Group Design Document

**Question:** Can our system predict the typical star rating of a particular type of business given its location in Pittsburgh?

**Data to use**:

* Business: Attributes and star rating serve as ground truth? Group businesses by location as well to calculate average stars in a location to serve as a baseline against type.
* Review + user: Overall sentiment analysis, evaluated by actual star ratings as compared to that user’s average star rating. Attributes parsed from features within the text, evaluated against business’s labeled attributes.
* Tips: Sentiment analysis, evaluated using business’s stars.

**Expected pipeline (this is kind of up in the air)**

* 2 systems? Sentiment and attributes?
* A training example is 1 review or 1 tip. Need to group by location.
* For the review, features can either be sentiment (“excellent”, “bad”) or attribute (“slow”, “good décor”) features. Probably going to be some overlap here.
* Train our systems to output a star rating based on a location and an attribute.

**What has been done so far:**

* (David) Parsed JSONs to remove a lot of unneeded fields and make them CSV. Business file reduced to contain only Pittsburgh businesses.
* (David) Reduced the latitude and longitude fields to specific neighborhoods in business file using Google Maps API.
* (Bogdan) Review preprocessing. Sentence boundary detection.
* (Tyler) Feature selection and extraction from reviews. Mix of sentiment and features
* (Tyler) Parsed reviews file to only contain reviews on Pittsburgh-based businesses
* (Xinhai) who?
* (David, Tyer)Sentiment analysis of reviews + tips. Use review stars as gold standard.

**General Pipeline:**

Yelp data 🡪 ParseDataIntoCSV.java 🡪 PittsbughReviewParser.java 🡪 boundaries.java 🡪 NLTKTest.py 🡪 NLTKTestAfter.py 🡪 NormalizeSentiment.py 🡪 BusinessCalcScore.py 🡪 Evaluation.py 🡪 results

any set of reviews 🡪 FeatExtractor.py 🡪 review features

**Script Descriptions:**

1. PittsburghReviewParser.java

Input: entire review data file, Pittsburgh business data file

Output: file containing all Pittsburgh review dataNormalizeSentiment.py

Input: review file with sentiment scores, user data file

Output: review file with normalized sentiment scores

Description: normalizes sentiment scores based on user average rating

1. BusinessCalcScore.java

Input: review file with normalized sentiment scores, Pittsburgh business data file

Output: business data file with average sentiment rating

1. Evaluation.py

Input: business file with average sentiment rating

Output: file containing results

Description: analyzes average sentiment scores with actual average scores and outputs results. It calculates strict and relaxed results. The relaxed results are: if sentiment score is within .5 stars of original, if sentiment score is within 1 star, and (less important) if sentiment score was within 1.5 stars