# Help Yelp Predict The Rating

## I. Group Information

**Group Name:** The Grand Truthers (same name on Kaggle)
**Group Number**: 18
**Group Members:** Omid Eghbali, Hengda Shi, Prabhjot Singh, Kanisha Shah, Garvit Pugalia

## II. Problem Definition and Formulation

Yelp is an easily-accessible, online review tool, which allows customers to rate businesses and provide feedback. It has collected over 135 million reviews worldwide, forming a huge database of customer information, business information, and their interrelationships. With such a large amount of data, there is potential for data mining and analysis.

The goal of the project is to access a sample of the Yelp review dataset, and design a predictor that can forecast a customer's rating of a business based on the customer's history and the business' previous reviews. The type of data available to the predictor includes business' rating and number of reviews, customer's average ratings, and a large bank of existing reviews. In our approach, the data will be processed first to retrieve the relevant features before fitting to our model. After training the model on a sample of the dataset, the RMSE of the prediction will be computed to tweak and optimize the model.

This problem can be formalized as:
*Given a previously unseen {customer_id, business_id} pair, with given attributes for each customer and business, predict a rating from [1, 2, 3, 4, 5] for the customer's review of the business.*

## III. Proposed Methods

### Tasks

In order to tackle this large data mining problem, we have set a plan to break down the problem into manageable units. Firstly, we start with data visualization in which we make histograms of the different features and look at the frequency of the ratings that are given. This has already proven to be tremendously helpful in our task of feature selection. Next, we move onto preprocessing the data sets, which is further described below. We plan to employ a forward selection approach of the features, since training time using all the features to start would take up a lot of our time. Finally, we move onto training our different models of interest, more about this can be seen in the relevant section below. The best model will be chosen based on the lowest RMSE we find, and will be reported on accordingly in the final report.

### Data Preparation and Preprocessing

Data preprocessing has been our largest focus over the past couple of weeks. Given how our features are spread out across multiple files, we have been using pandas to convert the data into a more manageable form which we can then feed into our models. Further, we have implemented what is it needed to normalize feature values to make comparisons between the normalized and un-normalized dataset. Using what we have so far we can train basic models, but to improve on accuracy more work is needed to clean the data further and finalize feature choices.

**Proposed Algorithms**

A well-known machine learning library, *scikit-learn*, will be used frequently. It helps users analyze large datasets and form classification/regression/clustering models from the same. Since it is built on NumPy and matplotlib, it is highly compatible with the other preprocessing and visualization tools used in the project. Some of the models, provided by *scikit-learn*, that we propose to use for the predictor include, at the very least:

**Linear Regression**: We can view the rating [1, 5] as a continuous variable, predicted by business and customer information, and then adjust to a class value.

**Multiclass Logistic Regression**: The algorithm is an extension of Logistic Regression to multiple classes. It is an attractive option as it doesn't assume linearity or normality of the dataset. (Starkweather, 2011)

**Linear SVM**: Known as one of the best classifiers, SVMs can be made highly accurate, robust and efficient (with a low RMSE) using kernel tricks and dimensionality reduction. (Aly, 2005)

**Random Forest**: Given the large scale of our dataset, notoriety of random forest for being good at classification tasks, and resilience to overfitting (Donges, 2018), this model will likely be key in getting a lower RMSE.

## IV. Experiments design and Evaluation

The Root Mean Squared Error (RMSE) can be calculated with the following equation:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2}{N}}$$

The RMSE describes the standard deviation in the differences between the predicted and real values. There are a few benefits from its alternative, MSE, that need to be taken into consideration.

A. The RMSE brings the error back into the units of the data (i.e. value on the y-axis), whereas MSE is in the square units of the data. Therefore, the RMSE is a more easily interpretable and accurate measure of the goodness of the fit.
B. The RMSE normalizes by the amount of training data provided, therefore, a relatively large sample of the Yelp dataset should be used.

We will select the model with the lowest, most consistent RMSE error, out of all our Models.
**Current Kaggle Score**: 1.20185

## V. Discussion and Findings

Given the number of features we are provided, we have definitely observed that feature selection is going to take a lot more work over the next week. In order to gain coherent results, it is necessary that we improve on our chosen features. Also, after initial data processing, we realized that there are possible unique customers and businesses in the validation set. Therefore, more advanced techniques with user correlation will be added to our preliminary model. As of now, we have used the business average stars and user average stars as the two features, so we can get a feel for the necessary complexity and confirm our models work as intended.

Currently, we have tested our first version of the preprocessed data, using only the two aforementioned features, to build logistic regression, linear svm, and random forest models. We observed that with linear svm, the time taken to complete the model scales heavily with the number of features included. As such we will have to be careful with adding more features while trying to lower our RMSE. As of now our linear SVM and logistic regression both get an RMSE of roughly 1.2, which is a satisfying start but must be improved. Surprisingly, our first run of random forest had the worst RMSE out of everything tried thus far. We fully expect these results to change in the coming weeks, as more features are selected and model tuning is done. For example, going forward, we plan on trying SVM using different kernels and implementing more models within our capability.

## VI. Schedule

| | Tasks | Members |
|---|---|---|
| **Week 7** | Clean up / Finalize Preprocessing Code | Hengda |
| | Test current Linear SVM/Logistic Reg/Random Forest model further | Hengda,Omid,Garvit |
| | Continue feature selection and model tuning | Omid,Garvit |
| **Week 8** | Linear Regression Model Implementation, Further model tuning | Prabhjot, Kanisha |
| | Start researching what we can do with review contents | ALL |
| | Finalize Model tweaking / features chosen for each | Hengda,Omid,Garvit |
| **Week 9** | Look further into review contents after learning naive bayes | ALL |
| | Try more advanced correlation tools to deal with unique user issue | Hengda,Kanisha |
| | Begin outputting graphs for report, writing report | Prabhjot,Omid,Garvit |
| **Week 10** | Finish Report and Submit final submission(s) to Kaggle early week 10 | ALL |

## VII. References

Aly, Mohamed. *Survey on Multiclass Classification Methods*. University of Utah, 2005. https://www.cs.utah.edu/~piyush/teaching/aly05multiclass.pdf.

Starkweather, John, et al. *Multinomial Logistic Regression*, 2011. https://it.unt.edu/sites/default/files/mlr_jds_aug2011.pdf.

Donges, Niklas. “The Random Forest Algorithm – Towards Data Science.” *Towards Data Science*, Towards Data Science, 22 Feb. 2018, towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd.