**Dataset**

You will use a dataset consisting of around 3000 user stories about smart home applications. These user stories are a subset of the user stories collected in published paper attached. I encourage you to read the paper, but all details required for this project are available on this page.

Each user story in the dataset is of the format “As a **role**, I want **feature**, so that **benefit**.” The second, third, and fourth column in the dataset correspond to role, feature, and benefit of a user story, respectively. The column is an autogenerated ID (note that that IDs are not contiguous).

For the following tasks, you can treat the combined text in the role, feature, and benefit columns of a row as a document.

**Part 1: Text Preprocessing [Detailed explanation of each step (3pts\*6 = 18pts) + source code (10pts)]**

In Part 1, you will perform the following text preprocessing on the user stories. After each step, you will only retain a subset of the words in each user story.

1. Tokenize.
2. Convert each token to be in lower case.
3. Parts of speech tagging: Retain only nouns, verbs, adjectives, and adverbs.
4. Stop word removal (standard): Remove words from the as stated in https://www.ranks.nl/stopwords
5. Stop word removal (custom): Create a custom list of stop words, e.g., smart, home, smart-home, and so on, and remove words from this custom list.
6. Lemmatize: Reduce each word to its lemma.

**Part 2: TF-IDF Computation and Vector Space Representation [Details of implementations in report (5pts) + source code (10pts) ]**

Represent the set of user stories in a [vector space](https://nlp.stanford.edu/IR-book/html/htmledition/the-vector-space-model-for-scoring-1.html). To do so, follow the steps below.

* Treat each user story as a vector.
* Treat each unique token (after all text preprocessing steps) in the corpus (set of all user stories) as a vector dimension.
* Compute TF-IDF scores as the values for each vector dimension. A TF-IDF value is the product of *term frequency* and *inverse document frequency*. Use these definitions of [term frequency](https://nlp.stanford.edu/IR-book/html/htmledition/term-frequency-and-weighting-1.html) and [inverse document frequency](https://nlp.stanford.edu/IR-book/html/htmledition/inverse-document-frequency-1.html).

### Deliverables

* You must implement the project in Python.
* You must use  [NLTK](http://www.nltk.org/) (Python) for text preprocessing.
* You can use any tool for visualizing the data and getting familiar with it.
* Important: You are being asked to implement TF-IDF computation from scratch. Del
* A professionally report in PDF format, providing instructions on how to run your project and describing the techniques you implemented. [ Correct grammar and without spelling mistakes (2pts)]
* The complete source code of your project.