Final Project
Machine Learning Applications
Bachelor in Data Science and Engineering

March 15, 2023

1 Introduction

In this project, students will use the knowledge and techniques acquired during the course to solve a machine learning task on text documents. Students can work in groups of a maximum of three people. It is important that, regardless of how each group chooses to distribute the work, all the components of the group know the complete project. The evaluation of the final project will be carried out through the delivery of a report and a group presentation followed by questions.

The project consists of the following tasks:

- Task 1. Natural Language Processing and text vectorization
- Task 2. Machine Learning:
  - Task 2.1. Classification, Regression using feature extraction or selection techniques
  - Task 2.2. Clustering using feature extraction or selection techniques
  - Task 2.3. Recommendation Systems
- Task 3. Implementation of a dashboard using the Python *Dash* library.
- Task 4. Final report and presentation.

For the execution of the final project, students must choose to implement any of the sub-Tasks 2 (either 2.1, 2.2 or 2.3), depending on their preferences and the possibilities of the database used.

2 Data set creation

For the completion of the final project, it is preferred that students work on their own data set. Said data set can be a collection of documents available in the open, or it can be created from a collection of Internet pages. Opting for the second of the proposed alternatives (i.e., creating your own data set) will positively affect the final grade of the project.

Bear in mind that for the completion of the final project it will be necessary for your data set to consist of at least several thousand documents in text format, in order to facilitate the construction of sufficiently representative topic models as well as other vectorization approaches. In addition, each document must have additional metadata that can be used for the implementation of Tasks 2.1, 2.2 or 2.3. For instance, to carry out Task 2.3 the data set needs to contain rating information or in Task 2.1 you need any additional variable (categorical or real) to be used as target of a classification or regression task.

In any case, check with one of the professors of the course, once you have decided on the collection of documents to use, to obtain guidance about its viability or possible difficulties in the implementation of the tasks. Make the selection of the data set early, to avoid possible delays that could jeopardize the presentation of the project within the established deadline.

3 Task 1: Text Preprocessing and vectorization

This task will consist of the thematic analysis of the collection provided. The steps you must follow in your work are as follows:

- Step 1: Implementation of a pipeline for the preprocessing of the texts. For this task you can use the usual libraries (NLTK, Spacy), or any other library that you consider appropriate.
- Step 2: Text vectorization. In this stage you will analyze the following vectorization schemes:
  - Classical BoW or TF-IDF representation.
  - Word2vec/FastText based representation or Doc2Vec vectorization.
  - Extraction of themes and vector representation of the documents using the LDA algorithm.

In the report you must include a description of the preprocessing pipeline used as well as the vectorization strategies that have been explored. For instance, in the Word2vec/FastText based representations you must explain how you convert a set of word vectors into a document vectorization or for the topic model you have to explain how you have carried out the selection of the number of topics. Any additional representation which helps to analyze this vectorization will be welcome (and positively evaluated).

4 Task 2: Machine Learning model

For this part of the project students must select at least one of the following subtasks.

Task 2.1: Classification or Regression Task

Implementation and **evaluation** of the performance of a classifier or regression model for the dataset used. Use one of the metadata available in the dataset as your target variable: a categorical variable if you opt for a classification task, or a real type variable for regression. Note that discrete but ordered variables (such as dates, scores, etc.) can also be used as target variables for a regression task.

For this task, you will need to compare the performance by using the different document vectorizations. In addition, you must use for your work some of the feature extraction or selection algorithms described in the course, analyzing their impact on the results obtained. Use the usual metrics for performance analysis, i.e., error rates, ROC curves, confusion matrices, etc., if you pose a classification task, or the root mean square error if you choose a regression model.

To adjust the hyperparameters of the classification or regression models, you must use a validation methodology that must also be explained in the report.

Task 2.2: Clustering Task

In case the dataset does not have a clear variable that can be used for document classification or to solve a regression problem, this task can be approached as an unsupervised learning task and document clustering can be performed.

In this case the clustering results obtained using the different vectorizations obtained from the previous task should be explored. To analyze and compare the results with each other, measures based on clustering consistency such as the silhouette coefficient[1] can be used or an analysis of the obtained clusters (centroids and distribution of documents in each cluster) can be carried out.

For the selection of the optimal number of centroids, the analysis of the silhouette coefficient or other measures specific to the particular algorithm (e.g. the sum of squared distances of samples to their closest cluster center for the K-means) can also be used.

In addition, you must include here some of the feature extraction or selection algorithms described in the course, analyzing their impact on the clustering results.

Task 2.3: Recommender Systems

In this case, you will have to implement a content based recommendation system using the different document vectorizations obtained from Task 1. You can easily design a neighborhood-based approach where you recommend to each user similar products to those they already like and use the vectorization of the documents describing each product to calculate these similarities.

The performance of this model must be compared with a collaborative filtering system where you can explore neighborhood based versions (either user based, content based, or both) or latent based methods such as ALS. Of course,

[1] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html

for the implementation and evaluation of these approaches you can use the Surprise library.

To properly complete this task, you will have to train the different methods (selecting their parameters adequately), and evaluate their performance. All these steps have to be clearly explained in the report.

5 Task 3: Implementation of a dashboard

Students need to implement a dashboard using Python library *Dash*. Said dashboard must include at least one figure related to the predominant LDA topic of each document, as well as a minimum of two additional representations related to other variables. Your dashboard must be interactive, i.e., the user will be able to make selections in one or more of the included charts, and the rest of the charts will modify their values according to the selection made.

6 Task 4: Final Report and Presentation

For the evaluation of the project, students will need to hand in a written report attaching the code for their implementation. A presentation will also be carried out, where the students must describe their methodology with a few slides.

6.1 Final report

Students must hand in via Aula Global, by the indicated deadline, the following items:

1. Python script or Jupyter notebook with the implemented code duly commented

2. A separated python script or Jupyter notebook with the dashboard implementation.

3. Descriptive report of the work carried out in .pdf format and a maximum length of 12 pages (excluding only cover and references). The report should consist of four main sections:

    - Task 1 (max. 6 pages)
    - Task 2 (max. 5 pages)
    - Brief description of the implemented dashboard (max. 1 page).
    - Acknowledgment of authorship. Inexcusably, the report must respect the principle of recognition of authorship. If you have used extraneous code snippets or any material from external sources, you must clearly specify this in the report. Failing to do so, may result in the loss of the entire grade for the final project.

4. (Optional) A short video illustrating the functionality of the dashboard.

6.2 Presentation

All team member participants should contribute to the presentation, and individual grades could be granted for this item. Later on, several time slots will be offered, so that groups can book their preferred slot for the presentation.

7 Grading

The maximum mark of the final project is 4 points (+0.5 extra), which will be distributed as follows:

- Project execution and documentation: $2,75$ points.
  - Task 1. Natural Language Processing, Topic Modeling and Document Vectorization: $1,25$ points
  - Task 2. Machine Classification or Regression/Clustering/Recommendation Systems using characteristics extraction or selection techniques: 1 point
  - Task 3. Dashboard: $0,5$ points

  For Tasks 1 and 2 the following aspects will be considered:

  - Methodology (45%): methodological correctness of your implementation. This includes the correct application of the methods, but also other aspects, such as normalization, hyperparameter validation, selection of evaluation metrics, strategy for evaluating the performance, etc.
  - Memory quality (40%): the most important aspect that will be assessed is the discussion of your results for which you are encouraged to provide graphical representations supporting your conclusions. Formal presentation will also be taken into account.
  - Code quality (15%): Organization, code efficiency, adequate comments will be taken into account here.
- Project presentation: $1,25$ points. All team member participants should contribute to the presentation, and individual grades could be granted for this item.

  Each presentation will be followed by questions by the teachers that can be addressed individually to the different members. The objective of this phase is to verify that all members of each group have participated effectively in the final project, and are aware of its implementation and results in sufficient detail.

  If during the Q&A it is found that a member of the team has significant knowledge gaps about the project, the teaching team could apply a penalty factor to that member, which could even result in the total loss of the project's qualification.

Peer evaluation of dashboards

Students will be able to get an additional +0.5 points through a peer review of the dashboard implemented using the *Dash* library. For this, the teachers will publish the videos received in a private YouTube list, and a voting system will be implemented so that the three winning groups will obtain the additional points.

To be eligible for this additional half point you must submit a short video (no more than two or three minutes in length) describing the implemented dashboard. Sending the video implies an implicit authorization for its publication in a private YouTube playlist.

All Projects must be handed in via Aula Global. The deadline will be Sunday, May 7, at 23:59. Delays will result in a penalty of $0,025$ point per hour of delay.

Presentations will take place on May 9 and 10. A doodle will be published soon for the teams to book time slots for their presentations.

Dashboard video deadline will be May 10, at 23:59. No late submissions will be accepted.