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 following table shows the first and last samples as rows and the values for the first 4 elements as columns, where the column name is the element index.

Training Sample	0	1	2	3
First	-3.137e-6	-2.268e-5	-1.180e-4	-4.071e-4
Last	-3.137e-6	-2.268e-5	-1.180e-4	-4.071e-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.

From the figure below, we can see that the mean vector for each class is basically an average representation of the clothing item representing the class, which makes sense intuitively. The two closest samples are simply the two pieces of clothing that resemble the mean vector (i.e. the simplest t-shirt, pants etc), and the two furthest samples are the ones with the most differences (i.e. crop-tops, skirts etc).

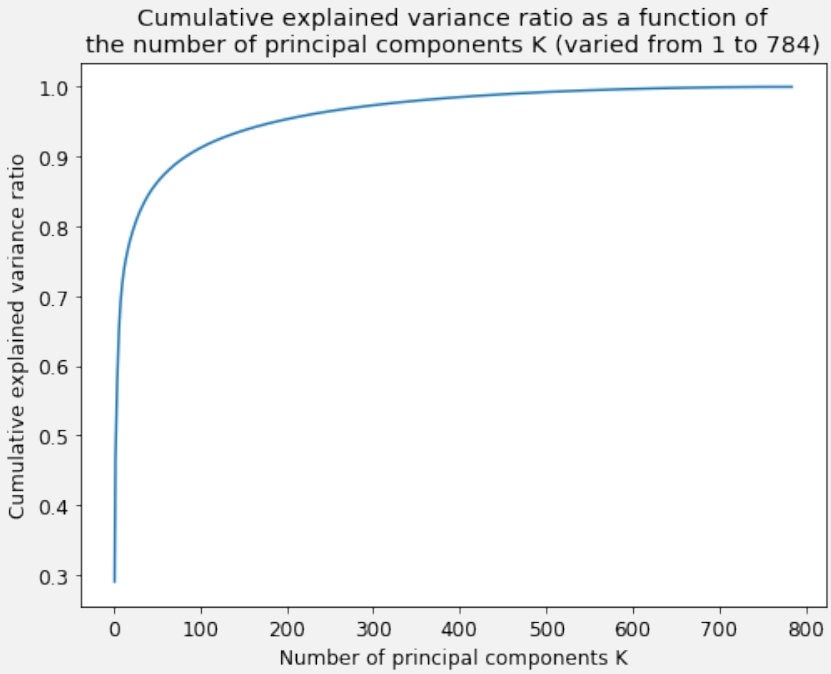
1.3 (3 points) Apply Principal Component Analysis (PCA) to the data of `Xtrn_nm` using sklearn.decomposition.PCA, and report the variances of projected data for the first five principal components in a table. Note that you should use `Xtrn_nm` instead of `Xtrn`.

The rows of following table show the first five principal components. The columns contain the values for the variance of the projected data on each principal component and the corresponding percentage of the total variance.

Principal Component	Variance explained	Percentage of total variance
1	19.810	29.039%
2	12.112	17.755%
3	4.106	6.019%
4	3.382	4.957%
5	2.625	3.848%

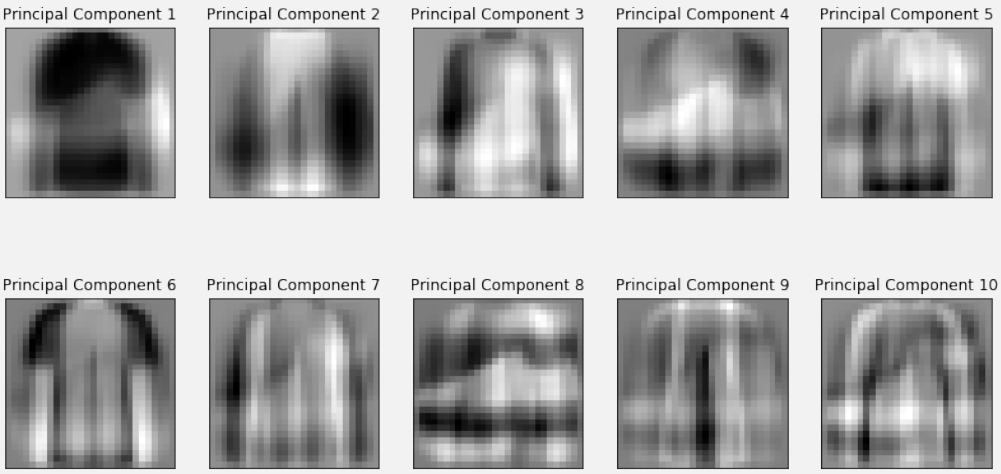
Question 1 continued on next page...

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.



In the plot above, we can observe that as K increases, the cumulative explained variance increases as well, but at a decreasing rate. The first 80% of the total variance is reached with K<50, and the 90% with K around 100. The remaining 10% is gained very slowly as K increases to 784, where it finally reaches 100%.

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.



The way I understand the findings above is that each principal component has two different and quite opposing items of clothing (shirt vs pants, or bag vs shoe) as positive and negative (black and white in the pictures; grey areas are neutral, or 0, because they don't contain important details there). The reason is that the main differences between the samples is basically the shape and details of each item of clothing. This way, projecting a new sample to this principal component is like saying "this new sample looks more like a shirt than pants", which makes sense intuitively. In addition, the more principal components one adds, the more classes are covered and more details are captured, thus explaining more of the total variance in the input data, as we saw above in question 1.4.

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.

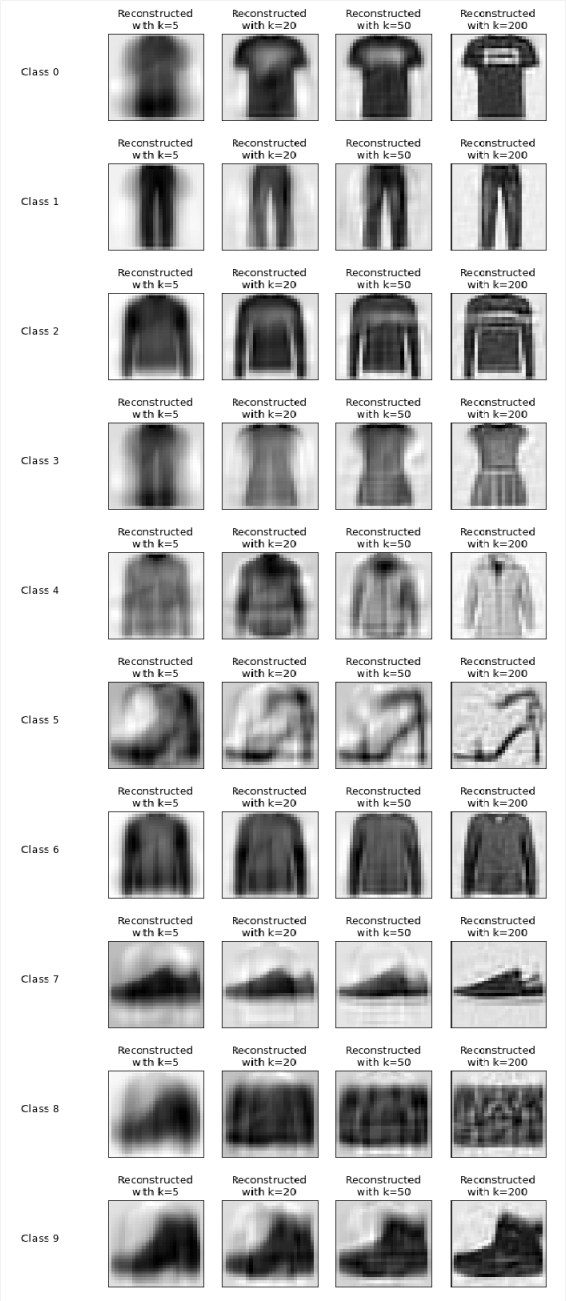
The table below lists the RMSE values between the original first sample of each class and the reconstructed one, using principal components $K = 5, 20, 50, 200$.

Class	K=5	K=20	K=50	K=200
0	0.256	0.150	0.127	0.060
1	0.198	0.140	0.094	0.038
2	0.199	0.146	0.123	0.079
3	0.146	0.107	0.084	0.057
4	0.118	0.103	0.088	0.048
5	0.181	0.159	0.143	0.092
6	0.129	0.096	0.072	0.047
7	0.166	0.128	0.107	0.063
8	0.223	0.145	0.123	0.092
9	0.184	0.151	0.121	0.072

Question 1 continued on next page...

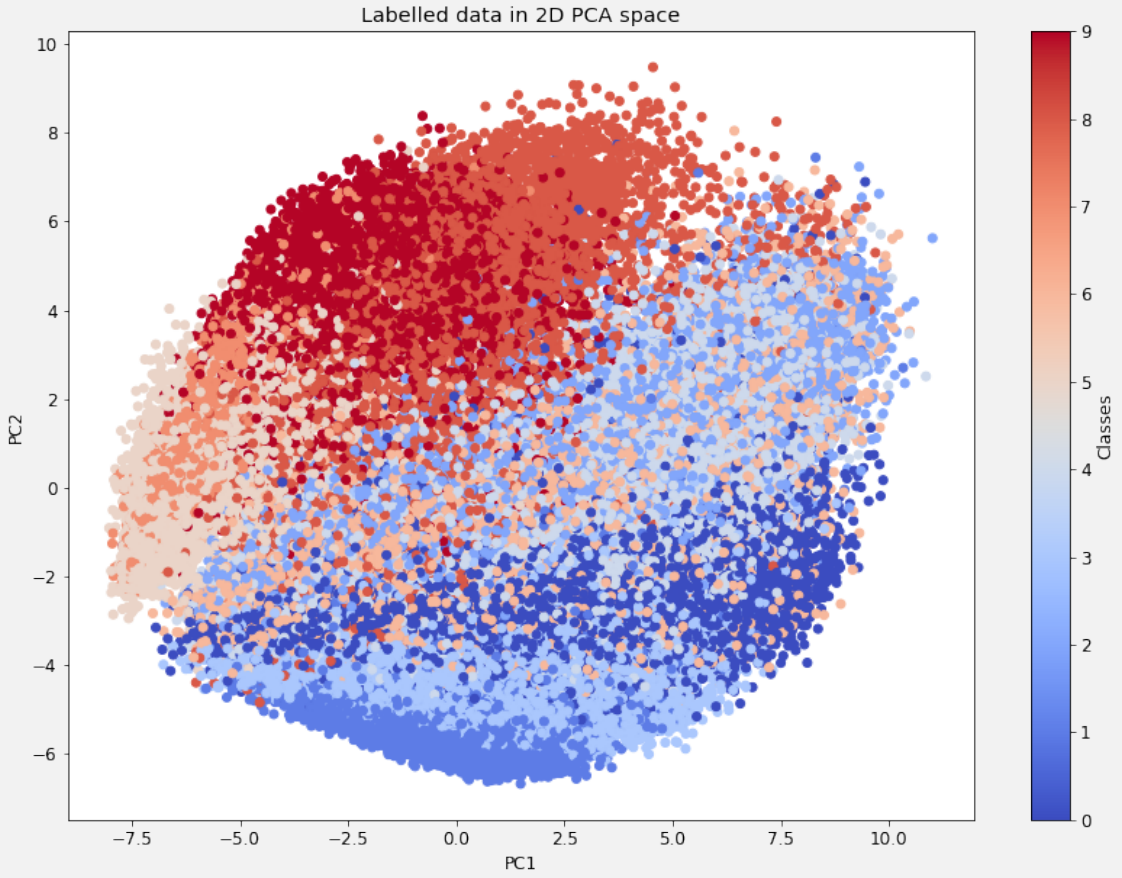
1.7 (4 points) Display the image for each of the reconstructed samples in a 10-by-4 grid, where each row corresponds to a class and each row column corresponds to a value of $K = 5,\ 20,\ 50,\ 200$.

The figure below shows how the first sample of each class is reconstructed using principal components $K = 5, 20, 50, 200$. It can clearly be seen that as the number of principal components increases, the reconstructed picture gets more clear and thus closer to the actual picture of the sample. This is because as the number of principal components increase, more and more details are captured, thus the reconstructed picture represents the original sample more accurately.



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.

The figure below shows all the training samples plotted on the two-dimensional PCA plane. The colour of each point represents its corresponding class. Even though all the points together look like a big clump, there is a consistent separation of the classes along the first two principal components, and it can be seen that points of the same class are usually close together. Classes such as 1, 5, 8 and 9 are more separated from the rest and have more distinct clumps.



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.

Logistic Regression classification accuracy on test set: 0.840

The Confusion Matrix:

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...

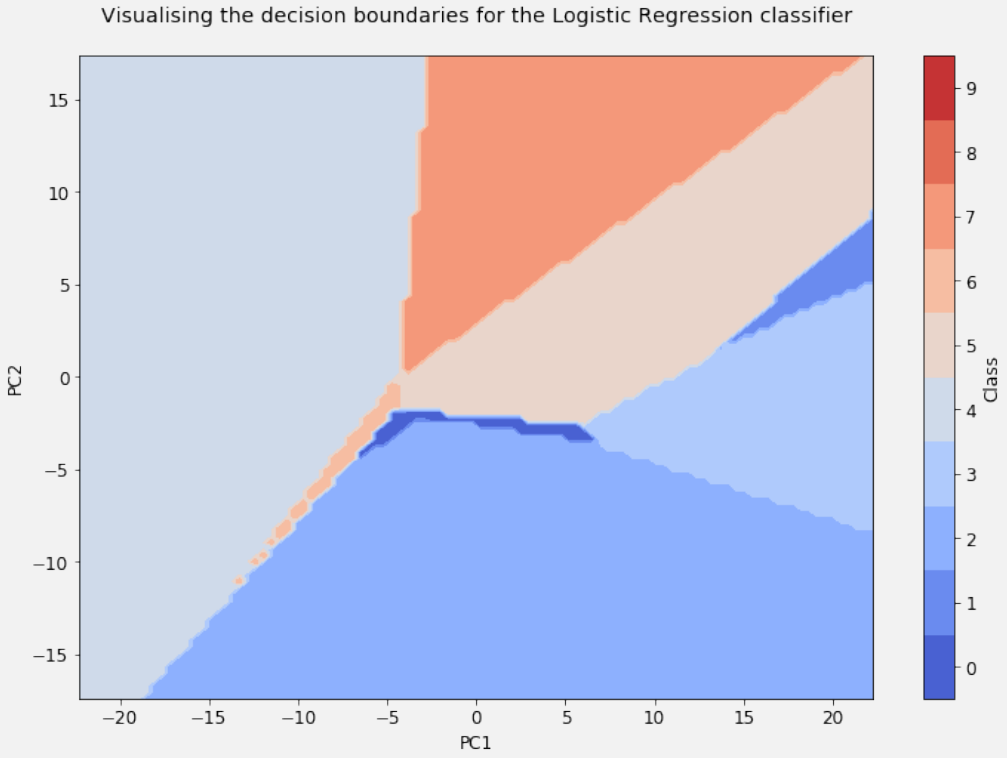
2.2 (3 points) Carry out a classification experiment with SVM classifiers, and report the mean accuracy and confusion matrix (in numbers) for the test set.

RBF SVM classification accuracy on test set: 0.846

The Confusion Matrix:

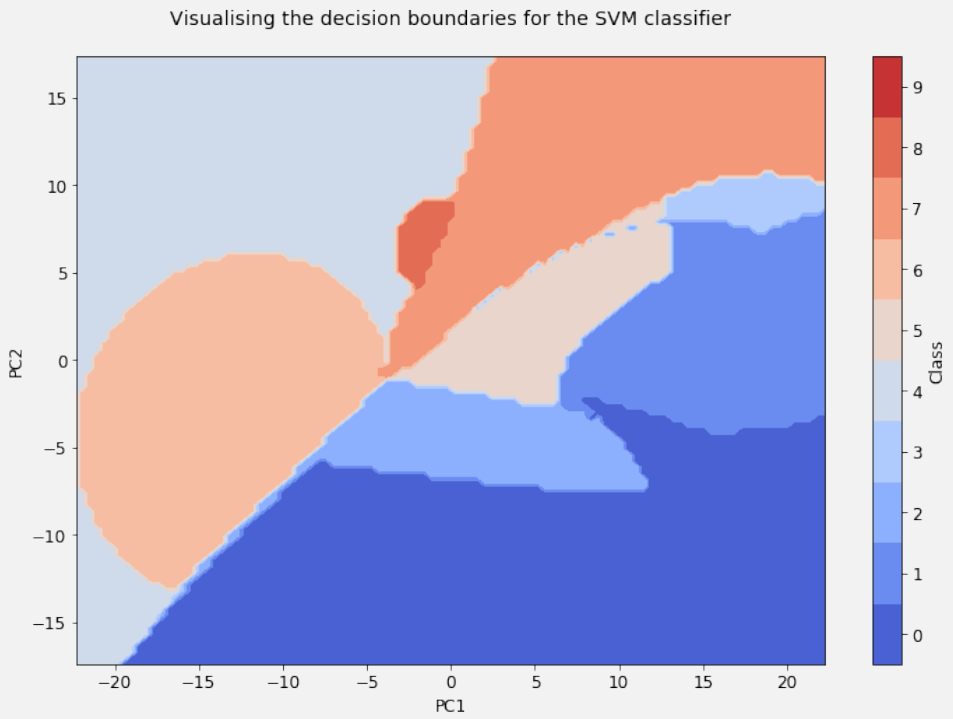
845	2	8	51	4	4	72	0	14	0
4	951	7	31	5	0	1	0	1	0
15	2	748	11	137	0	79	0	8	0
32	6	12	881	26	0	40	0	3	0
1	0	98	36	775	0	86	0	4	0
0	0	0	1	0	914	0	57	2	26
185	1	122	39	95	0	533	0	25	0
0	0	0	0	0	34	0	925	0	41
3	1	8	5	2	4	13	4	959	1
0	0	0	0	0	22	0	47	1	930

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



On the figure above we visualise the decision boundaries between classes for the Logistic Regression classifier, on the 2-dimensional plane spanned by the first 2 Principal Components. First, we notice that the boundaries are linear, as expected, since Logistic Regression is a linear classifier. It is also noted that the chosen cross-section of the original space does not contain points from all 10 classes, thus a couple of classes (8 and 9) are missing from the plot above (no boundaries for them).

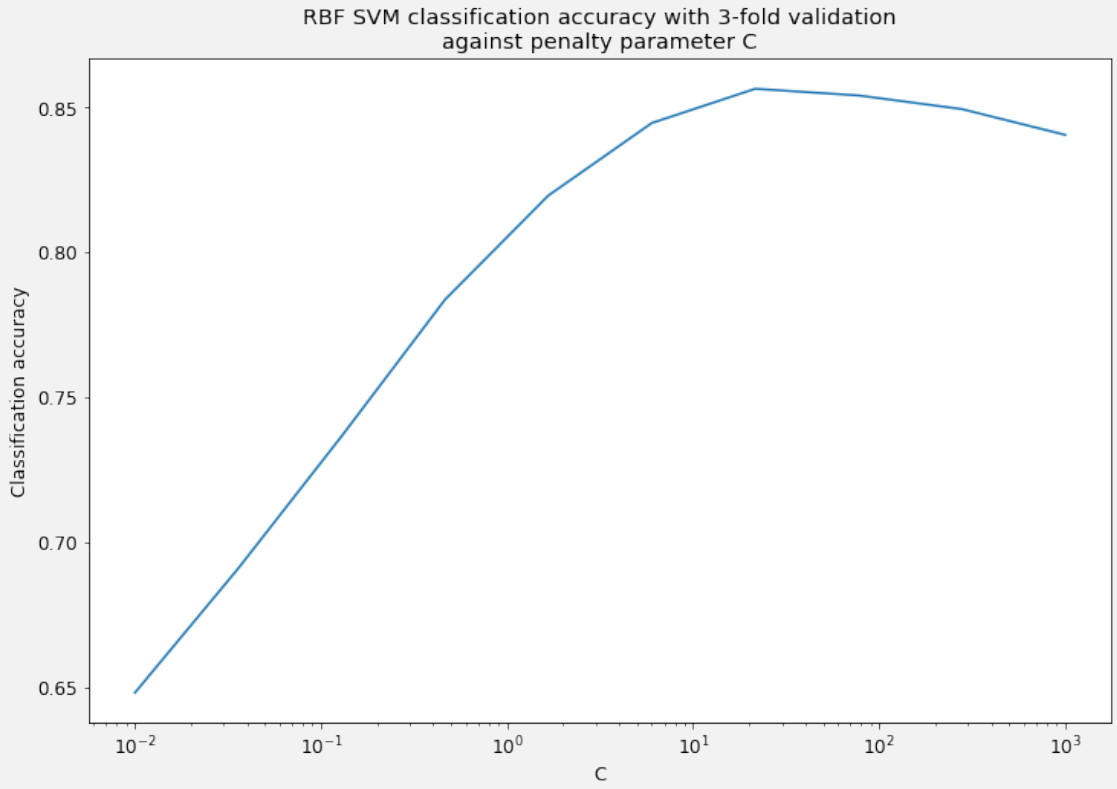
2.4 (4 points) Using the same method as the one above, plot the decision regions for the SVM classifier you trained in Question 2.2. Comparing the result with that you obtained in Question 2.3, discuss your findings briefly.



Now, in contrast to Logistic Regression, the boundaries are nonlinear, as expected, since SVM is a nonlinear classifier. There are some similarities between the two figures, such as the locations of classes 4 to 7, but also some differences such as the location and size of classes 0 to 3 (e.g. the bottom right of the plot is dominated by class 0 in this case, but it was class 2 for Logistic Regression). Again, we can see that there is no boundary for class 9, since there was no sample in this cross-section, but now a small bubble boundary for class 8 has appeared.

Question 2 continued on next page...

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 obtained mean accuracy score was 0.857,
with the value of $C = 21.544$.

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.

Using the optimal value of $C = 21.544$ and training the RBF SVM classifier on the whole training set, I obtain:

Classification accuracy on train set: 0.908
Classification accuracy on test set: 0.877

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

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

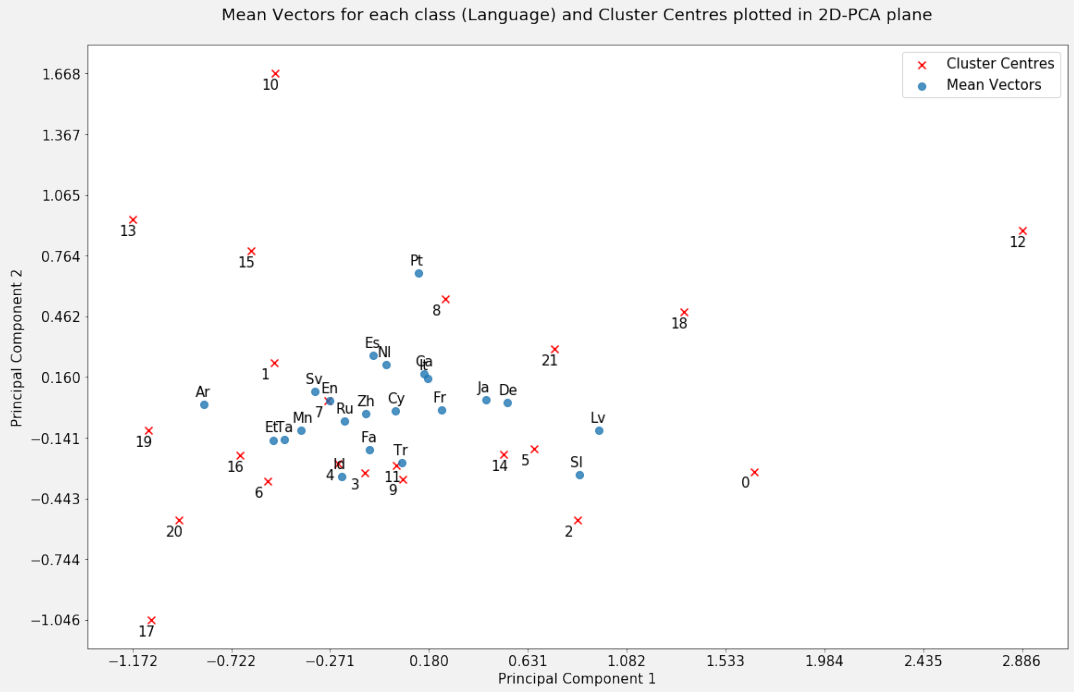
3.1 (3 points) Apply k-means clustering on `Xtrn` for $k = 22$, where we use sklearn.cluster.KMeans with the parameters `n_clusters=22` and `random_state=1`. Report the sum of squared distances of samples to their closest cluster centre, and the number of samples for each cluster.

The sum of squared distances of samples to their closest cluster centre is 38185.817. The table below lists the 22 cluster centres and the number of samples in each cluster after k-means.

Cluster Centre	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

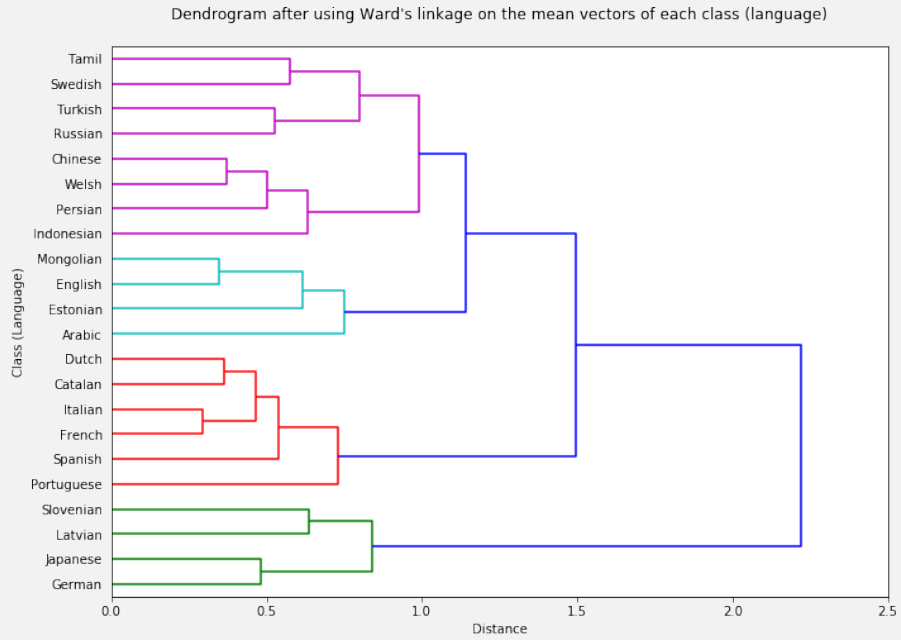
Question 3 continued on next page...

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.



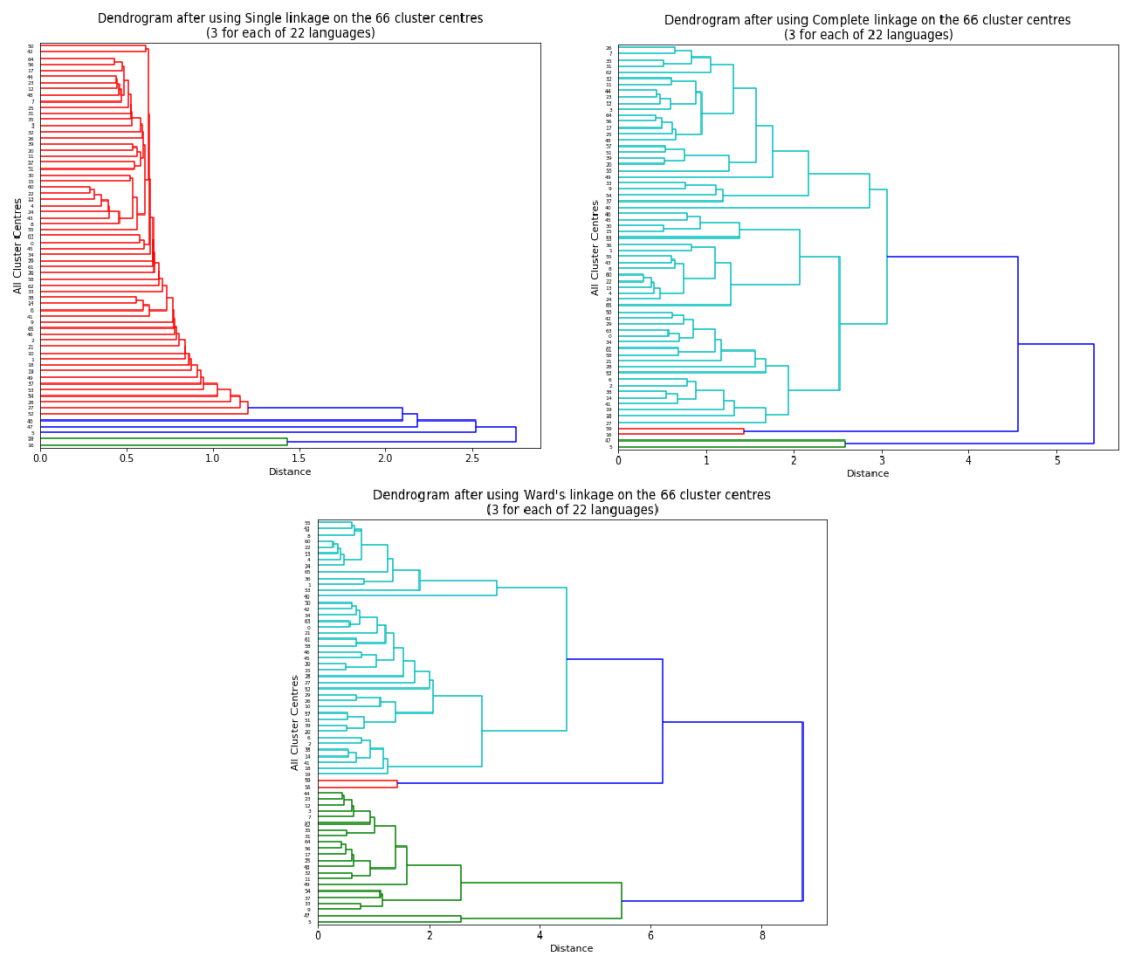
The mean vectors of each class are not similar to the locations of the 22 cluster centres. This could be because of the extreme simplification of the data that occurred before clustering, which probably led to losing important details and features of the data. This, in turn, means that the data is not well-separated by class, thus k-means clustering simply captures data points that are close together, even though they might not be from the same class. Also, we can see that the cluster centres are more spread than the mean vectors (e.g. clusters 10, 12, 13, 17 etc) and this could be because of large spread in the data and outliers which affect the cluster centres locations.

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.



After applying hierarchical clustering with Ward's linkage on the 22 mean vectors, we can see some underlying structure in the data. Languages such as Italian and French, Mongolian and English, Chinese and Welsh, are joined together early in the algorithm which indicates a small distance between them, thus they are similar according to this method. Eventually, the group clusters get filled in with other similar languages, and at distance 1.0, where I set it as my colour threshold, we can see four different clusters (purple, cyan, red, green), thus four families of languages. Eventually, at a distance around 2.2, all the clusters are merged into one.

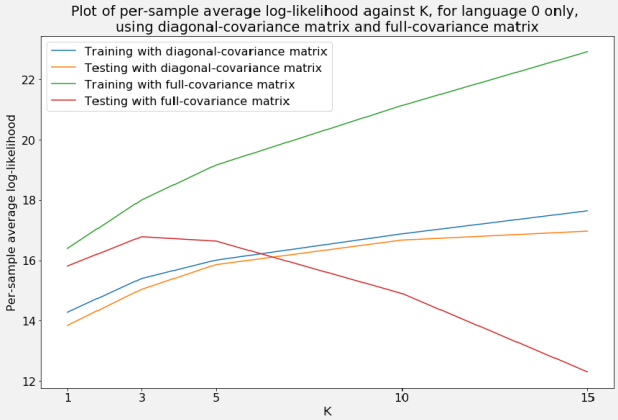
3.4 (5 points) We here extend the hierarchical clustering done in Question 3.3 by using multiple samples from each language.



From the figures above, we can see that each method groups the clusters in a different way. Single linkage produces a long chain since it merges based on the shortest distance between closest elements in the clusters. On the other hand, complete linkage produces more 'spherical' clusters with consistent 'diameter', since it merges based on the furthest elements in the clusters. It takes a distance of around 2.7 for single linkage to merge all clusters into one, but around 5.3 for complete linkage, which almost twice as long. Finally, Ward's method forms lots of small clusters initially at small distance, and then less and less merges happen with increasing distance. The final merge happens at around 9.0, which is almost twice that of complete linkage, thus the longest distance from all three methods.

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.,



K	Train Diag	Test Diag	Train Full	Test Full
1	14.280	13.843	16.394	15.811
3	15.398	15.039	18.091	17.065
5	15.982	15.830	19.005	16.592
10	16.858	16.416	21.169	14.858
15	17.688	17.092	22.806	11.871

The figure above, which plots the values listed in the table, shows that the per-sample average log-likelihood increases as K increases, for all cases except for the test data with a full-covariance matrix. First, it can be seen that for both types of matrices, GMM is more accurate on the train data than the test data, which is expected. Next, it is observed that training with a full-cov matrix is more accurate than training and testing with a diagonal-cov matrix. This would suggest that testing with a full-covariance matrix would also be more accurate for all values of K, but it only increases initially from $1 \leq K \leq 3$, and then drops lower and lower as K increases. This suggests that for large values of K, it safe to assume conditional independence among the variables for higher accuracy. This is an example of Occam's razor, where we should pick the simplest model that fits; either GMM with a full-covariance matrix and $K = 3$, or diagonal-covariance matrix and K $\approx$ 16.