Report Machine Learning Second Assignment

Stefano Branchi [207523]
stefano.branchi@studenti.unitn.it

University of Trento

Abstract

The report for the second assignment of Machine Learning. The aim of this assignment is to use base machine learning model, in this case a SVM is used, finetuning it with K-fold cross validation. It has been used Python 3, numpy, matplotlib and sk-learn.

1 Dataset

For this assignment is possible to choose one dataset between three different ones: OCR (Optical Character Recognition), Spambase (Spam email classification) and Presidential campaign tweets (Classification of tweets from D. Trump and H. Clinton). This report base its results on the Presidential campaign tweets dataset.
The structure of this dataset is the following one:

- **tweets-train-data.csv**
  Training data set is composed by 4833 records with the following composition:
  - **text**: text of the tweet
  - **datetime**: the time of the tweet
  - **retweet_count**: the retweet count
  - **favorite_count**: the favourite count
  - **place_full_name**: the place the tweet was posted from
- **tweets-train-target.csv**
  The person who posts the tweet between DT (Donald Trump) and HC (Hilary Clinton)
- **tweets-test-data.csv**
  It is composed by 1611 records and it has the same structure of the train data set
- **tweets-test-target.csv**
  It has the same structure of the train target set

2 Pre-processing: Data Preparation

Since this dataset is composed by words, a NLP (Natural Language Processing) pre-processing approach is necessary to clean data and make it useful to the Machine Learning algorithm. To clean data I've performed few steps:

1. **Create dictionary**
   A dictionary of all the words in the tweets is created, so a set of all words in lowercase is used as dictionary. In this way each tweets is taken as a feature vector

2. **Remove stop words**
   From the dictionary stop words are removed, to remove noise from the meaning of the tweet

3. **Remove hapaxes**
   words that appear only once are removed too, because they cannot be used in probabilistic terms

3 Test without Cross-Validation

For this task i chose as model the SVM (Support Vector Machine), a linear classifiers selecting hyperplane maximizing separation margin between classes. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall. I made a first experiment with standard parameters (C=10, kernel='rbf, gamma=0.02). Results are shown in figure 1.

	precision	recall	f1-score	support
DT	0.59	0.86	0.70	812
HC	0.74	0.40	0.52	799
micro avg	0.63	0.63	0.63	1611
macro avg	0.67	0.63	0.61	1611
weighted avg	0.67	0.63	0.61	1611

Accuracy: 0.6319056486654252

Figure 1: Results of first attempt of SVM with standard parameters

4 K-fold Cross Validation

After first results with SVM, to improve the performance of the model I used a K-fold Cross validation. In this method the training set is randomly partitioned into k equal sized subsamples. Each subsample will be used once as validation set, and the remaining portion as trianing set. The cross validation process is then repeated k times.
For this assignment I chose a 3 fold cross validation and as possible values to be evaluated:

- C_values = [0.1, 1, 10, 100, 1000]
- gamma_values = [1, 0.1, 0.01, 0.001, 0.0001]

After this i plotted the learning curve for the cross validation. This is shown in Figure 2.

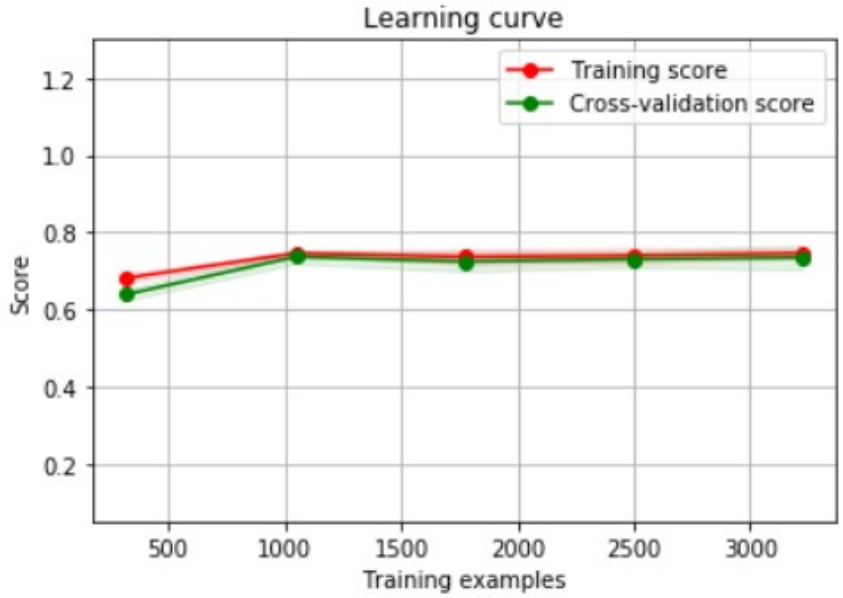

Figure 2: Learning curve of the k-fold cross validation

5 Final Results

After the cross validation the best parameters used are:

- C = 1000

- gamma = 1

I used this parameter to train once again the SVM on the training set and testing it on the test dataset. Results can be found in Figure 3.
It's possible to see an improvement on the results from the first attempt (0.6319) to the final one(0.7865). It also possible to see that results are above the minimum value of accuracy (0.50403).

	precision	recall	f1-score	support
DT	0.84	0.71	0.77	812
HC	0.75	0.86	0.80	799
micro avg	0.79	0.79	0.79	1611
macro avg	0.79	0.79	0.79	1611
weighted avg	0.79	0.79	0.79	1611

Accuracy: 0.7864680322780881

Figure 3: Results with best parameters retrieved from cross validation