**MSD2015 Final Project**

Jeff Hudson, Jon Campbell, Conan (Xu) Huang

**Motivating Problem**

For our project we built and tested several social recommendation algorithms using Yelp restaurant reviews in the state of Arizona. We had two main questions driving our research. First, can social network data be leveraged to improve the predicted ratings of a recommendation algorithm? Second, to what extent are friends more likely to have similar tastes? The motivation for these questions is both commercial and general interest: companies who recommend products to users and also have access to their users’ social network data, may be able to improve their recommendations, creating more value for their users. Additionally, knowing that users’ friends likely share the same tastes could help better target advertising at groups of friends rather than individuals. Finally, if people with high network centrality are found to be influential in determining ratings for their neighbors, then businesses may want to target these influential users with special promotions.

**Data**

* the goal:
  + What do you hope to learn?
    - How users tastes influence (or are influenced by) their friendship network
  + Why does it matter to you?
  + Why might it matter to others?
    - Businesses could target particularly influential/central individuals
    - If users have friends with very similar tastes, then ostensibly we could recommend friends based on combination of tastes and network structure
* how the data were collected
  + Yelp Dataset Challenge
  + python JSON to CSV converter
* some sanity checks and visualization are useful to gain a basic understanding of the data.
  + It is often necessary to clean or filter the data to deal with problematic observations—e.g., missing data, extreme outliers, etc.
  + State explicitly what design choices and assumptions you are making in this cleaning.
    - Only looking at restaurants.
    - Getting rid of users who have rated fewer than 10 restaurants
* Next, specify the modeling task—e.g., regression, classification, clustering, dimensionality reduction, recommendation system, etc.—along with the model(s) you’re considering.
  + Dimensionality reduction/recommendation/clustering/network analysis
* State explicitly why this learning task is a natural way to frame the goal. In most cases this should include a loss function that quantitatively balances model fit and model complexity, along with an algorithm for optimizing this loss function.
* Clearly define success metrics which quantify performance—e.g. accuracy, confusion matrix, ROC, etc.—and evaluate these metrics on both training and test data to assess fit and generalization.
* Be sure to address the issue of complexity control (a.k.a. “model selection”), as discussed extensively in class.
* Discuss the practical aspects of your project, including the scalability and computational complexity of the storage and runtime for the methods used.