INF342: Domain Name Classification Challenge

Athens University of Economics and Business

April 2022

1 Description of the Challenge

The goal of this project is to study and apply machine learning/data mining techniques to a real-world classification problem. In this classification problem, each sample corresponds to a domain name and the task is to assign each domain name into a predefined category (i.e., class). Such techniques find applications in several fields, such as in managing web hierarchies, and in question answering systems. In the context of this project, you are given both a web graph consisting of several thousands of greek domain names, and the textual content of the webpages of a subset of these domain names. Your task is to build a model for predicting the class label of each domain name. The problem is very related to the well-studied problems of text categorization and node classification. The pipeline that is typically followed to deal with the problem is similar to the one applied in any classification problem; the goal is to learn the parameters of a classifier from a collection of training products (with known class information) and then to predict the class of unlabeled products.

The challenge is hosted on Kaggle[1], a platform for predictive modelling on which companies, organizations and researchers post their data, and statisticians and data miners from all over the world compete to produce the best models. The challenge is available at the following link: `https://www.kaggle.com/competitions/inf342-datachallenge-2022/`. To participate in the challenge, use the following link: `https://www.kaggle.com/t/88f96f181e19411db2d0154f1612613f`.

2 Dataset Description

As mentioned above, you will evaluate your methods on a dataset of greek domain names. You are given the following files (which are available at: `https://www.dropbox.com/sh/l0tfrn4g0ilem6i/AACZApUNGuQyUG9PcbklnFCTa?dl=0`).

1. **edgelist.txt**: a large part of the greek web graph stored as an edgelist. Nodes correspond to domain names (i.e., *aueb.gr*) and edges to hyperlinks. For example, if there is a hyperlink from some webpage of domain v to at least one webpage of domain u, there is a directed edge from node v to node u in the graph. The graph consists of $65,208$ vertices and $1,642,073$ directed edges in total.

2. **domains.zip**: it contains the textual content of webpages extracted from $40,600$ greek domain names. The textual content has been extraced from the HTML source code of the webpages. For each domain name, there is a zip file containing the textual content of its webpages.

[1] `https://www.kaggle.com/`

3. **train.txt**: it contains $1,258$ labeled domain names. Each row of the file contains the name of a domain name and its topic. The comma character is used to separate the domain name from the class label.

4. **test.txt**: this file contains the names of 547 domain names. Each of these domain names belongs to one of the 10 possible classes. The final evaluation of your methods will be done on these domain names and the goal will be to predict the category to which each domain name belongs.

With regards to the 10 classes, they correspond to different topics such as sports, news, etc. The number of samples of each class ranges between 22 and 344 and thus the dataset is highly imbalanced.

3 Evaluation

The performance of your models will be assessed using the multi-class logarithmic loss measure. This metric is defined as the negative log-likelihood of the true class labels given a probabilistic classifier's predictions. Specifically, the multi-class log loss is defined as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(p_{ij})$$

where N is the number of samples (i.e., domain names), C is the number of classes (i.e., the 10 categories), y_{ij} is 1 if sample i belongs to class j and 0 otherwise, and p_{ij} is the predicted probability that sample i belongs to class j.

4 Provided Source Code

You are given two scripts written in Python that will help you get started with the challenge. The first script (`graph_baseline.py`) uses solely graph-based features with a logistic regression classifier for making predictions. The second script (`text_baseline.py`) uses features extracted from the webpages of the domain names along with a logistic regression classifier. As part of this challenge, you are asked to write your own code and build your own models to predict the class label of each domain name. You are advised to use both graph-theoretical and textual information.

5 Useful Python Libraries

In this section, we briefly discuss some tools that can be useful in the challenge and you are encouraged to use.

- A very powerful machine learning library in Python is `scikit-learn`[2]. It can be used in the preprocessing step (e.g., for feature selection) and in the classification task (several regression algorithms have been implemented in `scikit-learn`).

- A very popular deep learning library in Python is `PyTorch`[3]. The library provides a simple and user-friendly interface to build and train deep learning models.

[2] `http://scikit-learn.org/`
[3] `https://pytorch.org/`

- Since you will deal with data represented as a graph, the use of a library for managing and analyzing graphs may be proven important. An example of such a library is the `NetworkX`[4] library of Python that will allow you to create, manipulate and study the structure and several other features of a graph.
- Since you will also deal with textual data, the Natural Language Toolkit (`NLTK`)[5] of Python can also be found useful.
- `Gensim`[6] is a Python library for unsupervised topic modeling and natural language processing, using modern statistical machine learning. The library provides all the necessary tools for learning word and document embeddings.

6 Rules and Details about the Submission of the Project

Rules. The following rules apply to this challenge: (i) one account is allowed per participant (ii) there is a limit in the size of each team (at most 3 members), (iii) privately sharing code outside of teams is not permitted, (iv) there is a limit in the number of submissions per day (at most 5 entries per day).

Evaluation and Submission. Your final evaluation for the project will be based on (1) the presentation you will give (**50**%), (2) on your position on the private leaderboard and the log loss that will be achieved (**20**%), and (3) on your total approach to the problem and the quality of the report (**30**%). As part of the project, you have to submit the following:

- A 4-5 pages report, in which you should describe the approach and the methods that you used in the project. Since this is a real classification task, we are interested to know how you dealt with each part of the pipeline, e.g., how you created your representation, which features did you use, which classification algorithms did you use and why, the performance of your methods (log loss and training time), approaches that finally didn't work but is interesting to present them, and in general, whatever you think that is interesting to report.
- A directory with the code of your implementation (not the data, just the code).
- Create a `.zip` file containing the code and the report and submit it to the e-class platform.
- **Deadline: 10/6/2022 23:59**

Presentation: As mentioned above, you will be asked to present the approach you followed. Therefore, you will need to prepare some slides (using ppt or any other tool you like).
Date of presentation: 14/6/2022

[4] `http://networkx.github.io/`
[5] `http://www.nltk.org/`
[6] `https://radimrehurek.com/gensim/`