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

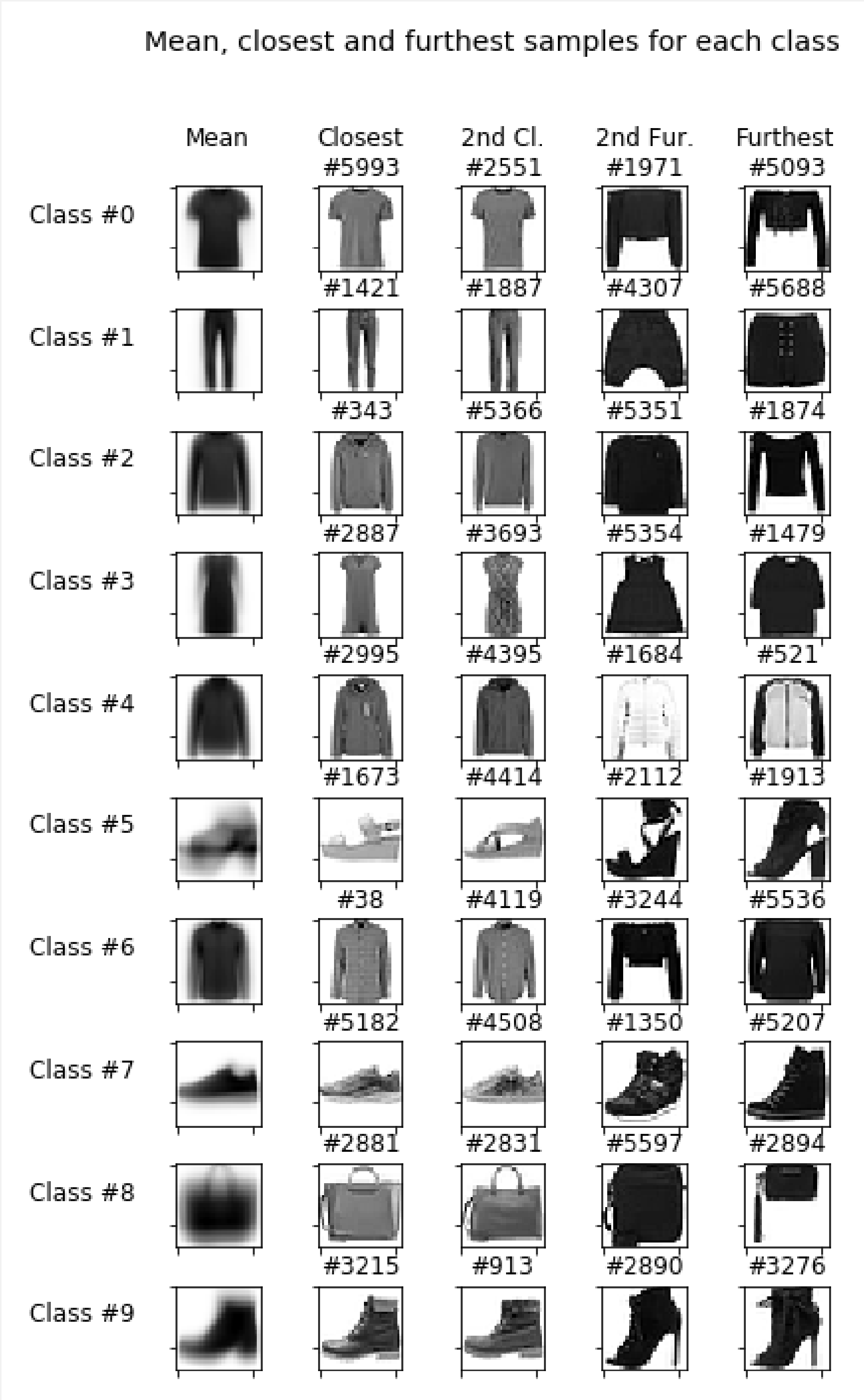
In this question we employ PCA to analyse image data

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,:]`.

$$X_{first} \approx (-3.137, -22.680, -117.974, -407.059, \ldots)^T \times 10^{-6}$$

$$X_{last} \approx (-3.137, -22.680, -117.974, -407.059, \ldots)^T \times 10^{-6}$$

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.

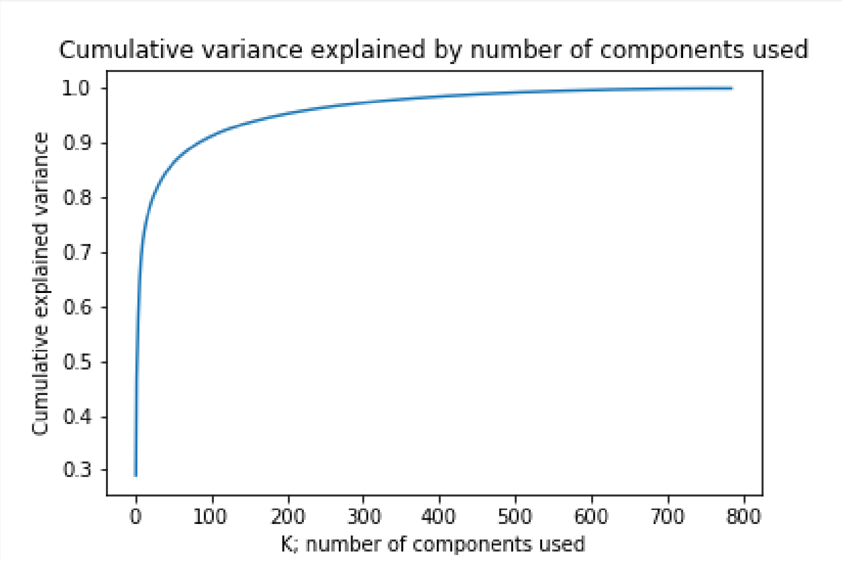


1.3 (3 points) Apply Principal Component Analysis (PCA) to the data of `Xtrn_nm` using sklearn.decomposition.PCA, and report the variances of projected data for the first five principal components in a table. Note that you should use `Xtrn_nm` instead of `Xtrn`.

PCA variances of projected data for the first 5 components:

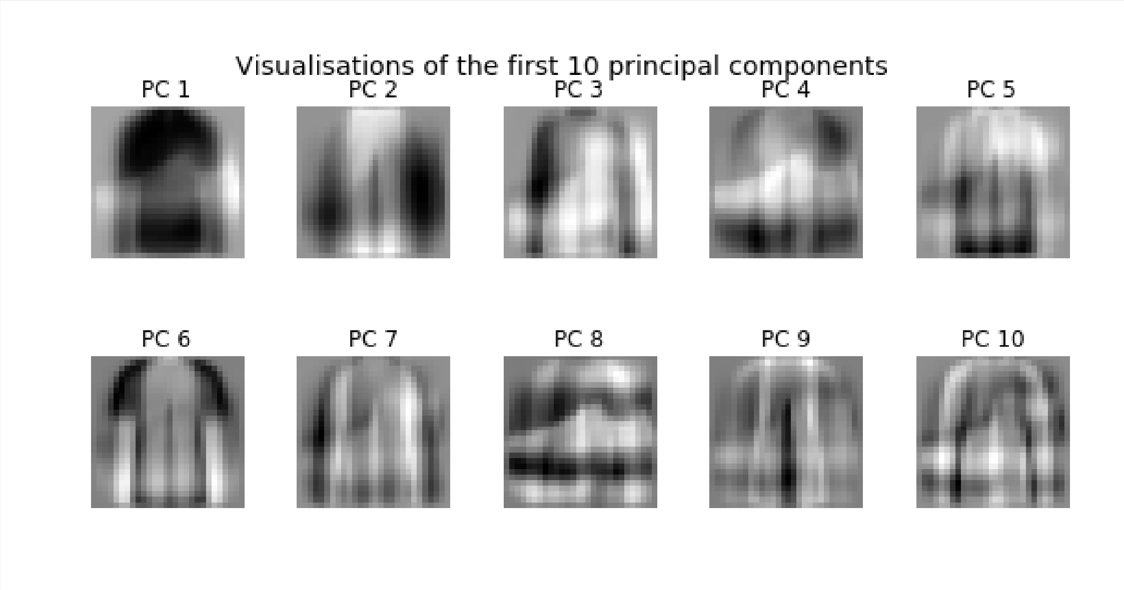
K	1	2	3	4	5
Var	19.81	12.11	4.11	3.38	2.62

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.



- We can see that the cumulative explained variance grows quickly in the $1 \leq K \leq 50$ interval. Following that, the function increases much more slowly.
- Therefore, as $K = 50$ PCA components explain $> 86\%$ of the variance, it looks like a good compromise between accuracy and complexity.
- It is often useful to plot decision regions on the plane spanned by the first 2 PCA components. They explain only $\sim 53\%$ of the total variance here, so one would need to be cautious when using them.

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.



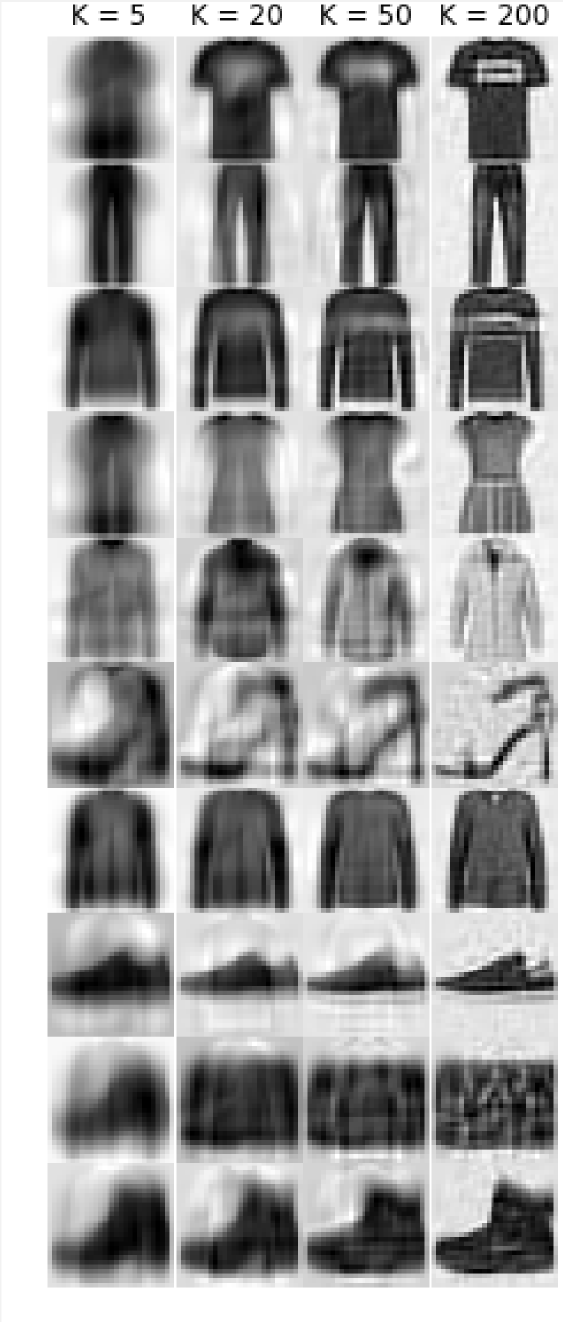
- We can see that the first component measures how much a picture resembles a shirt, as it has a shape of a t-shirt with gray areas in the shapes of sleeves. This makes sense as there are 4 classes with similar shapes (0: T-Shirt/Top, 2: Pullover, 4: Coat, 6: Shirt). These classes share a shape, so lots of variance can be explained by identifying it. Therefore, it is expected for the most important principal component to identify it (even though PCA is unsupervised).
- Out of the other principal components, we can similarily see that the 4-th and the 8-th one identify shoes. This makes sense, as there is a class with shoes and they have a unique shape.

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 original and reconstructed samples
for $K = 5, 20, 50, 200$:

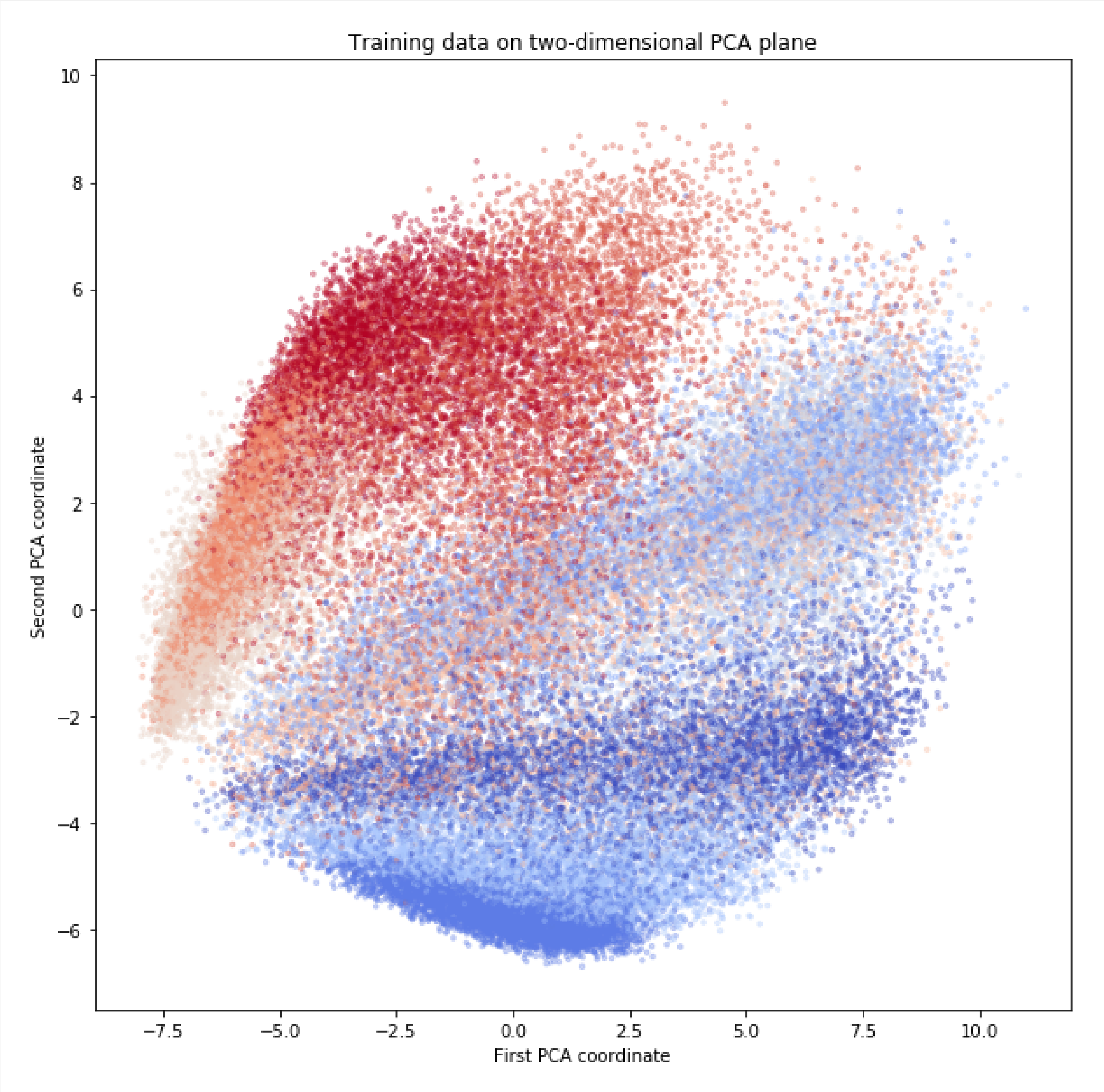
	K=5	K=20	K=50	K=200
Class #0	0.26	0.15	0.13	0.06
Class #1	0.20	0.14	0.10	0.04
Class #2	0.20	0.15	0.12	0.08
Class #3	0.15	0.11	0.08	0.06
Class #4	0.12	0.10	0.09	0.05
Class #5	0.18	0.16	0.14	0.09
Class #6	0.13	0.10	0.07	0.05
Class #7	0.17	0.13	0.11	0.06
Class #8	0.22	0.15	0.12	0.09
Class #9	0.18	0.15	0.12	0.07

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



We should expect more components to yield better reconstructions and that is visibly the case. The more components we use, the more variance they explain. Therefore, the images should also look more and more realistic (less blurred) and they do.

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.



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.

Accuracy: 84%

Confusion Matrix										
	1	2	3	4	5	6	7	8	9	10
1	819	3	15	50	7	4	89	1	12	0
2	5	953	4	27	5	0	3	1	2	0
3	27	4	731	11	133	0	82	2	9	1
4	31	15	14	866	33	0	37	0	4	0
5	0	3	115	38	760	2	72	0	10	0
6	2	0	0	1	0	911	0	56	10	20
7	147	3	128	46	108	0	539	0	28	1
8	0	0	0	0	0	32	0	936	1	31
9	7	1	6	11	3	7	15	5	945	0
10	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.

Your Answer Here

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.

Your Answer Here

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.

Your Answer Here

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`.

Your Answer Here

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.

Your Answer Here

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.

Your Answer Here

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.

Your Answer Here

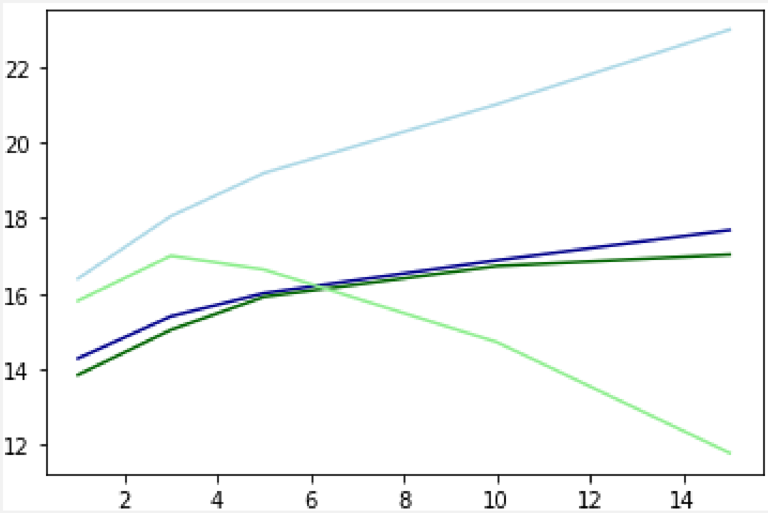
3.3 (3 points) We now apply hierarchical clustering on the training data set to see if there are any structures in the spoken languages.

Your Answer Here

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

Your Answer Here

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.,



	1	3	5	10	15
Diagonal/Training	14.28	15.40	16.01	16.88	17.68
Diagonal/Test	13.84	15.04	15.91	16.73	17.03
Full/Training	16.39	18.05	19.19	21.02	22.99
Full/Test	15.81	17.00	16.64	14.72	11.78

As expected, the most general model (full covariance matrix) fits the training data best. But, we can also see that this model begins to overtrain for $K > 3$. The less general model (diagonal covariance matrix) has a weaker fit, but doesn't overtrain for bigger Ks (test curve is increasing).