Image Classification and Clustering

M.Sc. in Data Science

Practical Data Science

Full Name: Eirini Mylona

Part 1

In this part was implemented the preprocessing of the data, created the appropriate dataframes and visualized some images of these data frames. We have at out disposal 2 datasets which contain images in grayscale with dimensions 28 x 28 pixels. After visualizing some images, which depicts Greek letters, we can easily understand that some of them cannot clearly defined even from a person, due to the fact that the pixels are not clear in some cases. Thus, it is expected that some algorithms may do not work satisfactory for the classification problem of the images.

Part 2

In this part, was implemented the visualization of the frequency per character per dataset and is inferred that the vowels ' α ',' ο ',' ε ',' ι ' was the most frequent characters both for the two datasets. The less used characters seem to be some consonants such as ' ξ ', ' φ ' etc. Also, it is represented the visualization of the frequency per TM and century.

Part 3

In this part, both for the two questions, was trained the above machine learning algorithms: K-NN, logistic regression, SVM algorithm with different kind of kernels, Decision Trees, Random Forests and Extremely randomized forests. Only the f1 score is taken into account due to the fact that the datasets in each case are unbalanced. Rare character and TMs have been dropped.

For the task of classifying the character depicted in the image, the higher f1 score for the specific split of the dataset, is given by the SVM radial algorithm with f1 score **53.26%**. Although, it is obvious that it is not recommended to be used for the specific classification problem, due to the fact that the percentage is low.

To similar conclusions we reach for the task of classifying the TM for the first dataset. More specifically, the higher f1 score for the specific split of the dataset, is given by the SVM radial algorithm (43.89%) with tunning (Grid Search).

For the task of classifying the century, it is observed that all the algorithms gave us better results (higher f1 scores) in comparison with the other two tasks of classification. In this case, it is observed that except the SVM radial algorithm (Grid Search) with f1 score **78.62%**, the extremely randomized trees worked slightly better with f1 score **80.27%**. Due to the fact that the number of classes we have at our disposal is small, decision trees seem to work better than in the other two tasks of classification.

Part 4

In this part it is performed the KMeans clustering of the images and is selected the optimum k using Normalized Mutual Information criterion for 2 instances: (a) where as ground truth is used the character, the optimal selection of k was 27 and (b) where as ground truth is used the century, the optimal selection of k was 7.

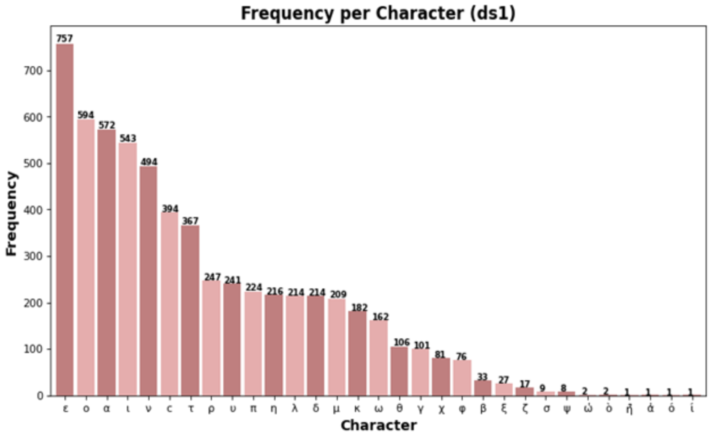
Part 5

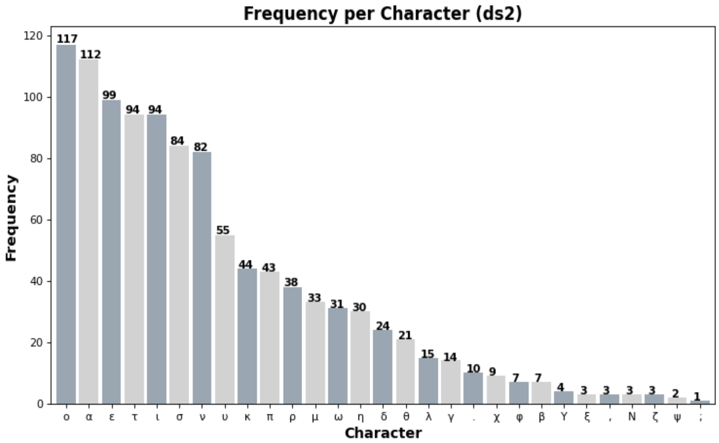
In this part, the clustering of the images was implemented by using the Davies Bouldin criterion. In this case, the optimum K per character was selected to be 31.

Note: The above was a briefly review of the results. It follows, informative description of the findings (figures and results are represented).

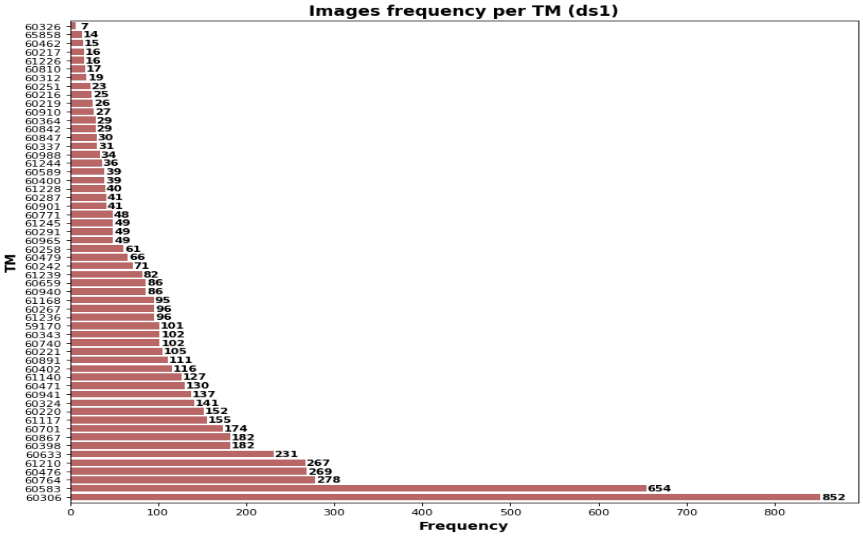
Part 2

In part 2, is visualized the character frequency per dataset, the TM frequency of the first dataset and the century frequency of the second one.



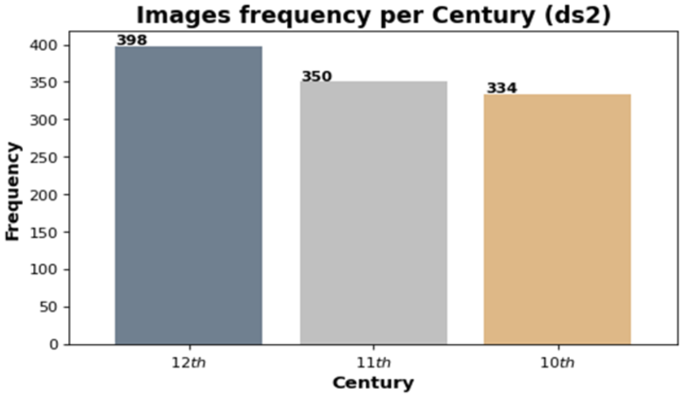


Compared the two datasets, we can easily understand that the most frequent used characters in the specific period of time seems to be the vowels ' α ',' ο ',' ε ',' ι ' and the less used characters seems to be some consonants such as ' ξ ', ' φ ' etc.



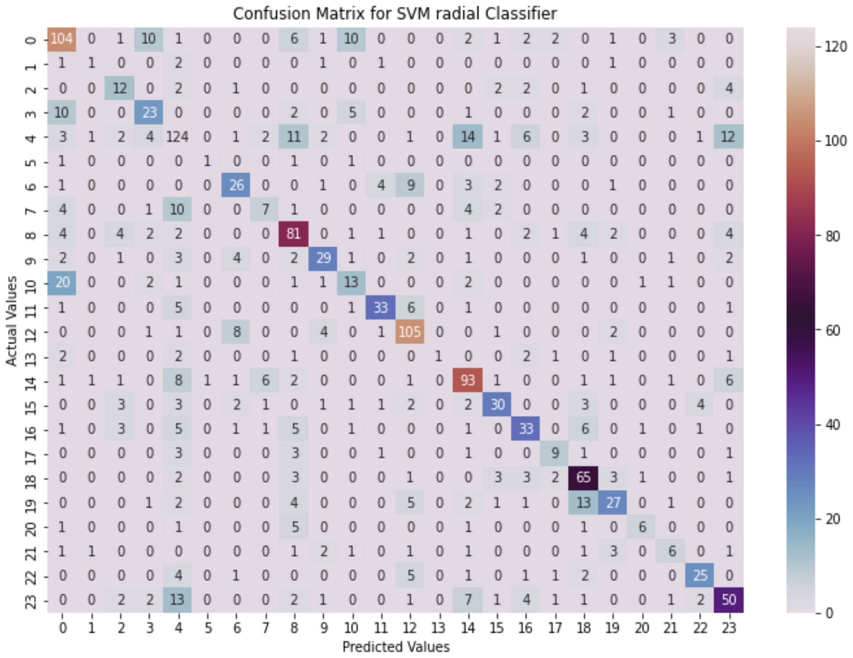
From the plot, it is obviously that most of the images contained in ds1 are contained in the TM with number 60306 (852). It follows the TM 60583 with a sum of 654 images. The TM with the less contained images seems to be the 60326 with a sum of 7 images.

From the plot, we can easily understand that the most images of the dataset 2 (ds2_new) are observed to belong to the 12th century. Although, it seems that the dataset is balanced in respect to century, since all the centuries differ slightly in their frequency number of images.



Part 3

In this part, both for the two questions, was trained the above machine learning algorithms: K-NN, logistic regression, SVM algorithm with different kind of kernels, Decision Trees, Random Forests and Extremely random forests. It is vital to be mentioned the fact that for each algorithm are represented the accuracy score and the f1 score. Given the fact that the data is unbalanced, only the f1 score is considered. Furthermore, we have dropped the rare characters and TMs in order the training to be more accurate.



Confusion Matrix for SVM radial Classifier

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	104	0	1	10	1	0	0	0	6	1	10	0	0	0	2	1	2	2	0	1	0	3	0	0
1	1	1	0	0	2	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0
2	0	0	12	0	2	0	1	0	0	0	0	0	0	0	0	2	2	0	1	0	0	0	0	4
3	10	0	0	23	0	0	0	0	2	0	5	0	0	0	1	0	0	0	2	0	0	1	0	0
4	3	1	2	4	124	0	1	2	11	2	0	0	1	0	14	1	6	0	3	0	0	0	1	12
5	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	26	0	0	1	0	4	9	0	3	2	0	0	0	1	0	0	0	0
7	4	0	0	1	10	0	0	7	1	0	0	0	0	0	4	2	0	0	0	0	0	0	0	0
8	4	0	4	2	2	0	0	0	81	0	1	1	0	0	1	0	2	1	4	2	0	0	0	4
9	2	0	1	0	3	0	4	0	2	29	1	0	2	0	1	0	0	0	1	0	0	1	0	2
10	20	0	0	2	1	0	0	0	1	1	13	0	0	0	2	0	0	0	0	0	1	1	0	0
11	1	0	0	0	5	0	0	0	0	0	1	33	6	0	1	0	0	0	0	0	0	0	0	1
12	0	0	0	1	1	0	8	0	0	4	0	1	105	0	0	1	0	0	0	2	0	0	0	0
13	2	0	0	0	2	0	0	0	1	0	0	0	0	1	0	0	2	1	0	1	0	0	0	1
14	1	1	1	0	8	1	1	6	2	0	0	0	1	0	93	1	0	0	1	1	0	1	0	6
15	0	0	3	0	3	0	2	1	0	1	1	1	2	0	2	30	0	0	3	0	0	0	4	0
16	1	0	3	0	5	0	1	1	5	0	1	0	0	0	1	0	33	0	6	0	1	0	1	0
17	0	0	0	0	3	0	0	0	3	0	0	1	0	0	1	0	0	9	1	0	0	0	0	1
18	0	0	0	0	2	0	0	0	3	0	0	0	1	0	0	3	3	2	65	3	1	0	0	1
19	0	0	0	1	2	0	0	0	4	0	0	0	5	0	2	1	1	0	13	27	0	1	0	0
20	1	0	0	0	1	0	0	0	5	0	0	0	0	0	1	0	0	0	1	0	6	0	0	0
21	1	1	0	0	0	0	0	0	1	2	1	0	1	0	1	0	0	0	1	3	0	6	0	1
22	0	0	0	0	4	0	1	0	0	0	0	0	5	0	1	0	1	1	2	0	0	0	25	0
23	0	0	2	2	13	0	0	0	2	1	0	0	1	0	7	1	4	1	1	0	0	1	2	50

For the task of classifying the character depicted in the image, the higher f1 score for the specific split of the dataset, is given by the SVM radial algorithm (0.5326, **53.26%**) with tunning (Grid Search). Although, nobody can dispute the fact that the score is low and thus the algorithm will confuse some classes with others. Also, is represented the confusion matrix of it. We observe that the matrix tends to be diagonal but classifies wrongly some instances. (a lot of instances have wrongly classified to other classes). However, the algorithm seems to confuse some classes more than others.

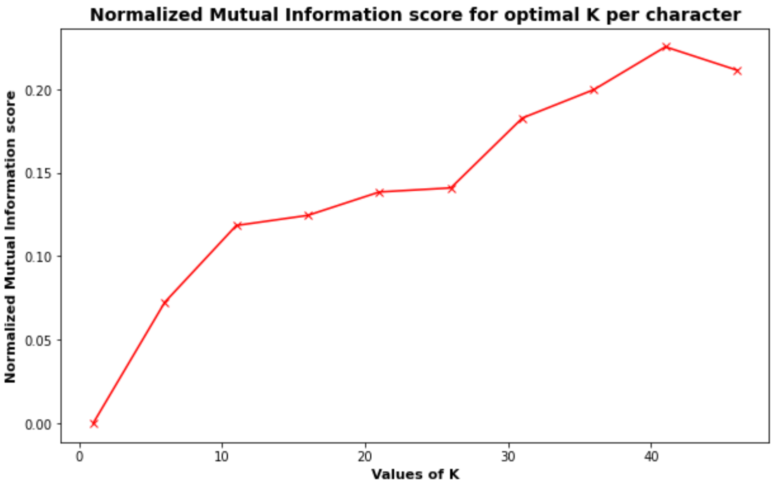
To similar conclusions we reach for the task of classifying the TM for the first dataset. More specifically, the higher f1 score for the specific split of the dataset, is given by the SVM radial algorithm (0.4389, **43.89%**) with tunning (Grid Search). From the classification report (represented in notebook), it is obvious, that some classes are confused with some other classes. Also, it is observed that the classes which have a large number of sample (support) has also higher f1 score which means that with larger sample the algorithm is more probable to work correctly. This fact seems logical, since when we have at our disposal a large number of sample the algorithm is trained better.

For the task of classifying the century, it is observed that all the algorithms gave us better results (higher f1 scores) than the other two tasks of classification. This fact is logical, since in this case we have only 3 classes and thus, the classification would be more 'accurate'. In this case, it is observed that except the SVM radial algorithm (Grid Search) with f1 score **78.62%**, the extremely randomized trees worked slightly higher with f1 score **80.27%.** Since the number of classes, we have at our disposal is small, decision trees seem to work better than in the other two tasks of classification.

Part 4

- **NMI, as ground truth is used the character.**

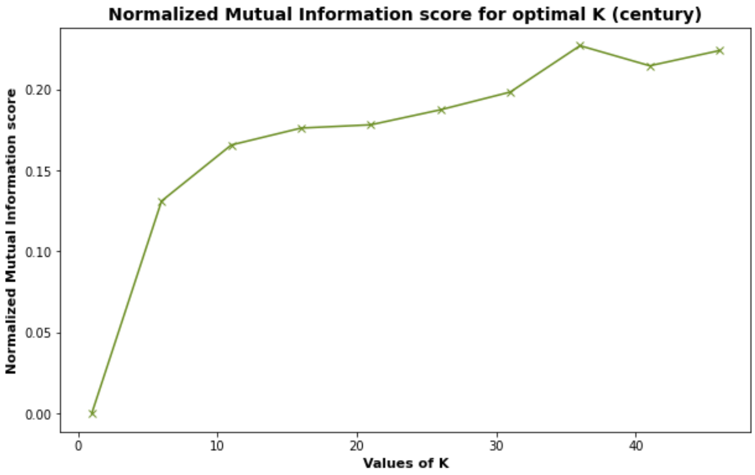
As we already know, the NMI score fluctuates between [0,1]. Thus, the preferred number of clusters by using KMeans algorithm and by checking the NMI score is this, whose value converges to 1. From results provided in the notebook, we can understand that as the number of clusters increases, the NMI score increases too. This fact, seems to be logical due to the fact that as the number of clusters increases the clustering will be more 'accurate' due to the fact that each instance would have each cluster (when n converges to the number of objects contained in sample) and thus no clustering would be considered.



It has been selected a range between 10 and 45 due to the fact that we know that we have at our disposal 18 characters of the specific data frame, and we want to make the prediction more accurate either by using a smaller number of clusters or a bigger which make sense though. Given the fact that the NMI score is too low for the different number of clusters we can claim that we want many clusters in order our clustering to be as accurate as desired. Although, we want a number of clusters which make sense. Thus, we will choose a number of clusters which make sense for the specific clustering given the fact that is extremely low, compared with the other k whose NMI scores are lower. Both from results represented in the notebook and the plot, we can understand that there is a peak in NMI score for 27 clusters. However, the NMI score is extremely low as we mentioned before but the number of clusters seems logical to be 27 since we already know that the actual labels are 18.

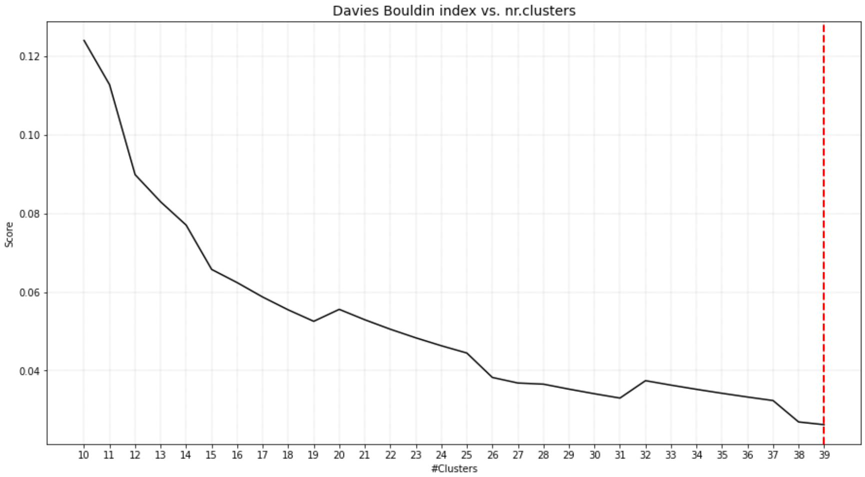
- **NMI, as ground truth is used the century.**

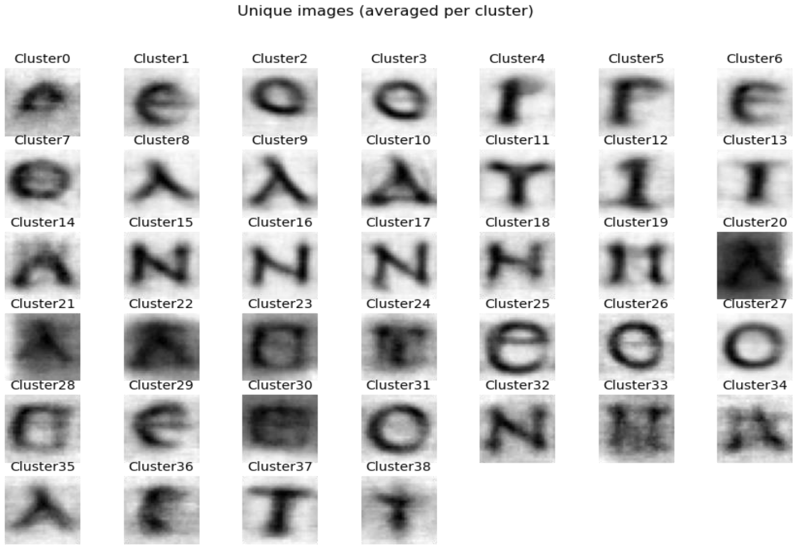
For this case, both for the mentioned results represented in the notebook and the plot, we can understand that there is a peak in NMI score for 7 clusters. However, the NMI score is extremely low, but the number of clusters seems logical to be selected 7 since we already know that the actual labels are 3. Also, if we want to make our clustering more 'accurate' we can choose 35 clusters. For k=35 NMI score seems also to have a peak. Although, as we already know the actual labels are 3, thus it does not make sense to take 35 clusters. Finally, assuming that the selection of k=35 will give as better results, it is obvious that NMI score even in this case remains extremelly low (almost 0.23).



Part 5

We are looking for the lower Davies Bouldin score as the lower the average similarity is, the better the clusters are separated and the better is the result of the clustering performed. Thus, from the below plot, it is obvious that for the specific range [10,40] the 'best' Davies Bouldin score is achieved for 39 number of clusters (db score=0.03). By doing experiments, it is observed that as the range of numbers of cluster increases, the db score decreases. This fact, seems to be logical since as the number of clusters converges to the number of the rows of the dataset the db score would be approximately zero. In this way, we have made a clustering which does not make sense. Our target is to find a number of clusters which make sense, since we already know that the actual labels are 23 in this case and give us a low db score value. From the above plot, we can easily understand that the actual labels which are 23, is a number of clusters which clusters the instances satisfactory with a value of db score almost 0.05. Then, we observe that for the range [24, 30] the db score decreases until it takes its minimum value for K=31 and then starts to increase again whereas for k=38 starts again to decrease. We have to mention that the minimum db score (for the specific range) is observed for k=39 (almost 0.03). However, given the fact that the db score decreases as the number of clusters increases, we would claim that the number of cluster k=31 is enough satisfying since the db score achieves a low score and also make sense to take 31 clusters, since we have already seen (from the part 3) that the classification of the images for 23 labels is not enough satisfying. Below, is represented the visualization plot per cluster per character.





From the specific plot, for 38 clusters, we observe that the algorithm cannot easily discrete some images which depicts the same character. For example, it seems to confuse the character N since it has made 5 different clusters with the same character. These results seem logical since even a person confront difficulties to discrete some letters given the fact that in some cases some pixels are not easily discreet.