**\*\*Summary\*\***

Initially, we wanted to simply recreate the results from the paper listed below. After discussing with Kluver, just replication here was not enough for our desired grade. So, we decided to take things a step further and do some analysis on whether including personality in a recommendation has any impact on model performance. We have personality data for a lot of users, so we can do analysis on the tendencies of ratings based on personality, popularity level, genre, and rating level, and bake those tendencies into the predicted rating. For example, let’s say we have a base user-user model, and for a given user we give them a predicted rating of 4. However, we see that this user has a high openness trait, the movie has a high popularity level, low rating level, and the movie’s genres were adventure and action. For this type of movie, and for this type of user, we see that the rating is 0.1 higher than average. So, we can add this into our predicted rating to make it 4.1 and see if that improves RMSE. To answer this question, we created 3 models. A simple user-user, a user-user with an offset based on uniformly weighted personality traits, and a user-user with an offset based on learned a personality weight via gradient descent. We must account for the case where a user belongs to multiple high/low personality groups, so that is where the uniformly weighted vs gradient descent weights conversation comes into play.

Project Team (list all members):

1. Mason Greseth
2. Don Pham

DOI/URL for Paper:

* <https://doi.org/10.1145/2959100.2959140>

Checklist for Replication Assignment Grading

\_\_\_ Basic Replication

* Was code available?
  + The code was not available. We used a plethora of colab notebooks for data preprocessing and model creation and evaluating. The tools we used include pandas, NumPy, surprise, and JSON.
* Was data available? (yes/no)
  + Yes, the original data included the following files:
    - Ratings.csv – MovieLens dataset with ~1,000,000 ratings
    - Personality.csv – Results from the personality survey from 1840 users (contains score 1-7 for all 5 traits for each user)
    - Movies.csv – Genre information for each movie
  + Data preprocessing
    - One hot encoding for each genre within movie dataset
    - Used JSON files to store dictionaries that represented data that was being queried often to reduce lookup times.
      * User ID’s to personality trait levels
      * Genre Averages per trait level
      * Movie ID’s to movie information such as rating level, popularity level and a set of genres
    - Remove ratings for movies that did not exist in the movie dataset.
* Did basic replication produce expected results? (yes/no)
  + For simple replication, no tuning was needed, just querying of the altered data.
* How many algorithms, datasets, metrics were included in basic replication?
  + In basic replication, no algorithms, 3 datasets, no metrics (we know this is not a lot, check enhancements)

\_\_\_ Enhancements

* Additional datasets (list ones evaluated)
  + JSON files:
    - userIdToPersonalityTraitDict.json
      * User ID to personality traits
    - result.json
      * Personality traits and genres to average rating
    - movieDetails.json
      * Movie ID to movie rating, popularity, genres, and average score
* Additional algorithms or variants (list ones evaluated)
  + Simple user-user, user-user with uniform weights, user-user with gradient descent
* Parameter exploration / tuning (list)
  + Minimizing RMSE with gradient descent
* Additional metrics (list ones included)
  + RMSE for the 3 different algorithms
* Additional forms of complexity added or encountered

\_\_\_ Written in form of a Replication Paper (yes/no) No, one final colab notebook

\_\_\_ Other factors that should be taken into account

See summary