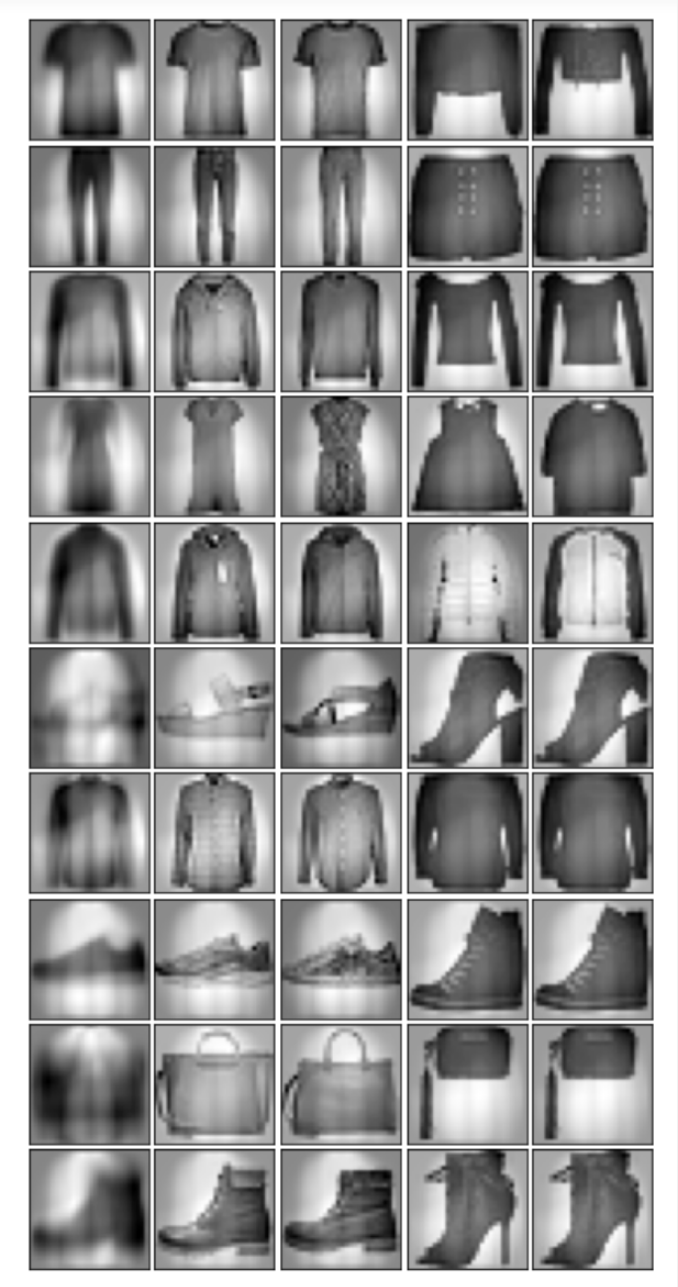
# Question 1 : (30 total points) Image data analysis with PCA

**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 four elements of the first training sample and the last training sample in normalised training data both have the following values: -3.13725490e-06, -2.26797386e-05 -1.17973856e-04, -4.07058824e-04. Their values are the same because the first four elements of the first training sample and those of the last training sample in original training data are all zero, and we subtracted the same values (Xmean) from both samples.

**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.



| | nearest | second nearest | second farthest | farthest |
|---|---|---|---|---|
| 0 | 59933 | 25846 | 20348 | 51163 |
| 1 | 13767 | 18720 | 56855 | 56855 |
| 2 | 3518 | 53758 | 18913 | 18913 |
| 3 | 28687 | 36680 | 53509 | 14842 |
| 4 | 30335 | 43937 | 17267 | 5346 |
| 5 | 16895 | 44193 | 18906 | 18906 |
| 6 | 344 | 40687 | 55023 | 55023 |
| 7 | 51327 | 44957 | 51601 | 51601 |
| 8 | 28998 | 28590 | 29088 | 29088 |
| 9 | 32622 | 9055 | 33141 | 33141 |

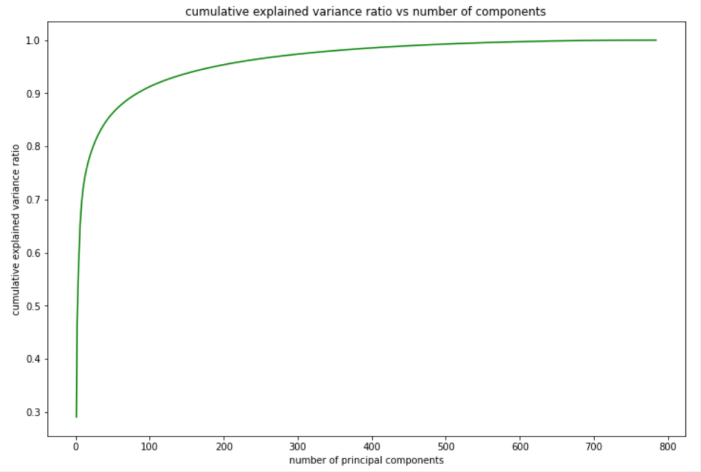
In the table, row numbers represent the class number and each cell represents the sample number of the corresponding image in the data set. It is difficult to clearly identify the object from the image of the mean vector but we rather see blurry silhouette. In contrast, we can identify the object from the images of the other four samples. The sample nearest to the mean vector is expected to resemble the image of the mean vector the most among other samples and the sample farthest from the mean vector to resemble the least, and this is apparently true.

**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`.

| | variance |
|---|---|
| 1 | 19.809806 |
| 2 | 12.112210 |
| 3 | 4.106157 |
| 4 | 3.381828 |
| 5 | 2.624770 |

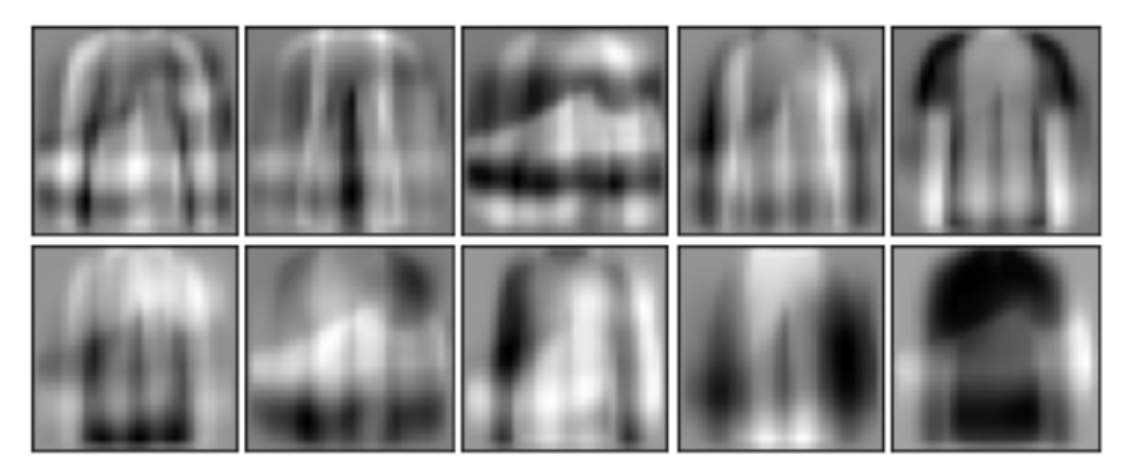
The row number represents which principal component is used to project the data (e.g. 3 means the third principal component), and each cell represents its explained variance (rounded up to six decimal places). Explained variance indicates how much information or variation of given data can be explained by the selected component. The first component accounts for variation of given data the most and the fifth component for the least. As we add more components, the model has a greater cumulative explained variance and therefore makes better predictions. However, if each component is considered individually, explained variance of i'th component is always less than that of (i-1)'th component. In other words, i'th component has the i'th highest explained variance among all components.

**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.




A crucial step before actually using principal component analysis (PCA) in practice is to estimate how many principal components are needed to appropriately describe the data. We can determine this by looking at the cumulative explained variance ratio as a function of the number of principal components because it allows us to find the point of diminishing returns where adding more component makes a very little improvement in accounting for given data. The function displays a concave down increasing graph, which means that more information of data is explained as more principal components are used but the rate itself is decreasing. For example, the cumulative explained variance ratio is already around 0.912 when we use the first hundred components but we need to add 684 principal components more in order to reach cumulative explained variance ratio of 1.

**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 principal component, which has the highest explained variance, outputs the clearest image compared to those of other components. As I already mentioned above, i-th component has the i-th highest explained variance. Therefore, the images in the table are sorted in descending order of explained variance. Hence, as we go from left to right and from up to down, the image gets more blurry and blurry and it gets more difficult to identify the object of the image. While we can quite confidently guess the image of the first component to be a shirts, the image of the last component is really difficult to identify: it has silhouette of a bag, sneakers and shirts simultaneously.

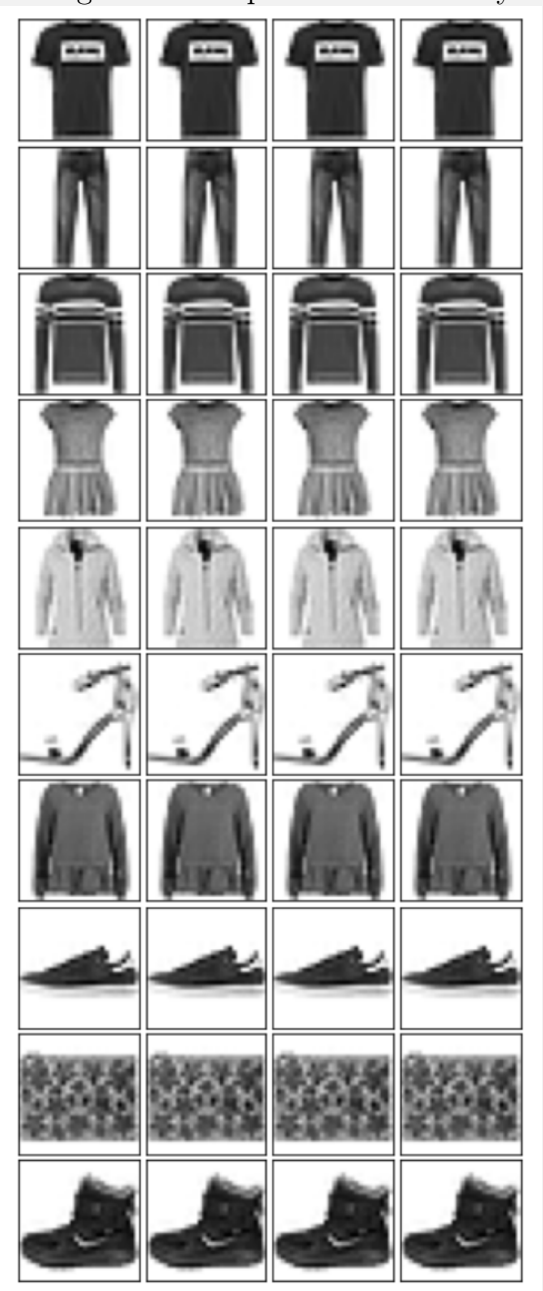
**1.6** (5 points) Using `Xtrn_nm`, for each class and for each number of principal components $K = 5, 20, 50, 200$, apply dimensionality reduction with PCA to the first sample in the class, reconstruct the sample from the dimensionality-reduced sample, and report the Root Mean Square Error (RMSE) between the original sample in `Xtrn_nm` and reconstructed one.

| | 5 | 20 | 50 | 200 |
|---|---|---|---|---|
| 0 | 0.256149 | 0.150013 | 0.127593 | 0.061699 |
| 1 | 0.198024 | 0.140437 | 0.095265 | 0.035855 |
| 2 | 0.198700 | 0.145634 | 0.123722 | 0.080728 |
| 3 | 0.145658 | 0.107512 | 0.083591 | 0.057013 |
| 4 | 0.118209 | 0.102628 | 0.087780 | 0.046034 |
| 5 | 0.181130 | 0.158720 | 0.142822 | 0.091222 |
| 6 | 0.129479 | 0.095887 | 0.071986 | 0.046070 |
| 7 | 0.165625 | 0.127794 | 0.106480 | 0.062685 |
| 8 | 0.223397 | 0.144979 | 0.124203 | 0.091325 |
| 9 | 0.183510 | 0.151106 | 0.121946 | 0.071451 |

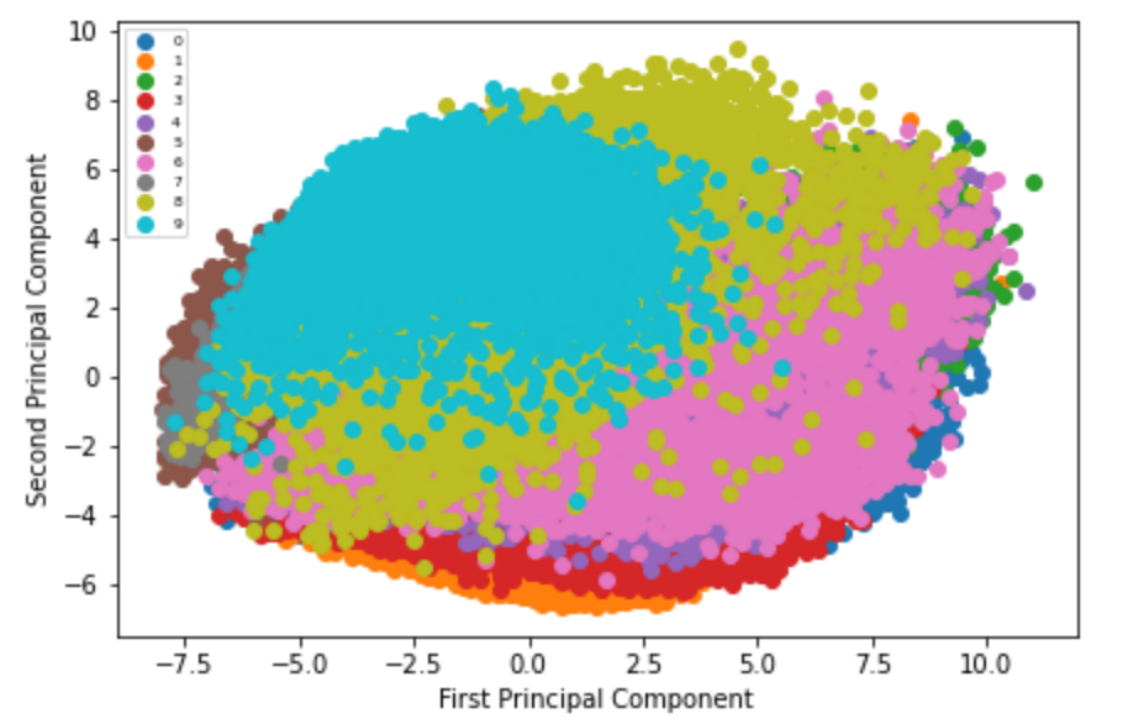
The mean squared error or root mean squared error between the original data and the predicted data should decrease as the number of principal components applied to project the predicted data increases. In fact, this is apparent in the table.

**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 have a clear picture of each class's image. The important thing to note is that using the first five principal components is actually sufficient to identify the object of every class. In fact, it is very difficult to detect any difference between the images to which the first 5 components are applied and the first 200 components are applied. The latter should be slightly better in terms of explaining given data, but the former is indeed clear enough. This interesting property of PCA can be attributed to the fact that there is always a point of diminishing returns where adding more component makes very little improvement in accounting for data.



**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 plot shows that all 10 classes are overlapped by each other enormously. Hence, it is very difficult to separate a class from another for all classes, and this means that it can be very difficult to classify the data if we reduce the feature dimensionality from 784 to 2, especially when dealing with such large data set.

# 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.

```
[[813   2  12  50   5   2 105   0  10   1]
 [  4 956   4  26   5   0   4   0   1   0]
 [ 25   4 736  11 120   1  90   1  12   0]
 [ 28  14  19 865  25   1  42   0   6   0]
 [  0   1 112  36 755   1  88   0   7   0]
 [  0   0   0   1   0 921   0  51   5  22]
 [143   3 126  38 103   0 561   0  26   0]
 [  0   0   0   0   0  28   0 944   0  28]
 [  6   2   7  11   2   5  20   6 940   1]
 [  0   0   0   0   0  15   1  40   1 943]]
```

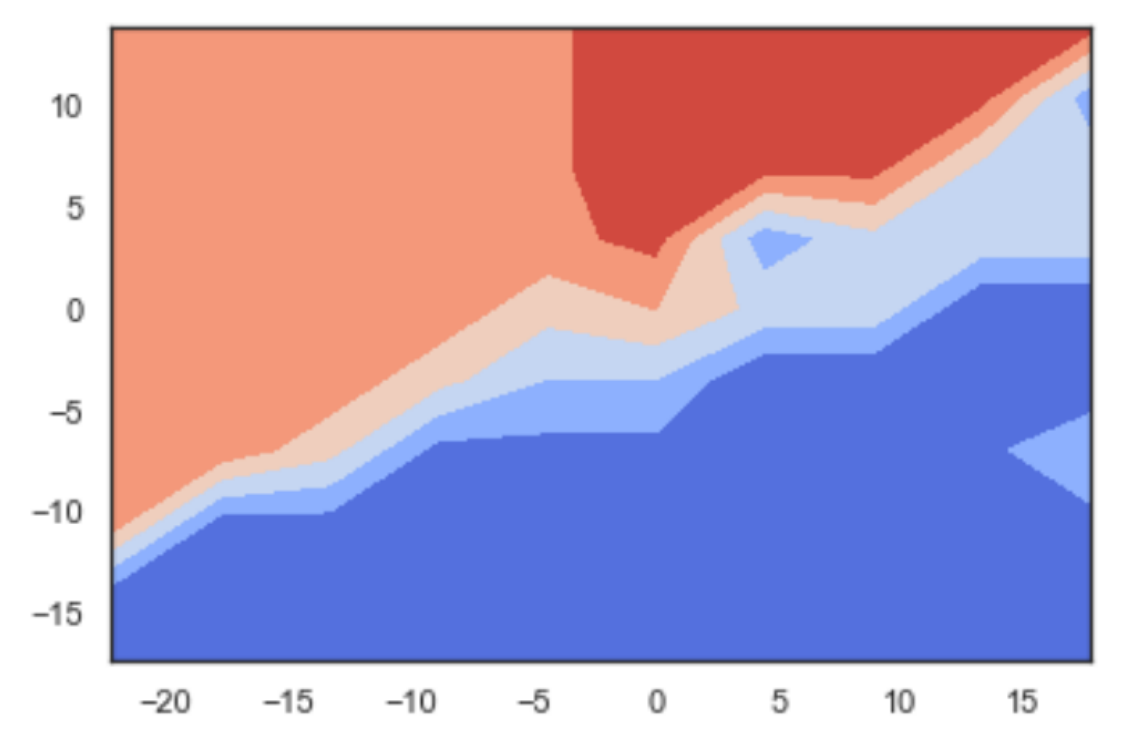
The classification accuracy is 0.8434.

**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.

```
[[863   0  13  26   3   2  85   0   8   0]
 [  4 961   2  26   3   0   4   0   0   0]
 [ 13   1 822  14  87   0  63   0   0   0]
 [ 25   3  13 893  32   0  30   0   4   0]
 [  1   0  83  29 822   0  63   0   2   0]
 [  0   0   0   1   0 960   0  27   1  11]
 [132   1  98  29  60   0 667   0  13   0]
 [  0   0   0   0   0  19   0 960   0  21]
 [  3   1   1   5   2   2   4   4 978   0]
 [  0   0   0   0   0  10   1  36   0 953]]
```

The mean accuracy is 0.8879.

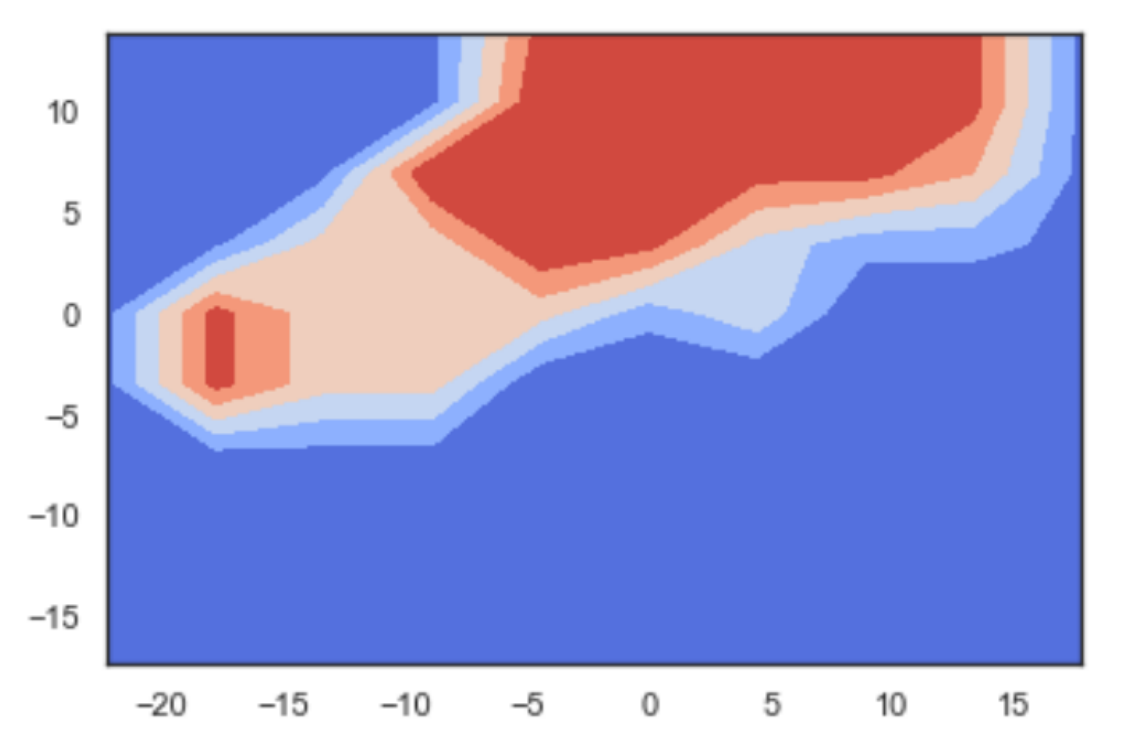
**2.3** (6 points) We now want to visualise the decision regions for the logistic regression classifier we trained in Question 2.1.




As we have expected, it is almost impossible to separate a class from another because a lot of their data are overlapped by each other. Therefore, the decision regions are also overlapped by each other, and hence we cannot see the decision regions of some classes. Despite being already taken up by some class, regions are continuously overlapped by another class.

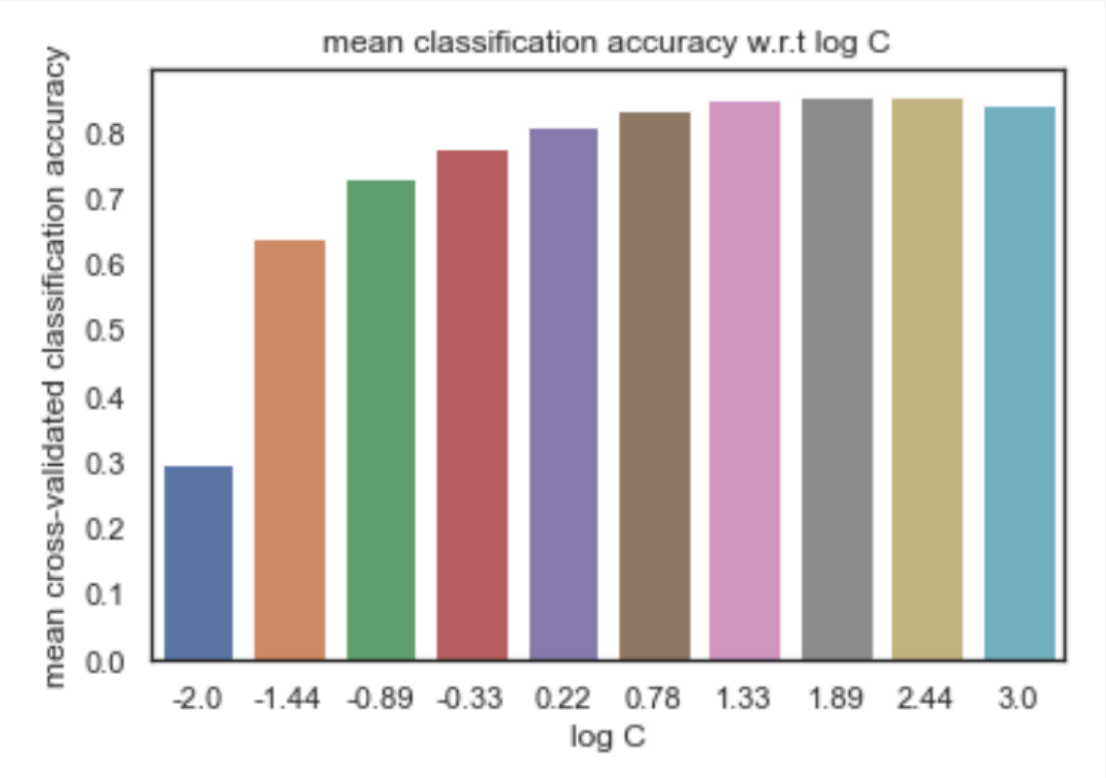
*Question 2 continued on next page...*

**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.




In this SVM model, it is also almost impossible to separate a class from another because a lot of their data are overlapped by each other. Therefore, just like for the logistic regression model, the decision regions are also overlapped by each other, and hence we cannot see the decision regions of some classes. The difference of this plot from that of logistic regression model is that the regions already taken up by some class are not overlapped by another class.

**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 classification accuracy of the SVM classifier is approximately 0.853285. The highest obtained mean accuracy score is 0.852256. The optimal value of C is around 77.426.

**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 on the training set and test set is 0.93273 and 0.8802, respectively.

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

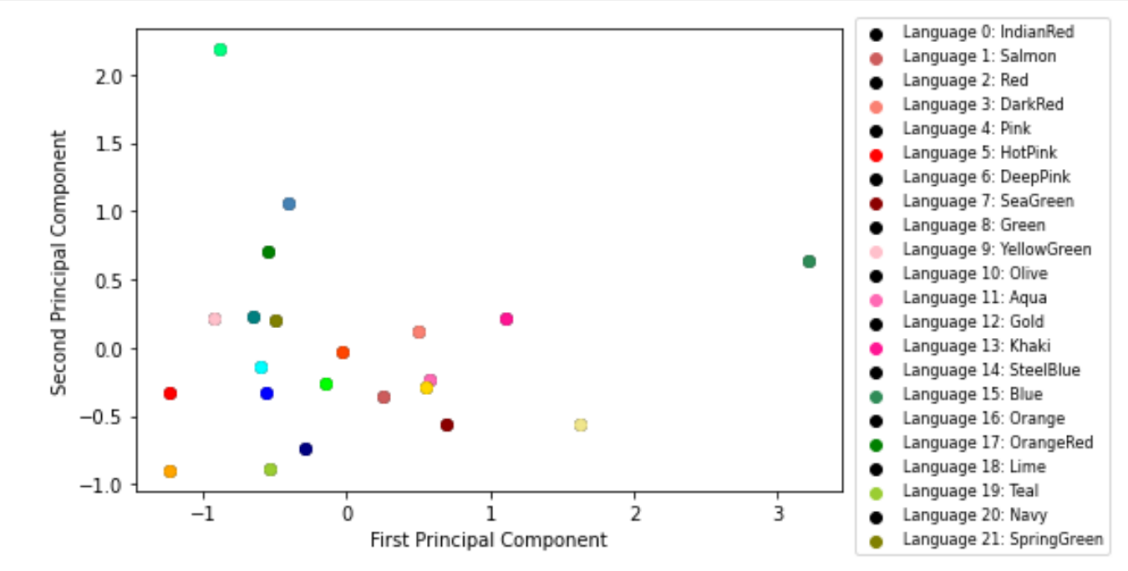
**In this question we will explore K-means clustering, hierarchical clustering, and GMMs.**

**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: 38150.9296875
Number of samples for cluster 1: 1060
Number of samples for cluster 2: 975
Number of samples for cluster 3: 936
Number of samples for cluster 4: 1281
Number of samples for cluster 5: 900
Number of samples for cluster 6: 1351
Number of samples for cluster 7: 1522
Number of samples for cluster 8: 152
Number of samples for cluster 9: 1610
Number of samples for cluster 10: 902
Number of samples for cluster 11: 1167
Number of samples for cluster 12: 1440
Number of samples for cluster 13: 912
Number of samples for cluster 14: 1231
Number of samples for cluster 15: 584
Number of samples for cluster 16: 748
Number of samples for cluster 17: 753
Number of samples for cluster 18: 1655
Number of samples for cluster 19: 859
Number of samples for cluster 20: 828
Number of samples for cluster 21: 990
Number of samples for cluster 22: 144
```

*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.




Various colours represent the mean vector of each language and black represents the cluster centres. It is difficult to see a black data point unless we zoom in significantly because all 22 data points are almost overlapped by the mean vectors. Hence, the mean vectors and cluster centres are enormously similar.

**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.