Homework 3

Web Information Retrieval

Deadline: **17 June 2016**

Exactly Two students for each group

Data and software are available at:

http://www.diag.uniroma1.it/~fazzone/Teaching/WebIR_2016_2017/WebIR_2016_2017.html

Description:

In this homework you are going to solve text classification problems by applying supervised learning techniques.
All datasets provided for the homework are composed by cleaned documents containing only words, numbers and single spaces. For this reason, no further cleaning processes on data are required.
To try to avoid overfitting, **it is mandatory to perform a 10-Fold-Cross-Validation** process in the training-validation phase. Due to the time consuming nature of this process, **it is mandatory to use all available CPU cores** on your machine.

The homework is split in two parts:

- In the first part, you have to tune and train a classifier based on **kNN** technique for detecting spam comments related to YouTube videos. Inside the file "*Ham_Spam_comments.zip*" you can find spam and ham comments already split in Training and Test set.

- In the second part, you have to tune and train **three** different classifiers to solve a sentiment analysis problem. These classifiers have to be able to classify sentences representing positive and negative opinions about movies.
  One of these three classifiers **must** be a kNN classifier, while the other two **must** use different classification techniques from kNN. The two classifiers that do not use a kNN technique must be able to achieve better performances on the provided Test set than the ones obtained by the trained and tuned kNN classifier.
  Inside the file "*Positve_negative_sentences.zip*" you can find sentences labeled as negative or positive. All these sentences are already split in Training and Test set.

To represent documents as vectors, you must use this class offered by scikit-learn libraries:
 `sklearn.feature_extraction.text.TfidfVectorizer`.
Please, have a deep read to the documentation.
Due to the fact that the documents are already cleaned, you should set "`strip_accents=None`" and "`preprocessor=None`".

To train and tune classifiers you must use this classes offered by scikit-learn libraries:
`sklearn.pipeline.Pipeline`
`sklearn.model_selection.GridSearchCV`
Please, have a deep read to the documentation associated to the `GridSearchCV` class.

To assess the performance of classifiers you must use the Matthews-correlation-coefficient (Sklearn.metrics.matthews_corrcoefs). In particular, to choose the best parameters for each classifier considered in the homework, you must use Matthews-correlation-coefficient. This means that the GridSearchCV object must use the Matthews-correlation-coefficient to choose the best parameters for each considered classifier.

Zoom On Part One:

In this part of the homework you have to tune and train a classifier for detecting spam comments associated to YouTube videos. To solve this Spam-Ham classification problem, you are forced to use a kNN classifier and, by using TfidfVectorizer, Pipeline, GridSearchCV and matthews_corrcoefs tools, you have to find the best combination of parameters for solving the problem.
To find the best combination of parameters, you must perform a 10-Fold-Cross-Validation process by using all CPU cores available on your machine. For that purpose, it is enough to initialize a GridSearchCV object in a proper way.
At the end of the previous process, you have to train your kNN classifier, tuned with the best configuration of parameters, using all documents in the Training-Set (you can do this in an easy way by using the GridSearchCV object) and evaluate its performance on the Test-Set.
For the evaluation process you have to

- use the `metrics.classification_report` tool provided by scikit-learn to have a report with the main classification metrics.
- compute the Confusion-Matrix using the method `metrics.confusion_matrix`.
- compute the Normalized-accuracy using the method `metrics.accuracy_score`.
- compute the Matthews_corrcoef, of course.

To tune and train the classifier, you have to use the data collected inside the directory "*Training*" contained inside the file "*Ham_Spam_comments.zip*".
To evaluate the performance of the final classifier, you have to use the data collected inside the directory "*Test*" contained inside the file "*Ham_Spam_comments.zip*".

You can find an example on how to use all these scikit-learn tools here. Of course, you can use that code to perform the homework. You are also allowed to use tools offered by NLTK libraries: for example, to perform stemming and to have a set of stopwords.

Zoom On Part Two:

In this part of the homework, you have to tune and train **three** different classifiers able to understand if a text sentence represents a negative or positive opinion about a movie.
One of these three classifiers must use a kNN technique for solving the problem, instead, the other two classifiers must use a classification technique different from kNN. Moreover, it is requested that the performances of these two classifiers must be higher than the ones achieved by the classifier that is based on the kNN technique.

You can find the best combination of parameters for all three classifiers by using TfidfVectorizer, Pipeline, GridSearchCV and matthews_corrcoefs tools.
Even in this part of the homework, you must perform a 10-Fold-Cross-Validation process by using all CPU cores available on your machine.
At the end of the previous process, you have to train all three classifiers using all documents in the Training-Set and evaluate its performance on the Test-Set.
For the evaluation process you have to

- use the `metrics.classification_report` tool provided by scikit-learn to have a report with the main classification metrics.
- compute the Confusion-Matrix using the method `metrics.confusion_matrix`.

- compute the Normalized-accuracy using the method `metrics.accuracy_score`.
- compute the Matthews_corrcoef, of course.

To tune and train the three classifiers, you have to use the data collected inside the directory “*Training*” contained inside the file “*Positve_negative_sentences.zip*”.
To evaluate the performance of the final classifiers, you have to use the data collected inside the directory “*Test*” contained inside the file “*Positve_negative_sentences.zip*”.

To complete the homework, you have to complete these steps:

1. Download and decompress the Dataset_for_Part_1 and Dataset_for_Part_2.
   You can find what you need at this link
   http://www.diag.uniroma1.it/~fazzone/Teaching/WebIR_2016_2017/WebIR_2016_2017.html in section “Homework_3”.
2. Solve the Spam-Ham classification problem as requested in “Part-One” using scikit-learn libraries by putting all code inside the file `WebIR__Homework_3_part_1.py`.
3. Solve the Negative-Positive classification problem as requested in “Part-Two” using scikit-learn libraries by putting all code inside the file
   `WebIR__Homework_3_part_2.py`.
4. Create a final report **in PDF**.
5. Create a zip file with this content:
   -) `WebIR__Homework_3_part_1.py`
   -) `WebIR__Homework_3_part_2.py`
   -) A single human-readable file containing the final report **in PDF**.
   -) Nothing less, nothing more.
6. The file name must have this format:
   WebIR_Homework_3__StudentID_StudentName_StudentSurname__StudentID_StudentName_StudentSurname.zip
7. Finally, you must to send this zip file to fazzone@diag.uniroma1.it with this email subject:
   WebIR_Homework_3__StudentID_StudentName_StudentSurname__StudentID_StudentName_StudentSurname.

What To Put in the Report:

For what concerns the first part, the final report must contain:

1. The following table:

	Training Set	**Test Set**
Spam	num_data	num_data
Ham	num_data	num_data

2. The list of all parameters to tune with the corresponding set of values.

3. A simply and short description on how to perform the training-validation process using more than one CPU core.

4. The best configuration of parameters found by the GridSearchCV object at the end of the 10-Fold-Cross-Validation process.

5. The output of the metrics.classification_report tool.
6. The Confusion-Matrix with labels on columns and rows.
7. The Normalized-Accuracy value.
8. The Matthews-Correlation-Coefficient value.

For the second part the final report must contain for each classifier exactly what requested for the first part.

For any problem/doubt/question, you must use Piazza.
Please, don't be shy :)