**Progress Report**

**RMatcher: Recommendation Engine on the Yelp Dataset**

Ameen Askar - Joon-Sub Chung

[**https://github.com/rmatcher**](https://github.com/rmatcher)

**What We Have Accomplished:**

Since our proposal we have made several initial headways into our recommendation engine. First, we wrote java code to parse the Yelp Dataset into a MySQL database. We wrote SQL scripts that randomly select reviews as a testing set from users with low, medium, and high reviewing activity. We then implemented Pearson correlation for a given user with relevant users. From the most correlated users, we return the average stars of businesses they have reviewed which will be the basis of the recommendation engine. The targeted user testing set’s ratings are compared with the average ratings from the top correlated users. The correlation-only results did not yield great results. The predicted ratings (i.e. averaged from other users) did not agree with the targeted user’s ratings for half of the testing reviews. Therefore, we included Euclidean distance between users based on their ratings. The distance is used in addition to the correlation score to pick the top users. Moreover, the count of ratings being averaged is used as a confidence level because the majority of discrepancies were a result of a single user’s rating. Another progress on the project is the sentiment analysis. Stanford POS tagger is used to identify the adjectives which are scored using SentiWordNet. Therefore, given a review, we can give a naive score of how positive or negative it is. The sentiment score will be used as enrichment to the stars rating.

**Changes of initial plan:**

We have realized that our benchmarks of precision and recall may not capture the results well because users have a small number of reviews on average. Therefore, we have decided to also capture the accuracy of predicted ratings using the mean squared error. The validity rate which was going to be judged subjectively won’t be used anymore since the above benchmarks will be sufficient.

Although this was not in the proposal, we have set up a skeleton code for a java webapp to output the recommendations to a web page. However, we realized that this is outside the scope of the project and we should rather focus on boosting the results.

**What still needs to be done:**

We have been meeting our proposed milestones on time. We still need to add a mechanism to weight users and reviews differently. Also, we need to incorporate the sentiment analysis scores to the ratings and see if it improves the recommendation results. Finally, we need to add business to business correlation using categories and check-ins data to enhance the results, especially for users with low number of reviews.

**Future Milestones:**

5/17 Optimize recommendations by: adding user weighting, incorporating sentiment scoring, and adding business to business correlation.

5/24     Cleanup code and collect results – Write-up Final Report & Presentation.