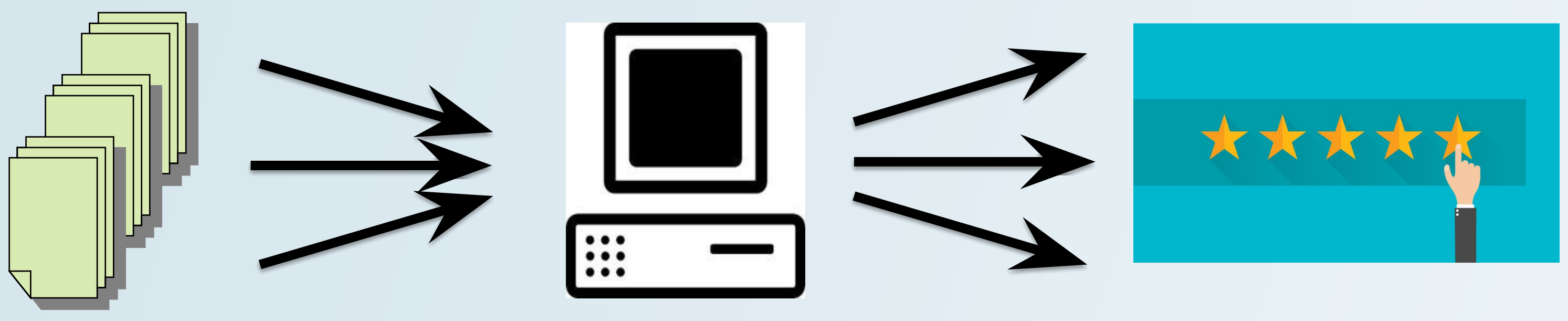
Course Review Helpfulness Classifier

Dakyung Song, Tong Pow



- **takes raw course reviews**
- **identifies criteria for what is helpful in comments according to the academic department they belong to**
- **returns rank of courses reviews according to their helpfulness**

Developed by: **Tong Pow, Dakyung Song**

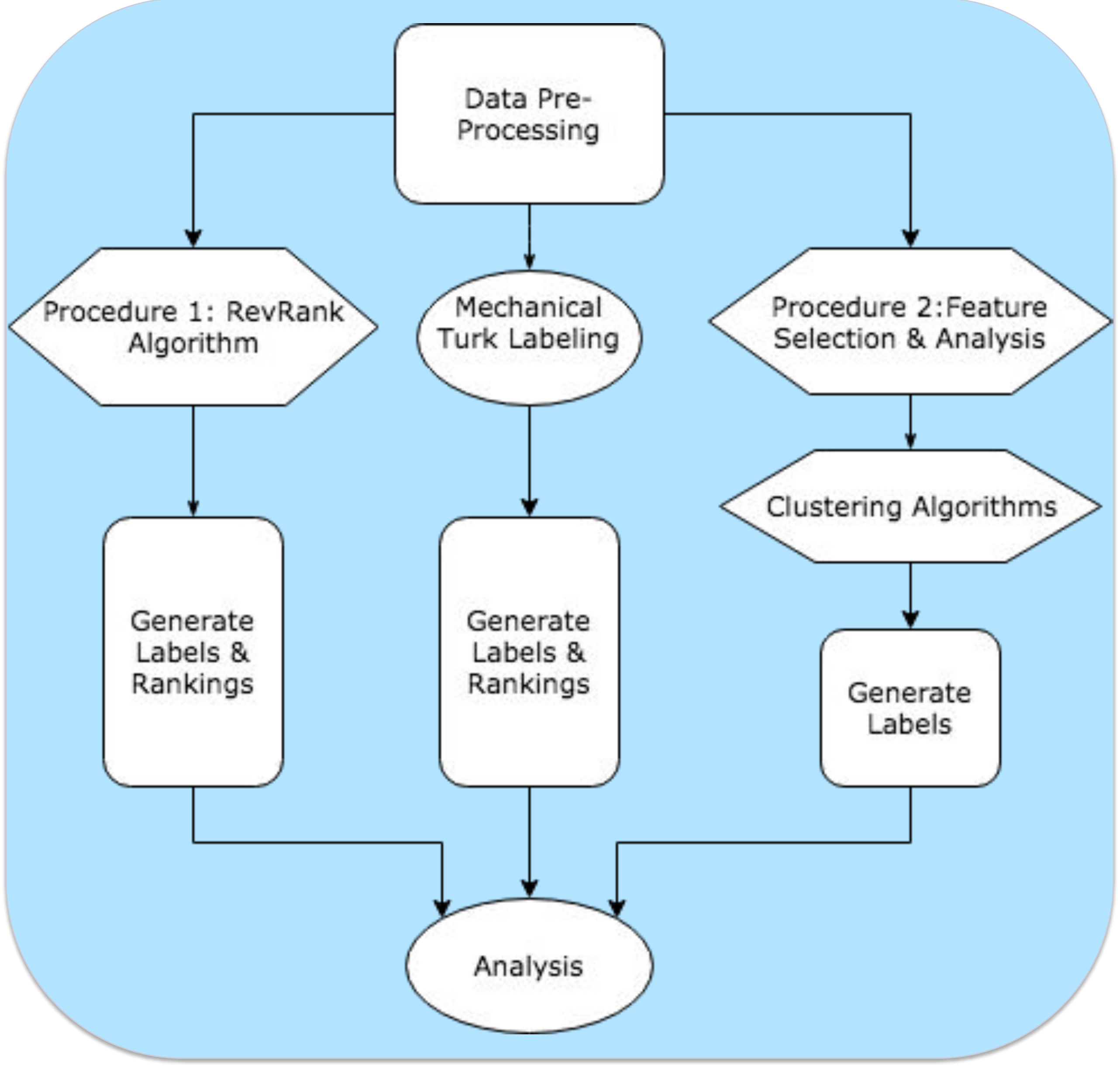
The Challenge

- There are many course reviews submitted each semester. It takes much time and resources to manually select the course reviews that are the most helpful and fit to display on the Penn course review website.
- Opinions about course reviews are often subjective.
- We address these problems by (i) annotating a subset of reviews according to a consistent criteria for helpful comments, and (ii) implement known supervised and unsupervised algorithms for this problem through (ii) training supervised models or testing unsupervised models using those annotations.
- Through feature engineering and experimenting with different models, we choose the most accurate combination to robust review helpfulness prediction model.

Our Procedure

- ❏ Clean, preprocess, & featurize our dataset(s) using feature selection
- ❏ Use Mechanical Turk to generate annotations for a subset of our dataset
- ❏ Implement and run RevRank and KMeans to label and rank comments
- ❏ Calculate statistics to analyze performance of algorithms
- ❏ Analyze the results of algorithms to determine any significant differentiating features

Procedure Flowchart


Data Pre-Processing
Procedure 1: RevRank Algorithm
Mechanical Turk Labeling
Procedure 2:Feature Selection & Analysis
Clustering Algorithms
Generate Labels & Rankings
Generate Labels & Rankings
Generate Labels
Analysis

MTurk Inter-Rater Reliability

Inter-rater reliability measures how much consensus there is in the labels given by the labelers. It is a crucial aspect of a well-designed experiment. We had two Mturk workers provide two labels to each of 105 comments: binary helpfulness index and comment helpfulness ranking by class. The former is labeled as 1 for helpful and 0 for unhelpful, while the latter is ranked 1, 2, …, n where 1 is the most helpful and n is least helpful (n = number of comments per class). We calculated two agreement indices: one for binary helpfulness index and one for comment helpfulness ranking by class:

Binary helpfulness index: 73.3%

Comment helpfulness ranking by class: 34.3%

While the second seems a bit low, it isn't as indicative of inter-rater reliability because one disagreement in ranking would automatically lead to multiple disagreements. Binary helpfulness index is a more direct and reliable measure.

Procedure 1 Results (Procedure 2 results in progress)

- An example of an helpful comment is

I think that the grading part of the course could have been changed. I enjoyed the lectures very much and believed that it was hard to decipher the knowledge learned in the course just by two tests. A group project or an essay would have been a great way of understanding the material better. I also believe that the book was outdated and did not help learning.

- An example of a slightly less helpful comment is

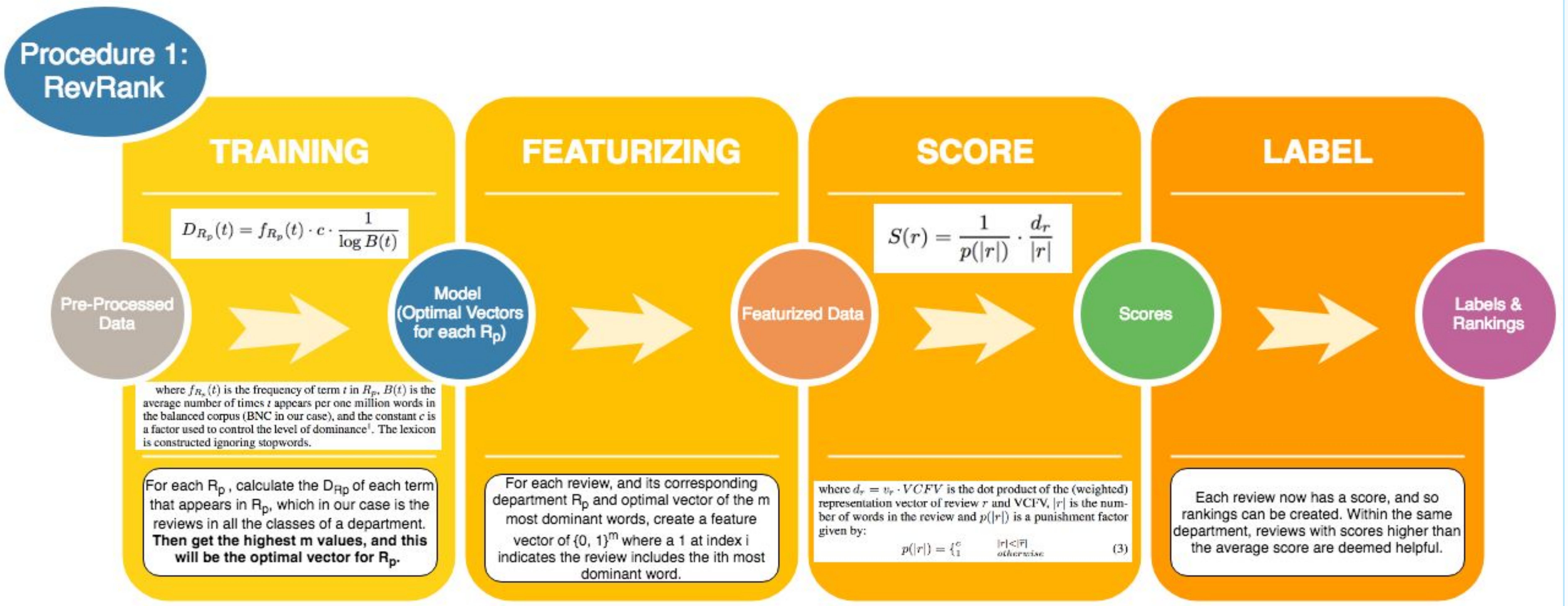
The course was, overall, disorganized. The most to be gained was from a few of the excellent guest lecturers. Otherwise, the importance of community nursing and its integral role in preventative healthcare was not well conveyed. The clinical sites also did not help to demonstrate what community nursing looks like.

- An example of a non-helpful comment is

Best nursing clinical I have had so far!

- An example of a slightly less non-helpful comment is

I feel as though multiple choice/in class exams would be better to help further solidify the course content.



$$D_{R_p}(t) = f_{R_p}(t) \cdot c \cdot \frac{1}{\log B(t)}$$

where $f_{R_p}(t)$ is the frequency of term t in R_p, $B(t)$ is the average number of times t appears per one million words in the balanced corpus (BNC in our case), and the constant c is a factor used to control the level of dominance[1]. The lexicon is constructed ignoring stopwords.

For each R_p , calculate the D_{Rp} of each term that appears in R_p, which in our case is the reviews in all the classes of a department. **Then get the highest m values, and this will be the optimal vector for R_p.**

For each review, and its corresponding department R_p and optimal vector of the m most dominant words, create a feature vector of $\{0, 1\}^m$ where a 1 at index i indicates the review includes the ith most dominant word.

$$S(r) = \frac{1}{p(|r|)} \cdot \frac{d_r}{|r|}$$

where $d_r = v_r \cdot VCFV$ is the dot product of the (weighted) representation vector of review r and VCFV, $|r|$ is the number of words in the review and $p(|r|)$ is a punishment factor given by:

$$p(|r|) = \{^{c}_{1} \quad ^{|r|<|\overline{r}|}_{otherwise} \qquad (3)$$

Each review now has a score, and so rankings can be created. Within the same department, reviews with scores higher than the average score are deemed helpful.