Exercises – Classification

1. Training a classification model

a) Study the code already available in the initial file (classification.py) – it is based in the scikit-learn library and starts by loading the MNIST dataset, a dataset of handwritten digits (0-9). The first 60000 images should be considered training samples, and the remaining 10000 images should be test samples.
b) Separate the dataset into training and testing data. Note that the samples are ordered by digit, and therefore the training data should be shuffled before using it to train a model.
a) Create a classifier, for example a linear SVM (using `SGDClassifier`), and train a model (`fit` method);
b) Test this model (`predict` method) using the sample `some_digit` - did it classify correctly?;
c) Perform cross-validation to train the model using `cross_val_score`, and analyse the scores. Use the following parameters: `cv=3, scoring="accuracy";`
d) Standardize the input features using the `StandardScaler`. Retrain the model using cross-validation and analyse the scores – did it improve?;
e) Plot the confusion matrix using `confusion_matrix` and `plot_confusion_matrix` (available in the `classification.py` file). Use `cross_val_predict` to get the predictions that are used to create the confusion matrix;
f) Test a different classifier, for example a KNN classifier (using `KNeighborsClassifier`);
g) To evaluate different hyperparameters of the KNN classifier use grid search with GridSearchCV. Test the following set of hyperparameters: `[{'weights': ["uniform", "distance"], 'n_neighbors': [3, 4, 5]}].`

2. Challenge (closer to the project)

a) Download the CIFAR-10 dataset [https://www.cs.toronto.edu/~kriz/cifar.html]. The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class – 5000 should be used for training and 1000 for testing;
b) Extract a Bag of Words descriptor for each image using the code implemented in the exercises of the previous class. This step may take a while, therefore store the descriptors.
c) Train and evaluate a classification model;
d) Try different classifiers.

These exercises were implemented and tested using scikit-learn 0.20.0. Documentation of the API and useful code examples can be found here:

- https://scikit-learn.org/stable/modules/classes.html