SIG731 2023: Task 8HD

Data Cleansing and Text Analysis Challenge

Last updated: 2023-11-24

Contents

Tasks 5–8 are not obligatory; you can submit them in any order (or decide not to tackle them at all). C/D/HD is merely a subjective estimate of their difficulty level. For each task that you successfully complete, you score 10 points (and for those that are not 100% correct, no points will be given).

1 Task

First, research/gather the data:

1. Choose one StackExchange site dealing with topics that you find interesting; see https://stackexchange.com/sites?view=list#traffic for a list. The site cannot be too small, but also avoid selecting any of the largest ones (especially *StackOverflow*, *Mathematics*) unless you *really* want to challenge yourself. As a rule of thumb, let's say that the site must have at least 10,000 questions *and* 10,000 answers.
2. Download the site's most recent data dump from https://archive.org/details/stackexchange.
3. Read the description of all the data tables published at https://meta.stackexchange.com/questions/2677/.

Then:

1. Convert all the data tables (Badges, Comments, PostHistory, PostLinks, Posts, Tags, Users, Votes) from XML to CSV, using custom code that you write yourself. Ideally, you should write a Python function that takes a single input file name (.xml) and output file name (.csv) and performs the conversion of a single dataset.
2. Load the CSV files as **pandas** data frames.
3. Create at least five nontrivial data visualisations and/or tables, at least three of which are based on the extraction of information from text (e.g., tags, keywords, locations, etc.). You must demonstrate that you have learned how to write your own regular expressions (regexes).
4. Draw insightful and interesting conclusions. Do not forget to reflect on the potential data privacy and ethics issues that arise during the data analysis process.

   This HD-level task is purposely under-defined – you will not be told precisely what to do. Your aim is to generate some ***interesting*** *insights into data featuring lots of textual information.*

In the course of the report preparation, you should apply a wide range of data frame wrangling and text processing techniques. In particular, you must demonstrate that you mastered *regular expressions*.

Do not use pie charts (as we discussed during the lecture). Go beyond the basic plots that we have covered in this course. Draw at least one map (e.g., of the world) and a word cloud.

2 Artefacts

At the start of the notebook, you need to provide: the **title** of the report (e.g., *Task 42: How Much I Love This Unit*), your **name**, **student number**, and **email address**.

Then, add 1–2 introductory paragraphs (an introduction/abstract – what the task is about).

Before each nontrivial code chunk, briefly **explain** what its purpose is. After each code chunk, **summarise and discuss the obtained results** (in a few sentences).

Conclude the report with 1–2 paragraphs (summary/discussion/possible extensions of the analysis etc.).

Checklist:

1. Header, introduction, conclusion (Markdown chunks).
2. Text divided into sections, all major code chunks commented and discussed in your own words (Markdown chunks).
3. Every subtask addressed/solved. In particular, all reference results that are part of the task specification have been reproduced (plots, computed aggregates, etc.).
4. The report is readable and neat. In particular:
    - all code lines are visible in their entirety (they are not too long),
    - code chunks use consecutive numbering (select *Kernel - Restart and Run All* from the Jupyter menu),
    - rich Markdown formatting is used (`# Section Title`, `* bullet list`, `1. enumerated list`, `| table |`, `*italic*`, etc.),
    - the printing of unnecessary/intermediate objects is minimised (focus on reporting the results specifically requested in the task specification).

Submissions which do not *fully* (100%) conform to the task specification *on* the cut-off date will be marked as FAIL.

Good luck!