Model description:

I used the principles of model-based recommendation systems, XGBregressor(a regressor based on the decision tree) to train the model.

There were total of 5 files that were given:

a. train_review.json
b. user.json – user metadata
c. business.json – business metadata, including locations, attributes, and categories
d. user_avg.json – containing the average stars for the users in the train dataset
e. business_avg.json – containing the average stars for the businesses in the train dataset

Attribute selection:

I used these files to extract features required for the classification. From the user.json file I extracted features such as compliment_photos, fans, funny and important among this was the friends list that I could use to analyze the ratings made by a user's friends on a business. The mean and mode of the friend rating is being used to train the model.

From the business.json file, I extracted attributes such as is_open and category of business, postal code and number of days the business is open. I stored this information in the business_dict data structure.

Similarly I extracted the information from the user_avg.json and business_avg.json files.

For every (user_id, business_id) pairs, I extracted the feature attribute information, tabulated them in lists and collected the actual star rating given by the user on a business. I also calculated the average rating by all the friends of a user.

Hyperparameters selection:

To find the optimal hyper parameters that will give the best results, I ran GridSearchCV on a set of parameters to find the best parameter values for the XGBregressor. Some parameters that were searched was learning_rate, max_depth, gamma, min_child_weight and colsample_bytree. These parameters must be optimal to avoid overfitting.
I divided my dataset into 2 parts one having the values of user-average ratings and the other one which is null. I generated 2 model files.

Prediction:

For every (user_id, business_id) pairs, I extracted the feature attribute information, tabulated them in lists. If the user was a new user or business was a new business, no features were added. It was then passed to the xgb predict function which provided the results. This information was written to the output file.