**Overview**

* Goal: document clustering of subreddits based on subreddit descriptions, comments, and purchasing history
  + Make a recommendation system for users of a subreddit: find others that might be interesting
* TWO desired outcomes: recommend subreddits based on browsing history AND recommending text posts within subreddits (maybe for resolving hobby questions)
  + BIG theme is hobbies/interests
  + I can also go back and validate how well this model works by manually assigning labels
* OR we could turn this into a classification task: label hobby subreddits and scrape comments, etc. and classify with manual labels

**Data Collection**

* Dataset 1: <https://www.kaggle.com/residentmario/things-on-reddit>
  + Top products for each subreddit
  + Take product titles and categories as input for word vectors
  + Idea: similar subreddits will have similar purchasing interests
* Data collection:
  + Reddit’s API: <https://praw.readthedocs.io/en/latest/getting_started/quick_start.html>
  + Scrape subreddit descriptions, titles, and comments
    - Comments: top 10 posts on the subreddit
    - Subreddits: 10 largest or the most interesting ones (could find hobby subreddits like gaming, sports, history, science, etc. and manually label)

**Word representation and similarity**

* Bag of words: each word assumed to appear independently, and order doesn’t matter
  + Make into a vector with dimensions being words and weights being frequencies
  + Weights can be counts or TF-IDF weights
* Cosine similarity: cosine of the angle between document vectors
  + Vectors are all unit length to avoid long documents dominating
* Text processing:
  + Remove stop words
  + Lemmatization or stemming
  + Feature selection: which words to include in bag of words
* Metrics
  + Euclidian distance: used in k-means (DISTANCE)
  + Cosine similarity (SIMILARITY)
    - Independent of document length
  + Jaccard coefficient (SIMILARITY)
  + Correlation coefficient (SIMILARITY)
  + Averaged Kullback-Leibler Divergence: treat BOW as probability distributions and compute differences (DISTANCE)
* Clustering algorithms:
  + Hierarchical
  + K-means
    - Standard and bisecting