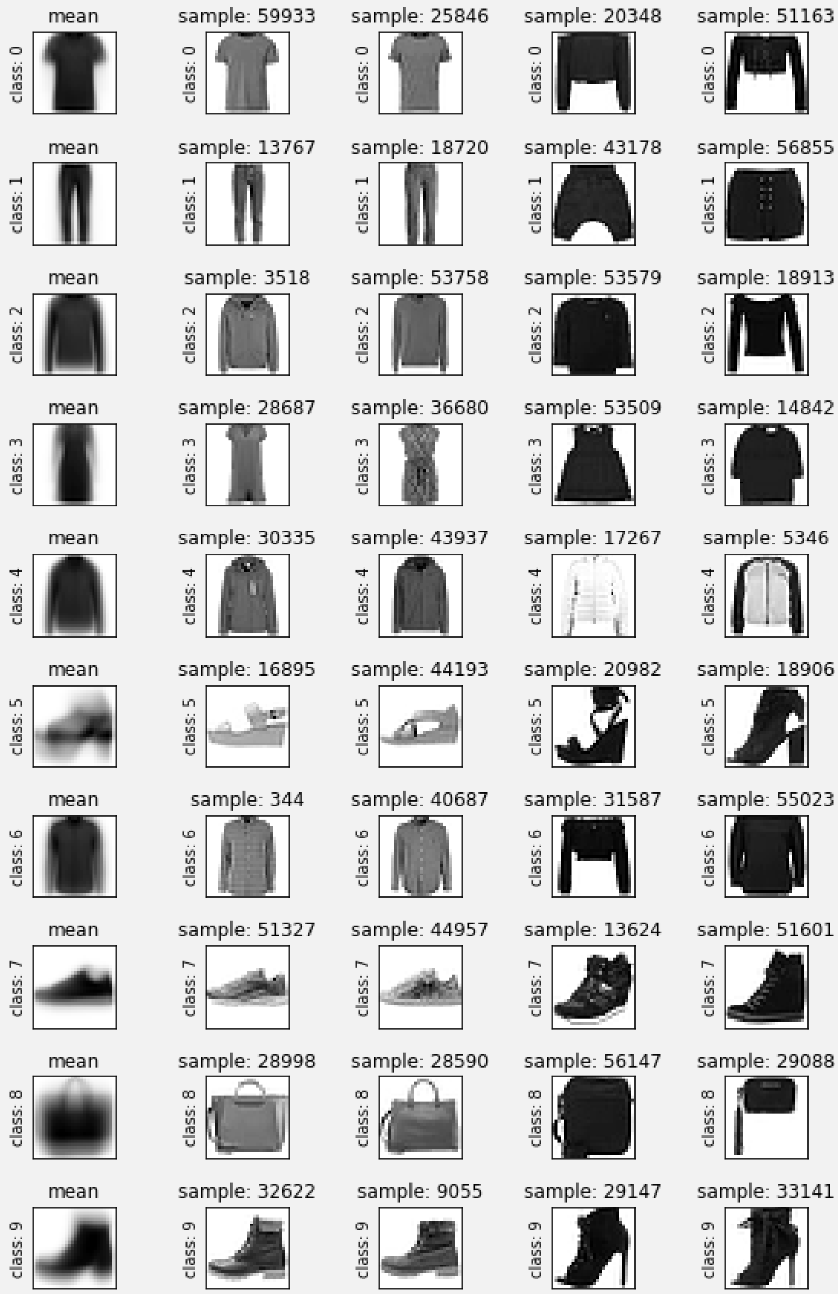
Question 1 : (30 total points) Image data analysis with PCA

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 first training sample in `Xtrn_nm` are: -3.137e-06, -2.268e-05, -1.180e-04, -4.071e-04. The first 4 elements for the last training sample in `Xtrn_nm` are identical to the first 4 elements of the first training sample: -3.137e-06, -2.268e-05, -1.180e-04, -4.071e-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.

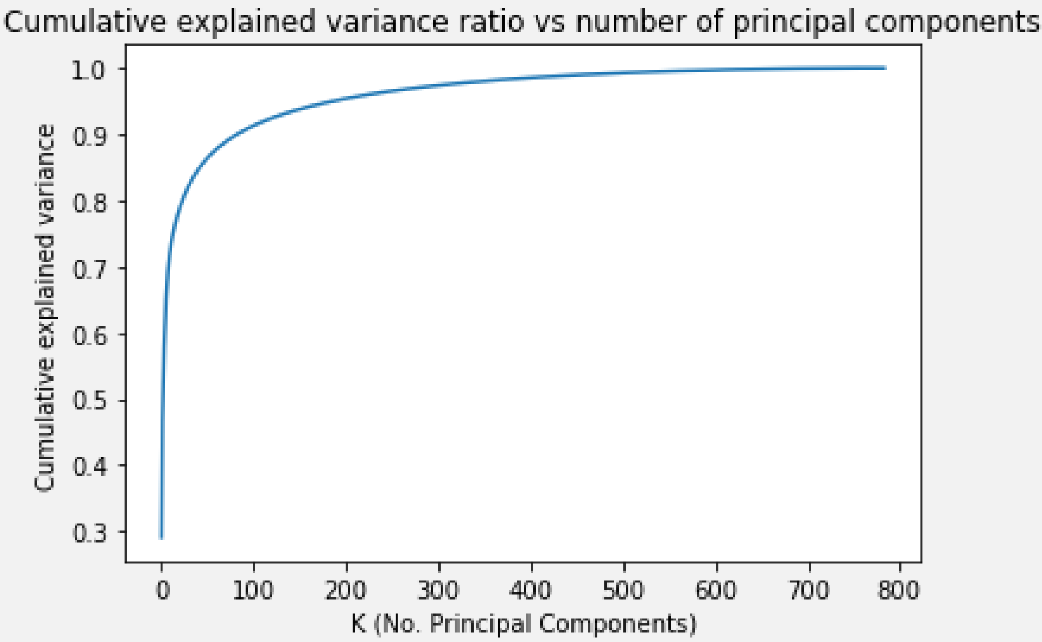


The two closest samples to the mean vector for each class, represent images with a very similar shape to the mean vector but with clear defined outlines/features, while the two furthest samples show images with quite a different shape and features to the mean but it is still clear they belong in the same 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`.

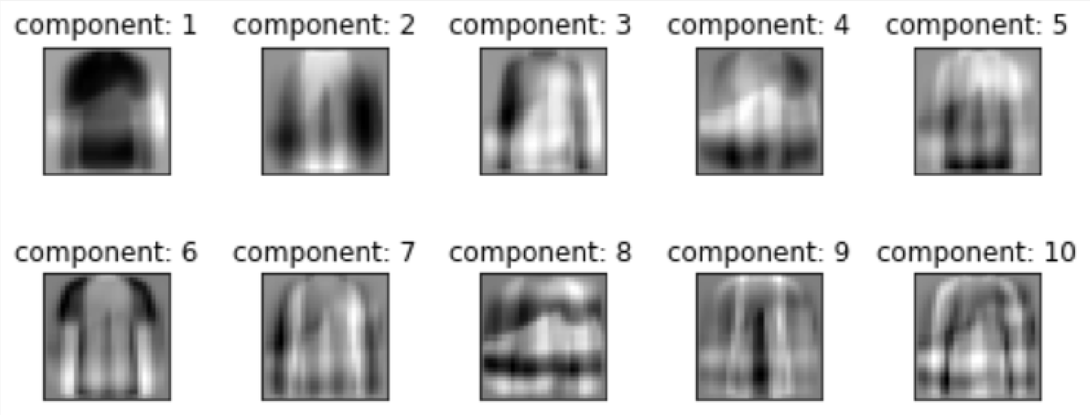
Principal component	Variance
1	19.810
2	12.112
3	4.106
4	3.382
5	2.626

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.



The graph above shows similar traits to logarithmic growth(i.e.steep gradient initially before bottoming out to almost flat). This means the first few components cover a large amount of the total explained variance, but the higher the component number the less variance it covers, which is what gives the graph this shape.

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.

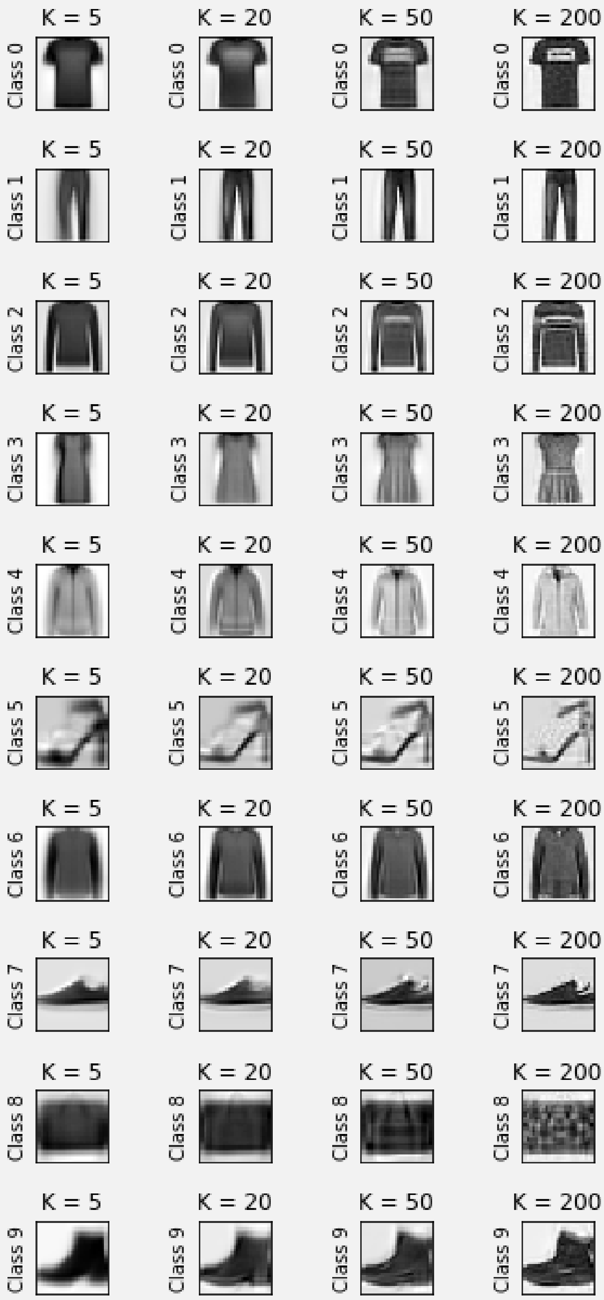


The first 2 components represent a large portion of the variance meaning they both show relatively clear images with distinguishable features. Then with each later component the images become filled with more noise as the components cover less of the variance and result in less defined features as they become more of a mix of all the different classes' features.

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.

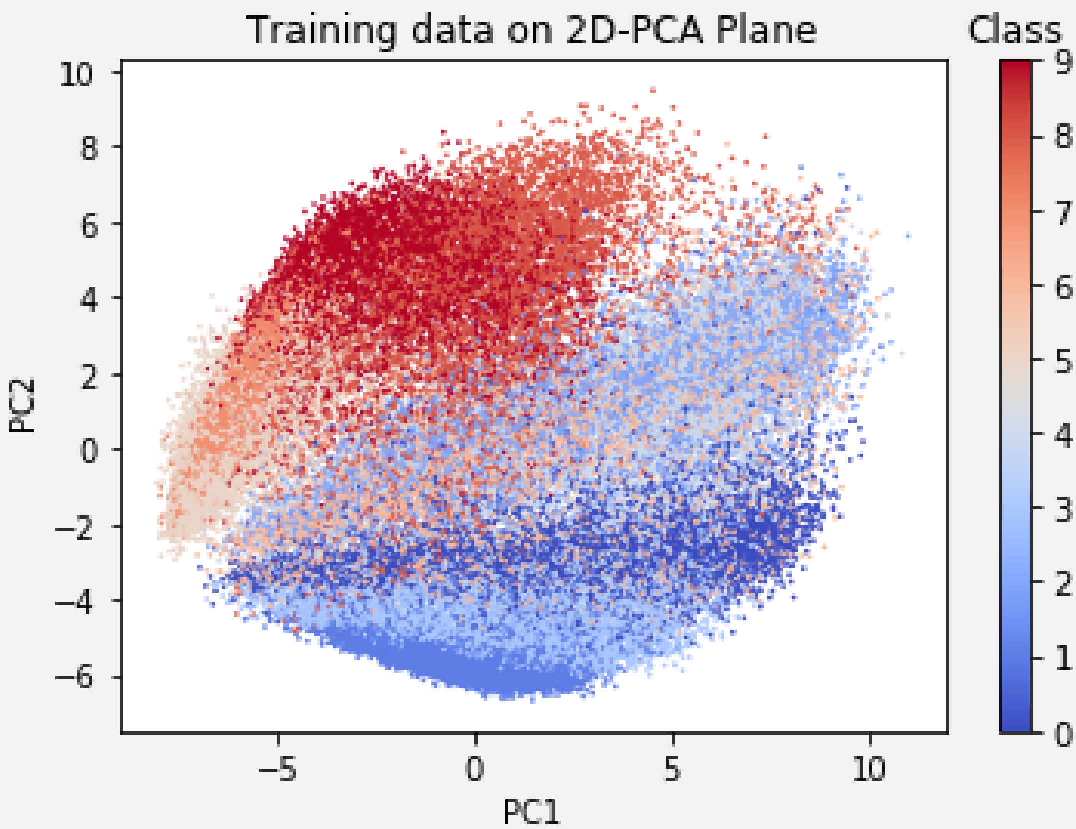
Class	K = 5	K = 20	K = 50	K = 200
0	0.159	0.124	0.090	0.039
1	0.135	0.072	0.036	0.015
2	0.145	0.130	0.106	0.064
3	0.138	0.097	0.076	0.043
4	0.112	0.098	0.065	0.031
5	0.151	0.125	0.113	0.076
6	0.120	0.077	0.063	0.035
7	0.117	0.095	0.066	0.026
8	0.146	0.129	0.116	0.081
9	0.184	0.117	0.090	0.050

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 results, it can be seen that with more components, each image is reconstructed with more details and sharper outlines. Although even with just 5 component, it is clear what the image was meant to be, and with more components, the improvement rate diminishes. This is because the first few components cover most of the variance, and each next component is worth less and less.

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 majority of the samples appear to be well separated and around points of the same class, although all the classes have a minor number of points which overlap with points in other classes and are not close to other points of the same class. Despite this it means that not too much information was lost when transforming the data into 2 dimensions, as the classes still have maintained relatively separated.

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 on the test was: 84.0%
The confusion matrix:

Predicted	0	1	2	3	4	5	6	7	8	9
Actual										
0	819	3	15	50	7	4	90	1	11	0
1	5	953	4	27	5	0	3	1	2	0
2	27	4	731	11	133	0	82	2	9	1
3	31	15	14	866	33	0	37	0	4	0
4	0	3	115	38	760	2	72	0	10	0
5	2	0	0	1	0	911	0	56	10	20
6	147	3	128	46	108	0	539	0	28	1
7	0	0	0	0	0	32	0	936	1	31
8	7	1	6	11	3	7	15	5	945	0
9	0	0	0	1	0	15	1	42	0	941

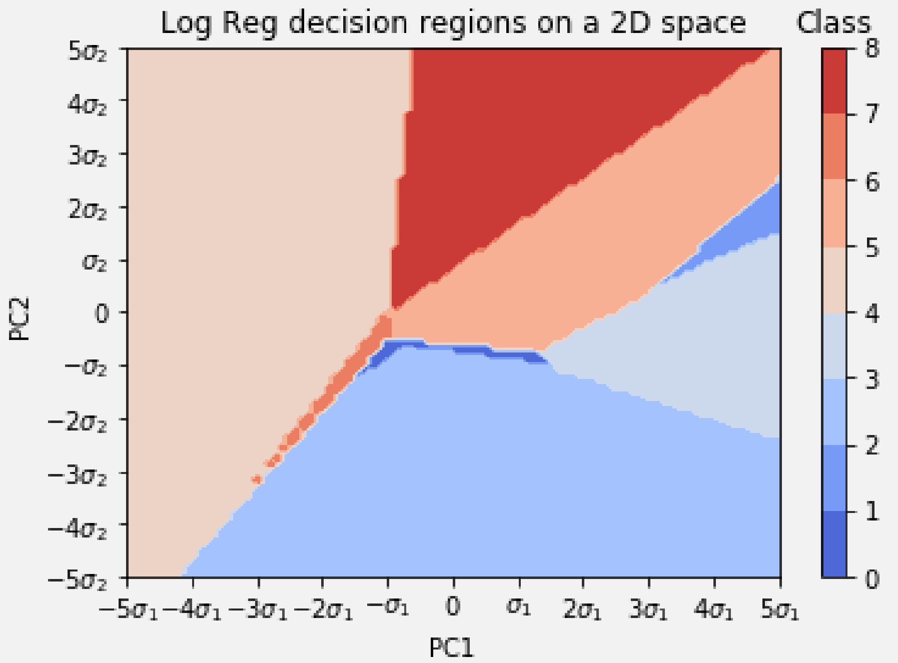
2.2 (3 points) Carry out a classification experiment with SVM classifiers, and report the mean accuracy and confusion matrix (in numbers) for the test set.

The classification accuracy on the test was: 84.6%.
The confusion matrix:

Predicted	0	1	2	3	4	5	6	7	8	9
Actual										
0	845	2	8	51	4	4	72	0	14	0
1	4	951	7	31	5	0	1	0	1	0
2	15	2	748	11	137	0	79	0	8	0
3	32	6	12	881	26	0	40	0	3	0
4	1	0	98	36	775	0	86	0	4	0
5	0	0	0	1	0	914	0	57	2	26
6	185	1	122	39	95	0	533	0	25	0
7	0	0	0	0	0	34	0	925	0	41
8	3	1	8	5	2	4	13	4	959	1
9	0	0	0	0	0	22	0	47	1	930

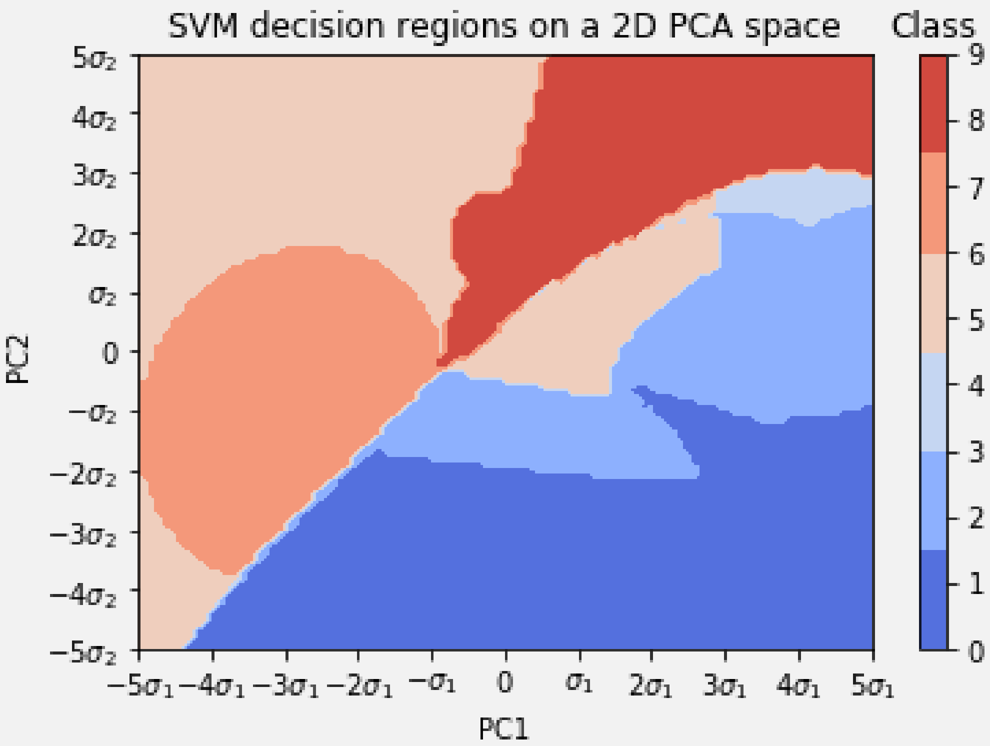
Question 2 continued on next page...

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



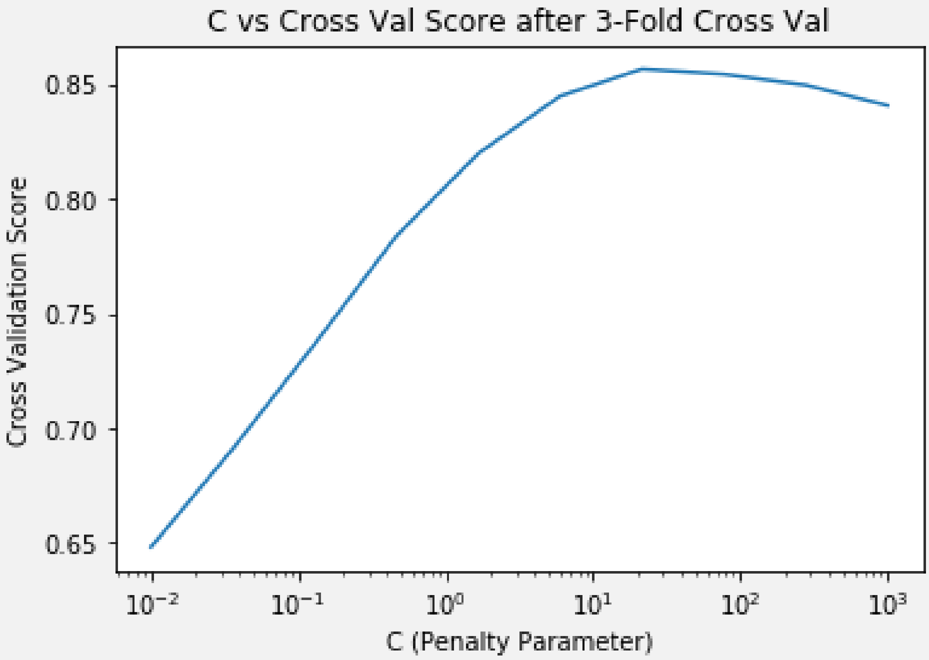
The main thing that can be noticed is that the logistic regression model in this 2D space never predicts any sample as class 9. Apart from that the decision regions appear to be well defined and separated from others, although they vary a lot in size.

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.



The SVM classifier predicts all classes, but there are only 6 distinct decision regions when projected onto the 2D space. This means the decision boundaries appear to overlap for the classes as there is not a clear region for each class. Even out of the 6 decision regions that are present, they are not very separated from others and appear to intrude on other regions.

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 score was 0.857 or 85.7% achieved with a value C of 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 for the training set was: 90.8%. The classification accuracy for the test set was: 87.6%.

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

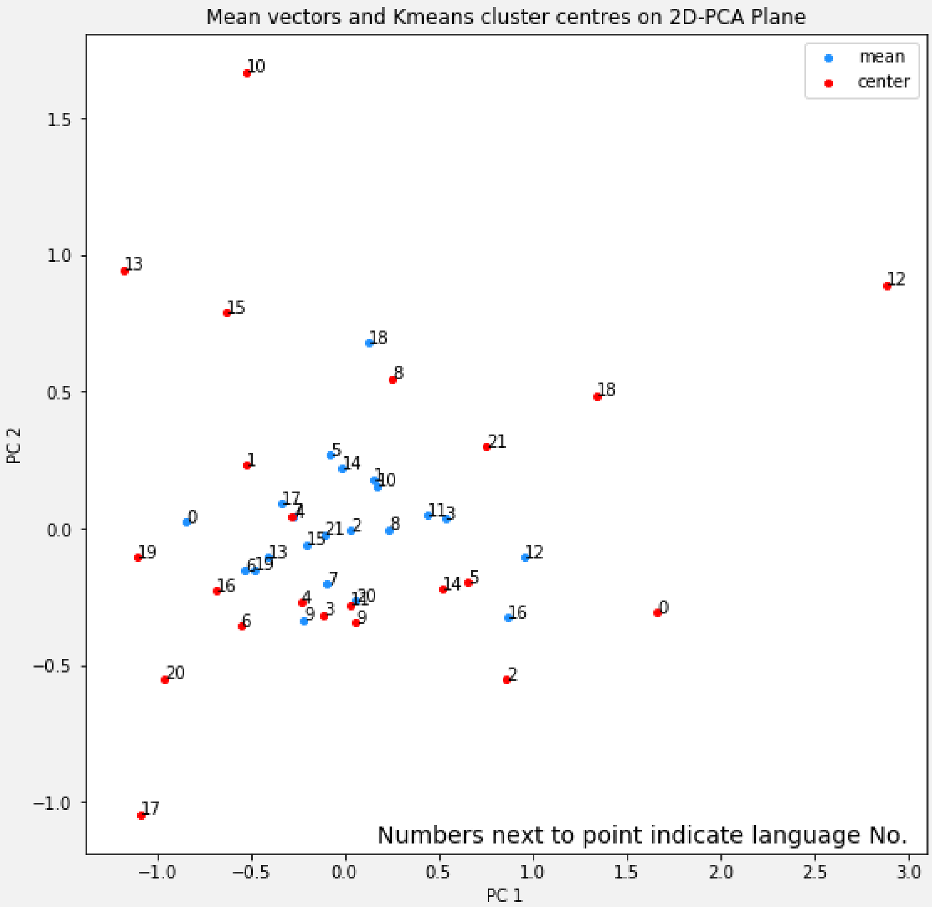
In this question we will explore K-means clustering, hierarchical clustering, and GMMs.

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 over the whole of `Xtrn` was 38185.817.

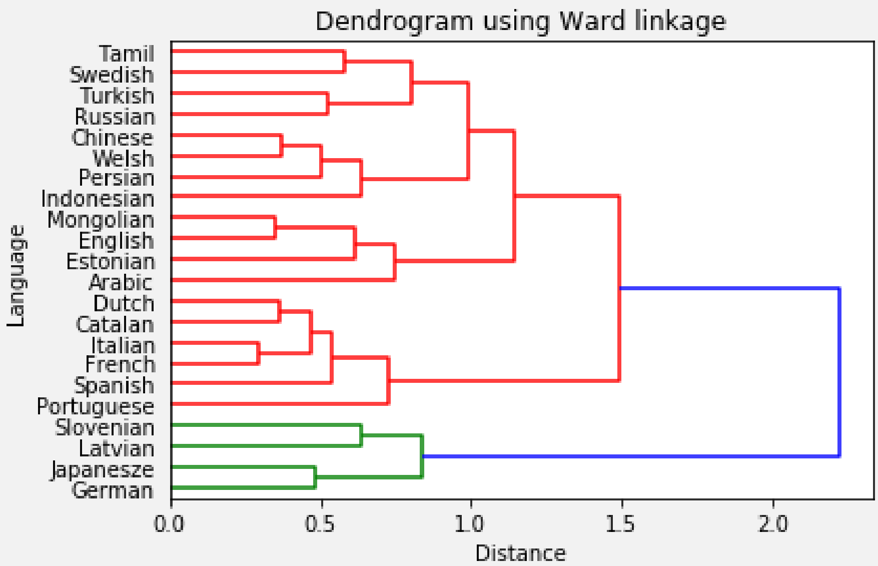
Cluster	Samples in Cluster
0	1018
1	1125
2	1191
3	890
4	1162
5	1332
6	839
7	623
8	1400
9	838
10	659
11	1276
12	121
13	152
14	950
15	1971
16	1251
17	845
18	896
19	930
20	1065
21	1466

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.



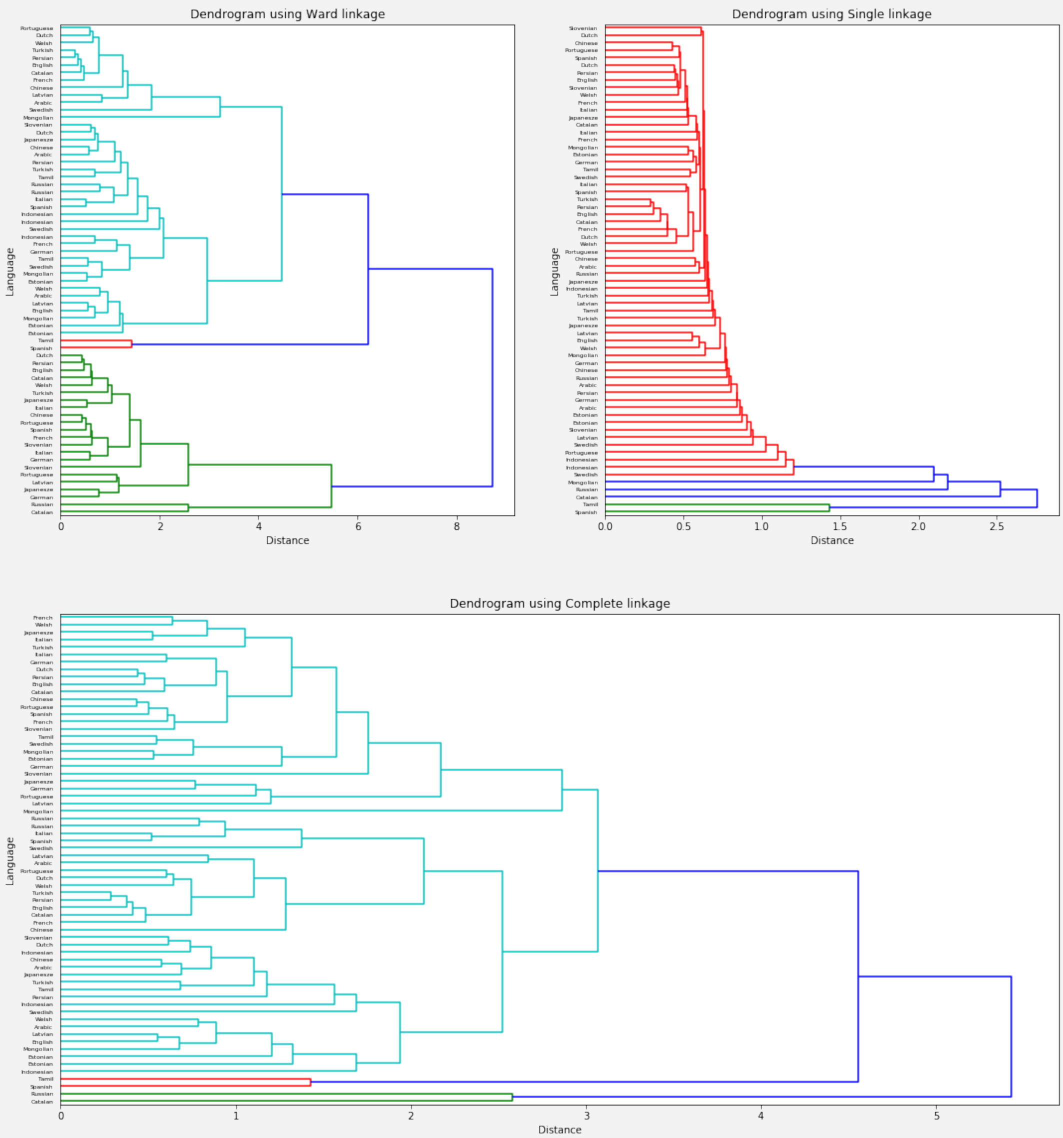
For most languages the mean vector is not similar to the cluster centre for that language. Also the spread of the cluster centres seems to be much higher while all the mean vectors are more densely distributed across a smaller area.

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.



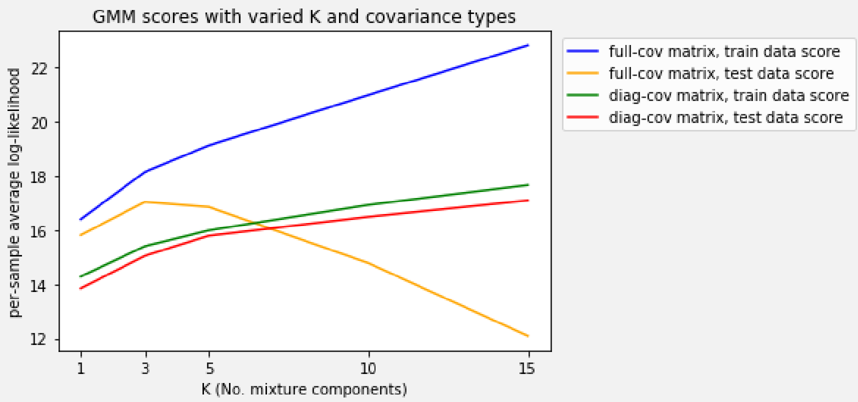
The dendrogram treats Slovenian, Latvian, Japanese and German as a separate group to all the other languages and only clusters those 4 with with the remaining languages at the end once all the others are clustered together. This suggest that these 4 languages do not have a similar structure to any of the other languages this is further proved by the relatively high linkage distance between the cluster of those 4 languages and the cluster of all the other languages.

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



The most obvious thing to note is that Single linkage results in the smallest distance between all the clusters, while Ward linkage puts the clusters as furthest away from each other, complete linkage is somewhere in between.A similarity is that all 3 linkage methods put Spanish and Tamil in a separate cluster far away from all other languages, implying that Spanish and Tamil are relatively similar in structure to each other, but very different to all the other languages. This also happens to Russian and Catalan with all linkage methods. So despite using different methods to find differences(distance) between languages, all methods used above still find most of same languages as being very different in structure to others.

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	Full,K = 1	Full,K = 3	Full,K = 5	Full,K = 10	Full,K = 15
Train	16.394	18.086	19.036	21.062	22.786
Test	15.811	17.066	16.489	14.622	11.848
Dataset	Diag,K = 1	Diag,K = 3	Diag,K = 5	Diag,K = 10	Diag,K = 15
Train	14.280	15.398	16.010	16.917	17.505
Test	13.843	15.041	15.909	16.568	16.902

Using a diagonal covariance matrix resulted in very similar GMM scores(per-sample avg. log likelihood) for both train and test data. However with the full covariance matrix on the test dataset, the score was only similar to the train dataset up to K = 3, but once K exceeded 3, it's score fell drastically. Suggesting that with the full covariance matrix, the GMM model likely overfitted to the training data.