Exercise 2: Digit Recognizer task

Task

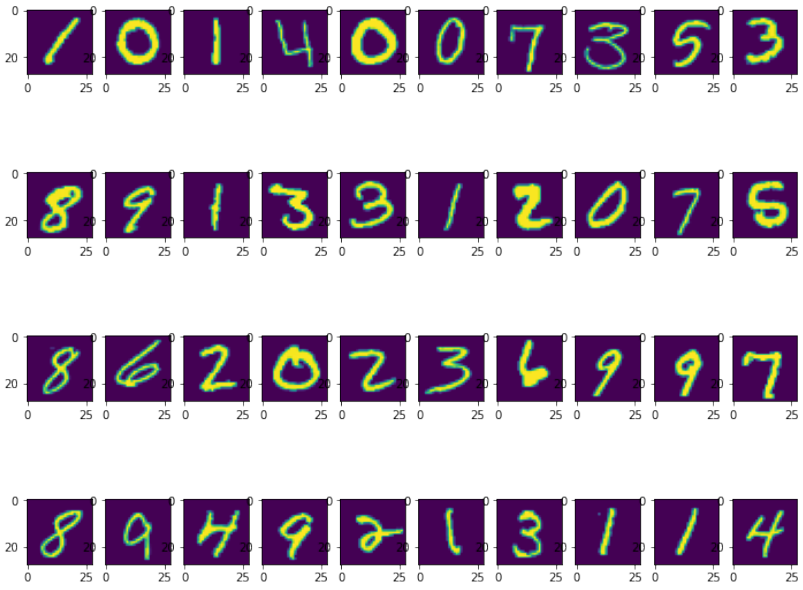
In this assignment we had to experiment with different classification algorithms and setting to decide on the model achieving the best performance on the MNIST ("Modified National Institute of Standards and Technology") dataset.

Data Analysis

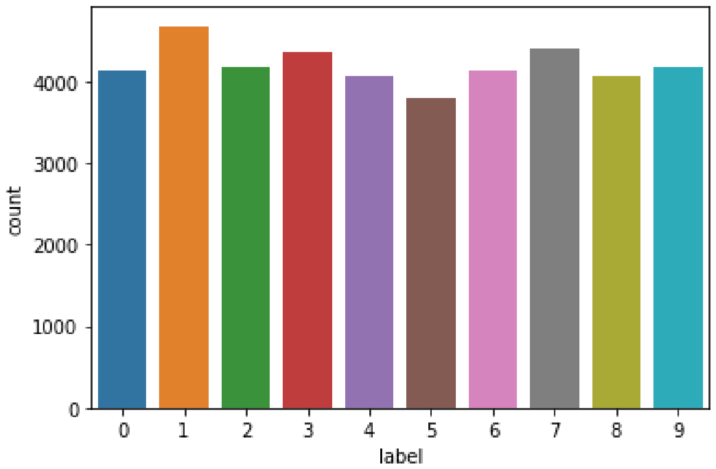
Before we begin experimenting with different classification methods and settings, we performed a preliminary data analysis.

The dataset we used, consists of 42000 observations and 784 features for each observation. Each observation is an image of a handwritten digit and each feature is the value of a pixel of this image. Each image is 28 pixels in height and 28 pixels in width, accounting for 784 pixels in total. This pixel-value is an integer between 0 and 255, inclusive.

Afterwards, we inspected manually some of the images. These images can be seen below. From a first sight, we can observe that the pixel on the edges are the same (totally black) in the vast majority of the samples. This is a strong side that these pixels can be omitted during the classification task without a significant loss of information about the digit depicted. Of course, this procedure won't be done manually, but by using a feature selection algorithm.



We also analyzed that the frequency of each digit. A plot for this analysis can be found in the image below. It is easily observable that the samples for each class are almost balanced. This means that the macro-average of the evaluation metrics for each class will be representative of the performance of the machine learning algorithms we will experiment with. In addition, we don’t need to deal with the difficult problem that a class imbalanced dataset creates.



Machine learning algorithms

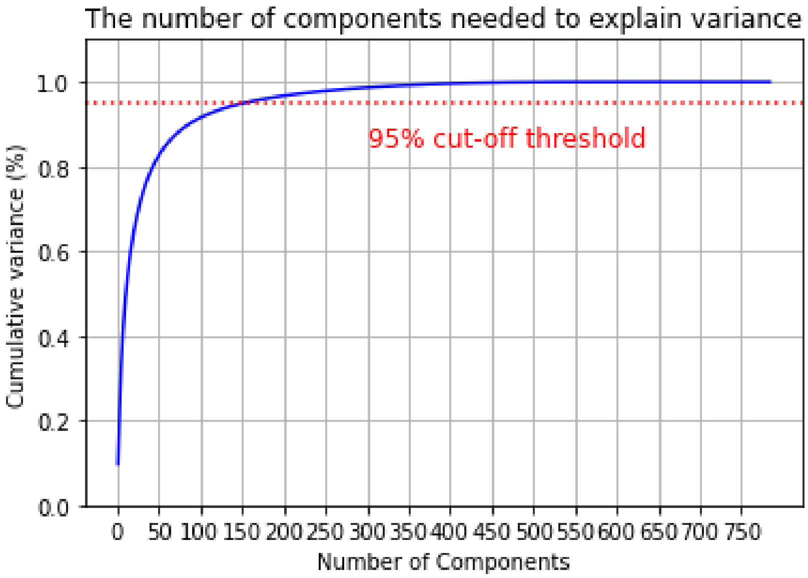
We decided to initially experiment with a linear classifier to investigate whether a linear combination of the features is able make the right classification decision. In particular, we tried a logistic regression classifier which is one of the most famous and well-performing linear classification algorithms.

Afterwards, we tried some non-linear classifiers, in particular Random Forest and SVM with RBF kernel, to investigate whether by adding non-linearity on the combination of the features used to make the classification can improve the results. Within this scope, we also experimented with an ensemble non-linear classifier, namely Random Forest.

Feature Selection and Feature Reduction

As we mentioned in the data investigation section, the pixels on the edges of the images seem to provide no information about the sample class. Thus, we decided to use a feature selection algorithm to reduce the number of feature used and eliminate the noisy features. The algorithm we used was the *SelectKBest* with Chi-squared score function and 200 features as target.

Furthermore, we experimented with (not at the same time with the feature selection) Principal component analysis (PCA) in order to reduce the number of features used by creating new features that captures the vast majority of the variance of the initial features. We set as a target to capture 95% of their variance, and as a result we ended up with 153 new features. The plot of the explained variance according to the number of components can be seen in the figure below. This figure explains why we chose to use 153 components.



Evaluation metrics

We evaluated our classification algorithms by using the well-known accuracy, precision, recall and f1 score metrics. In addition, for each experimental setup we plotted the confusion matrix where we can observe what kind of missclassification we have, meaning which digits are classified and which is the predicted class.

We also used both stratified holdout and stratified cross-validation to ensure that our results are solid.

Experimental Setup

As mentioned above, we experimented with four different classification algorithms. For each one we tried to evaluate its performance without any feature selection/reduction, with PCA applied and after feature selection. Also we evaluated their performance with both stratified holdout and stratified cross-validation.

Results

In the Jupyter Notebook with the code used for this assignment, we report all the results we obtained. In particular, it contains the results for each class, each experimental setup, each classification algorithm and each evaluation metric.

From these results, we can observe for the setup without feature election or feature reduction, both the three non-linear classifiers are having almost the same performance (96-97% for f1 score). On the other hand, Logistic Regression is the least effective algorithm with 91% f1 score.

When PCA was applied, we the “order” of performance remained the same but the results were slightly better. Finally, feature selection led us to worse results.

Conclusions

First of all, it is clear that all the classification algorithms we tried achieved very high scores, which was expected as it is well-knows that the task is not so difficult. In addition, as we expected non-linear algorithms has better performance as the were able to capture the relationship between the features and the labels.

Also we observe that PCA improved slightly the results and also reduced the training time of the algorithms. On the other hand, feature selection worsened their performance, although it reduced the training time too. Therefore, it is not clear if applying PCA is beneficial as this procedure it time-demanding and the improvement in the results is not clear.

A limitation of our experiments is that we were not able to tune the algorithms, as it was proven to be time-consuming. However, given the performance achieved, we believe that it will not provide any significant improvement.