Data Mining

Homework 2

Due: 15/11/2020, 23:59

Instructions

You must hand in the homeworks electronically and before the due date and time.

The first homework has to be done by each **person individually**.

Handing in: You must hand in the homeworks by the due date and time by an email to Andrea (`mastropietro@diag.uniroma1.it`) that will contain as attachment **(not links to some file-uploading server!)** a .zip file with your answers. The filename of the attachment should be `DM_Homework_1__StudentID_StudentName_StudentLastname.zip`;
for example:
`DM_Homework_1__1235711_Robert_Anthony_De_Niro.zip`.
The email subject should be
`[Data Mining] Homework_1 StudentID StudentName StudentLastname`;
For example:
`[Data Mining] Homework_1 1235711 Robert Anthony De Niro`.
After you submit, you will receive an acknowledgement email that your project has been received and at what date and time. If you have not received an acknowledgement email within 2 days after you submit then contact Andrea.

The solutions for the theoretical exercises must contain your answers either typed up or hand written clearly and scanned.

For any questions on the homework, clarifications, and so on, contact Andrea (`mastropietro@diag.uniroma1.it`).

For information about collaboration, and about being late check the web page.

Problem 1. In this exercise we will get some practice in using some Python libraries, for downloading web pages, parsing them, and performing some analysis. We will obtain data about articles available on Amazon. For downloading the web pages you may use the package `Requests` or the package `urllib2`. To parse the page you can either use regular expressions through the package `re` (it is anyway a good idea become familiar with regular expressions), or, probably better, use an HTML/XML parser. The `Beautiful Soup` package is a good one but it loads the whole file in memory. This is fine for this problem, since the pages to parse are small, but be careful if you want to use it on large XML files; for those ones check the `lxml` library and the tutorial at `http://www.ibm.com/developerworks/xml/library/x-hiperfparse/`.

Write a program that will download from `https://www.amazon.it/` and parse all the products output for a given keywork. You can search for a product page using the url `https://www.amazon.it/s?k=KEYWORK&page=X`, where KEYWORD is the product you want to look for and X is the page number. For this exercise use the keywork *computer* (you will see that you'll have many different products, like PCs, notebook covers, stands, etc.). Save the products in a tab-separated value (TSV) file in which, for any product (one line per product) you have: *product description*, *price*, whether it is a *prime product or not*, *url* of the product page, *number of stars* of the product. Because you will make a lot of calls to the Amazon site, make sure that you have a delay (use: `sys.sleep()`) between different downloads of Amazon pages, to avoid being blocked. After downloading, do the following:

1. For each product description, use `NLTK` Python library to perform any preprocessing you may find useful (stemming, normalization, stopwords removal, etc).

2. The next step is to build a search-engine index. First, you need to build an inverted index, and store it in a file. Build an index that allows to perform proximity queries using the cosine-similarity measure. Then build also a query-processing part, which given some terms it will bring the most related product.

Hand in the code, along with some examples of queries and screenshots of the results. Try both short and long queries (like a full product description).

Problem 2. Implement Problem 1 using Apache Spark (with the already downloaded pages).

Problem 3. We want to play a little bit more with Spark. For this problem, you will use `spark.ml`, a Spark package that provides you with several high-level machine learning APIs (feel free also to use a different library, may you know one). You will perform some forecasting using Covid-19 data, available from `Kaggle` here `https://www.kaggle.com/c/covid19-global-forecasting-week-5/overview`. You will download a csv file in which, for any country, you have the time series of coronavirus cases and fatalities, day by day. As you will see, the dataset is already split into train and test. What you have to do is to use any machine learning algorithm suitable to work with time series to perform a prediction of coronavirus cases (include fatalities if you want) in future days. Your algorithm will be trained on the data you will find in `train.csv` and it will be tested on `test.csv` (ignore the file `submission.cvs` that you will find on the website). In the files you have all the countries of the world; consider only some (but remember you can parallelize your computation). What you have to hand in is the code along with some plots of the forecasting made by your model, for some of the countries. Predicting such time series is hard, so I won't expect accurate results. However, comment your results stating why in your opinion your model was or was not able to give an accurate prediction.