



Dataset	Covariance	K=1	K=3	K=5	K=10	K=15
Training set	Full	16.3936	18.0596	19.0654	21.0328	22.8094
	Diagonal	14.2804	15.3986	15.9279	16.9601	17.5824
Testing set	Full	15.8105	16.9898	16.5821	15.1745	12.3302
	Diagonal	13.8429	15.0414	15.6240	16.4060	17.0772

In general, as the number of mixture components increases, the per-sample average log-likelihood (which is a measure of the performance of the GMM model) increases. There are 26 features for a given language, which is likely also same as the number of mixture components of the optimal GMM model for the language. This explains why as we increase the number of mixture components from 1 to 15, we are able to better express the language GMM as a linear combination of its components. The GMM with full covariance performs poorly on the testing data due to an insufficient amount of samples.