Bologna Business School – Data Science & Fintech Masters
Text Mining Unit – Opinion Mining & Sentiment Analysis
Prof. Gianluca Moro, Dott. Ing. Nicola Piscaglia – DISI, University of Bologna

Opinion Mining Teamwork (1.5 h)

The following work must be executed in teams of two persons indicated by the teacher.

To get started, create a new Jupyter notebook file and enter the following commands in a code cell:

```
!wget -q -O exam.py https://tinyurl.com/4ufh4xbk
%cat exam.py
```

Execute the cell once, copy the code that appears in a new cell. In the new code cell, set the value of the variable *your_team_number* to the number assigned to your team by the teacher and then execute the cell: all the data necessary for the teamwork will be downloaded and stored in variables, whose names will be printed in output. After this, execute the points indicated below by developing them in the notebook.

At the end, the file must contain **all the required results** (as code cell outputs) along with **all the commands** necessary to reproduce them; the function of **every command** or group of related commands **must be documented** clearly and concisely. The name of **every variable** defined in the commands (not counting the ones provided by the initial steps) must have the **initial letters of your last names as a prefix** (e.g. "***sj***_train_set" for *Smith and Johnson*). In order to work in pairs, you can access the same Google account that you created for your group and edit the notebook on Google Colab, but be careful to not overwrite the changes made by the other member of your group (to avoid this, you can edit a separate copy of the notebook and then merge the two members results before the end of the test). **You have 1.5 hours to complete the test**.

When finished, the team member with an alphabetically lower surname will send the notebook file (having .ipynb extension) via mail (using your BBS email account) to the teacher (nicola.piscaglia@bbs.unibo.it) indicating "[BBS Teamwork] *Your last names*" as subject, also keeping an own copy of the file for safety.

You are allowed to consult the teaching material and to search the Web for quick reference. **It is severely NOT allowed to communicate with other teams.** Ask the teacher for any clarification about the exercises.

(1) Unsupervised classification with opinion lexicon

A lexicon with sets of commonly used positive and negative words is provided in the variables pos_words and neg_words, respectively. Classify the reviews provided in the reviews_A test set by first assigning to each a score equal to the number of known positive words within it minus the number of negative words, then return "pos" for reviews with a positive score and "neg" for reviews with a negative or null score.

Evaluate the classification accuracy, i.e. the ratio between the number of correctly classified reviews and the total count of test reviews.

(2) Supervised classification with plain term frequency

Train a logistic regression classifier on the reviews provided in reviews_B. Build a vocabulary of words from the training set, filtering out words appearing less than 3 times, then represent each review with a bag of words based on count of occurrences of each term.

As above, test the model by computing the accuracy on the reviews provided in reviews_A.

(3) Supervised classification with tf.idf

Train and test a logistic regression model following the same specifications given in **(2)**, but using the tf.idf weighting scheme in place of the plain occurrences count.

(4) Supervised classification with bigrams

Train and test a logistic regression model following the same specifications given in **(3)**, but including bigrams as feature in addition to single words.

(5) Cross-domain classification

Test the same three models trained in **(2)**, **(3)** and **(4)** on the reviews_C dataset, containing reviews about a different category of items.

(6) Comparison of models

For each pair of the four classifier considered in **(1)**, **(2)**, **(3)** and **(4)**, indicate whether the predictions provided by the two compared classifiers on the reviews_A test set are significantly similar or different using the McNemar's test (consider a 95% confidence level, i.e. p-value must be > 0.05 for models to be significantly similar). To obtain the p-value, use the provided mcnemar_pval function providing the arrays with the labels predicted by the two models.

```
mcnemar_pval(model1_predictions, model2_predictions)
```