Alphas and Alohas: Modeling Google Local Ratings in Hawaii

Eric Pham, University of California San Diego
Kenneth Tran, University of California San Diego

Figure 1: Waikiki Beach in Honolulu, Hawaii.

Exploratory Data Analysis

For our project, we picked the 2021 Google reviews dataset. We specifically chose the Hawaii 10-core review dataset (including the metadata), which means that each user wrote 10 reviews and each business has been reviewed 10 times.

We chose the state of Hawaii because the sample size was not too large (~1500000 compared to California's 40000000 reviews) and is simultaneously dense (Hawaii is also a cool state).

But before we could take any steps towards building a recommender system on this data, we had to do some preprocessing. Each business is associated with an "gmap_id", which corresponds to an entry in the metadata set. Taking advantage of this, we constructed a pandas dataframe, joining both the review data and the metadata on the "gmap_id" column. This would allow us to explore the review data as well as the features much more easily.

Once the data was added to a dataframe, we used the "address" feature to extract the "zip code" and the "city" of the business to add as additional features. We also used the "time" feature to decompose the datetime data into a "year" feature which shows the year the review was written as a float. Now that the data is properly preprocessed and cleaned, we began performing analysis.

Some surface level exploration revealed that there is much to be said about missingness in the data. Though it includes "text" and "price" as a feature, less than half of the reviewers left any text and many businesses did not have a "price" feature associated with it.

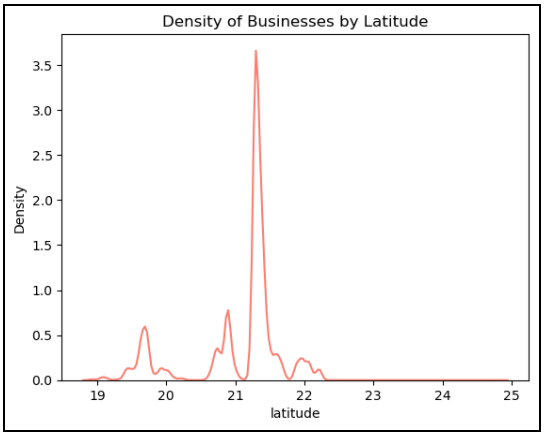

Figure 2: The four spikes well represent each of the four islands.

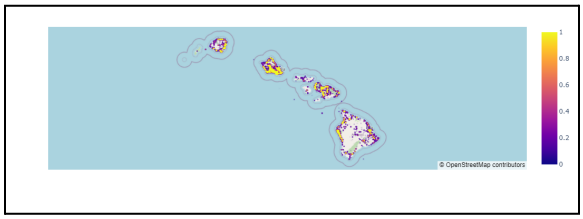

Figure 3: Density of businesses by island.

Since Hawaii is a set of islands, each with their own community, this begs the question whether or not users on different islands have different rating behaviors. We thought we would have to go through the trouble of encoding a new feature that represents which island a business is on, but after looking at average rating by city, it was clear that geography was not a big factor in determining rating in this dataset. The dataset was dominated by reviews written for Honolulu, which is Hawaii's largest city. But even other cities did not stray too far from the average rating.

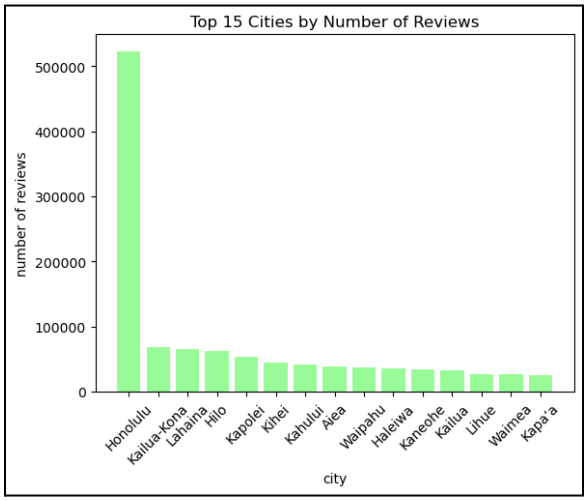

Figure 4: Honolulu is Hawaii's largest city by far, which is why the data is skewed towards that city in particular.

Based on this data, it seems that large denominations of geography like city and island are not useful features when it comes to predicting ratings. We reasoned that we are better off leaving these features out of our model since they don't encode much information about either the business or the user. We then turned our attention to other features in the dataset.

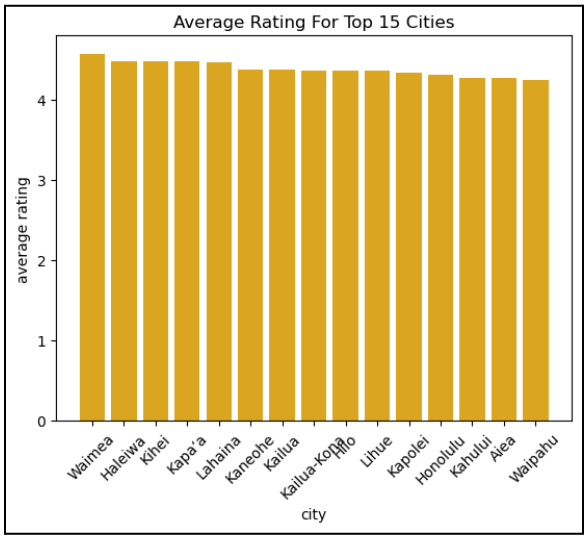

Figure 5: The four spikes well represents each of the four islands, which means latitude may be a good feature.

Some of the more interesting findings include a weak association of average rating by zip code and an upward trend of rating over time.

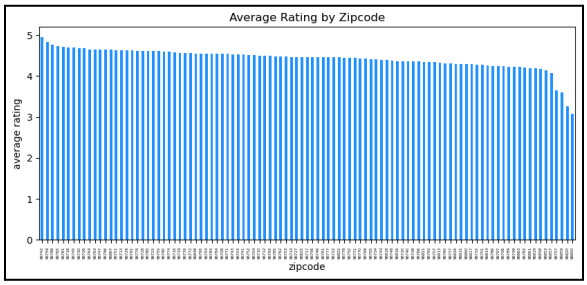

Figure 6: The range between average rating by zip code is comparatively high.

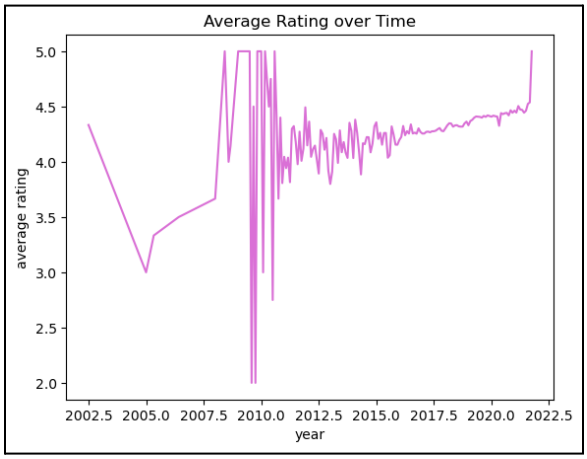

Figure 7: Review data becomes more frequent as time goes on and ratings also trend upwards.

We thought that time would be a good feature at first since ratings do change over time, however after some careful consideration, we decided to scrap the idea, with the reason being that the time series data showed that ratings were very infrequent early on and became more dense as time progressed. This means that our data is very messy early on and might lead to poor model performance.

The only feature we really felt we could use as a good predictor of rating was zip code, and even then we thought that it would only marginally help.

Predictive Task

Our project aims to predict user ratings for a particular business, given the user's rating history, location ratings, and other metadata features. Something of interest to note is that while in practice, ratings are integer based (e.g. 1 star, 4 stars), since we mainly chose a regression based model, our model will predict floats.

When it came time to decide on a metric of evaluation, we settled for mean-squared error since it is a good evaluation of regression type problems. Knowing this, we defined a baseline model, which predicts the mean rating of the training set for each set of user & business rating pairs. This is a simple enough model that will allow us to easily compare across different models.

For some of our models, we decided that we would try to incorporate zip code as a feature since we saw decent correlation between zip code and rating. For our data, Google Local's 1,504,346 ratings, the training, validation, and testing sets are split at approximately 80/10/10.

Model Selection

The models considered were: the baseline, linear regression, linear, latent factor, and factorized machine. Ranked in order of performance:

Linear Regression Model

Model: $f(x) = X \cdot \theta$.
Features: zip code, average rating by business.
Performance: $MSE = 0.9027621222041154$.
Strength: Accounts for features.
Weakness: Fails to capture interaction.

Baseline Model

Model: $f(u, i) = \alpha$.
Features: none.
Performance: $MSE = 0.8509562841591932$.
Strength: Simple and efficient since most reviews are either 4 or 5.
Weakness: Fails to capture interaction.

Factorized Machine Model

Model: $f(x) = w_o + \sum_{i=1}^{F} w_i x_i + \sum_{i=1}^{F} \sum_{j=i+1}^{F} [\gamma_i, \gamma_j] x_i x_j$.
Features: user_id, gmap_id, zip code.
Performance: $MSE = 0.8420558700788963$.
Strength: Allows for both feature and interaction capture.
Weakness: Difficult to understand and train, not scalable.

Latent Factor Model

Model: $f(u, i) = \alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i$.
Features: user_id, gmap_id.
Performance: $MSE = 0.8131820971621532$.
Strength: Allows for discovery of hidden dimensions.
Weakness: Cold-start may lead to poor model performance.

Linear Model

Model: $f(u, i) = \alpha + \beta_u + \beta_i$.
Features: user_id, gmap_id.
Performance: $MSE = 0.6264581116074759$.
Strength: captures user business interaction in a simple manner.
Weakness: Perhaps too simple in the way it models interaction data.

We thought that including features in models would allow for better performance but it seems that models that incorporated features and not just user, business interaction fared poorly compared to other models.

An important thing to note is that factorized machines did not scale very well, and we had to cut down our data from ~1500000 to ~15000. Only after sizing down our data by 99% were we able to train the factorization machine. While its results were not the best, it may have performed better if trained on a larger dataset.

Evaluating all model performances, we determined that the best model for our dataset was the linear model, which simply takes into account a set of biases for each user and business when predicting ratings. All models aside, the linear model performed the best because of the density of the dataset. By modeling user rating tendency and typical business ratings, we were able to simply capture interaction patterns. The dataset is dense enough that this yielded the best measures since there is plenty of data for each user and business.

Literature Review

Rating predictions has been an important predictive task for many recommender systems. In a larger context, it provides insight on whether or not an individual would like something and to what extent. Much of this project was influenced by the Netflix Prize competition, in which teams were given only interaction data of users.

.

Koren, Yehuda. "Factorization meets the neighborhood: a multifaceted collaborative filtering model." Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. 2008.

Koren, from the first winning team in the Netflix Prize competition, describes his team's approach in combining collaborative filtering and latent model techniques to capitalize on available explicit feedback, while also integrating the more abundant implicit feedback data.

Koren, Yehuda. "The bellkor solution to the netflix grand prize." Netflix prize documentation 81.2009 (2009): 1-10.

In the final solution that ultimately won the Netflix Prize, Koren elaborated on the combination of techniques in his team's model which lead to better predictions given the Netflix data. Notable changes from his previous publication on the Netflix Prize was the inclusion of temporal variability. One such phenomenon observed was "frequency" which describes how many movies a user rates in a short period of time. This would overall skew that user's ratings for certain movies as well as the movies' overall ratings.

Yan, An, et al. "Personalized Showcases: Generating multi-modal explanations for recommendations." Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2023.

In this study, Yan, He, Li, Zhang, and McAuley use the larger Google Local data set to provide personalized textual and visual explanations of businesses for users. While this predictive task differs from the traditional rating predictions, it is also unique in the way that the model attempts to generalize businesses to users as accurately and concisely as possible. Images take less time to process for humans but also tell much more than words do.

Chang, Biao, et al. "Predicting the popularity of online serials with autoregressive models." Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. 2014.

This study uses autoregressive models to predict the popularity of serials (e.g. TV shows, multi-part movies), whose popularities spontaneously change due to a number of temporal factors. The model described in this paper focuses on sequence, release dates, and update regularity as the main features of the model.

Lin, Chenxi, et al. "Using probabilistic latent semantic analysis for personalized web search." Web Technologies Research and Development-APWeb 2005: 7th Asia-Pacific Web Conference, Shanghai, China, March 29-April 1, 2005. Proceedings 7. Springer Berlin Heidelberg, 2005.

This early study of integrating user behavior into curating a personalized search query uses probabilistic latent semantic analysis, which is similar to latent factor models, which identify features based on user interactions with other items. One setback, however, is the amount of memory needed to map out every user and page interaction. The model proposed by this paper suggests an algorithm to compress and generalize similar groups of users.

Results

After careful and rigorous testing we determined that our Linear Model is most fit in predicting rating data. This model performs significantly better than the other four models with an MSE that is about 20-30% lower than the other models.

The density of the dataset is certainly a factor that allowed this model to succeed. Since users and businesses are well represented in the dataset, "learning" a bias for each of them was not difficult since there was so much data to work with. 10 reviews is plenty of data for a good bias to be determined, which is why it far outperformed the other models at our disposal.

Although we experimented with zip code representations in our features, we just didn't seem to get any noticeable results. In our linear regression model, adding a feature of the average rating by zip code barely improved the MSE. It's also difficult to determine the impact of zip code as a feature in the factorization machine since we were not able to scale that many dimensions on the whole dataset.

Linear Model

Model: $f(u, i) = \alpha + \beta_u + \beta_i$.

Features: user_id, gmap_id.

Performance: $MSE = 0.6264581116074759$.

With the above model, β_u represents typical user ratings, while β_i represents typical business ratings. In conjunction, these two biases plus a regularization term were enough to outperform many of the more complex models. Using only 'user_id' and 'gmap_id' as our features is sufficient enough to capture rating tendencies, which once again shows that with fine tuning, a simpler model can outperform more complex models.

While our linear model yielded great results, one model we overlooked was the collaborative filtering model. We think that this could potentially have even better results since it allows us to use similarity between users and items to our advantage. Autoregression might have also helped with predicting ratings for more recent data, since more recent ratings got more consistent with time, and it would be more practical for predicting current and future ratings.

With our model finally selected, we feel confident that the linear model is the best choice for predicting rating data in the "Aloha State", and we know for a fact that it will be able to tell users which shaved ice place they will enjoy the most (island vintage all the way), or which state beach they will rate the highest.