# MA3831 Assessment 2 debrief

* Assessment 2 (40%) – NLP Mapping
  + Recap Recommendation system
  + Notes on dataset
  + Dataset extension
  + Tasks / deliverables
  + Expected solution
    - Approach
    - Input
    - Output
    - Some ideas
  + Marking Rubric

## Recommender System

**Content based algorithms.**

• Based on deriving the content of the item. try to find *look alike* items and recommend them.

• Context level information is easier to get when the product/item explained with few dimensions

• TF-IDF score for text. The higher the TF\*IDF score (weight), the rarer the term and vice versa.

**Collaborative filtering algorithms (User = course, item = resource)**

* not dependent on any additional information. (only transaction level information)
* User-User Collaborative filtering (User-Based KNN)
  + find look alike customer to every customer and offer products which first customer’s look alike has chosen in past
  + This algorithm is very effective but not scalable since it requires to compute every customer pair information
* Item-Item Collaborative filtering (Item-Based KNN)
  + finding item look alike
  + having item look alike matrix, recommend alike items to customer who have purchased any item from the store
  + less resource consuming than user-user collaborative filtering
* Other simpler algorithms: other approaches like market basket analysis, do not have high predictive power.

## Three use cases

* Recommend existing course material to similar subjects, or
* Recommend new reading material to existing subjects, or \*\*\*
* Provide a complete reading list of existing readings for a new subject.

## System: Input and output

* Input: a set of keywords
  + eg, computational biology, ethics in artificial intelligence
* Output: a list of recommended readings (Top N) sorted according to some relevance score
  + Book: <title>*<isbn/issn/doi><publisher><author?>*(77%)
  + Journal:<title>*<isbn/issn/doi><publisher><editor>*<(76%)
  + Journal:<title><isbn/issn/doi><publisher><editor>(75%)
* How to measure accuracy of the system?
  + Confusion matrix (sensitivity, specificity, F1-score)
  + Acceptance of the provided output (user study)
* Like any ML problem
  + 80% training data 20% test data

## Example approach: recommendation system

* Statistical approach
  + Most frequent words: course vs title
* User approach vs product approach
  + Consider ‘coursename’ as user and ‘combined\_title’ as product.
* Build course profile.
  + Use own taxonomy or FOE
  + Grouping, labelling
  + Pull data from heimshelp
* Build title profile.
* Associate each title to many user profiles.
  + With a score
  + This is the knowledgebase.

## Expected solution: recommendation system

* How it works
  + Input: transform input query into a new ‘user’ like the user profile vector. In this case this is a new user (course)
* Content based filtering
  + Recommend resources used by similar courses to this new course
* Collaborative filtering
  + User – User: new user to similar existing users
  + Item – Item: new item to recommended existing user (out of scope)

## Expected solution:

* Preprocessing: stemming, lemmatisation
* Parsing -> NP
* N-gram analysis -> dictionary approach
* Text similarity: Word2Vec embedding
* Vectorisation: BOW, TF-IDF
* Clustering
* Association rule: {bow} => {bow}
* ANN: bow for courses => bow for titles

Recommendation engines:

* 1:
  + Bag of words
* 2:
  + Bag of bi-grams
  + Lemmatize
  + Stop word removal
  + \* Removing non-english words