Data Science Competition: Restaurant Predictions - Final Report

Andrew Floyd and Daniel Fuchs
Missouri University of Science and Technology
1201 N State Street
Rolla, MO 65409

***Abstract*— This report contains the results for the Restaurant Rating Prediction problem, which served as the semester-long project for the course "Introduction to Data Science". Given a large set of user and restaurant information alongside users, as well as ratings given for user-restaurant pairs, the goal of this project was to predict what rating would result from new user-restaurant pairs. This project involved the use of a variety of regressions, and required a large amount of preprocessing to get the data into a trainable state; ultimately, a "Logistic Regression"-centric model performed the best on the data set, attaining an accuracy of 80.52%.**

I. INTRODUCTION

While there are many factors that go into deciding what restaurant consumers eat at, one of the most noteworthy factors that restaurant's rating. And while a high average rating may lead to person initially trying a restaurant, the chances that they return to a restaurant they then rate poorly is fairly low. Naturally, an individual is significantly more likely to repeatedly visit a restaurant they rate highly.

However, simply trying restaurants at random or merely picking restaurants by highest average rating is not an effective method to determining which types of restaurants a user would attend and then afterwards rate highly. Further, there are many greatly varying components and features from restaurant to restaurant, and an equally-greatly varying mix of preferences from person to person; the reasons one person rates some location highly may be the same reason another rates it poorly.

So then, one can't assume some restaurant's future ratings by different individuals will always have similar scores to those of previous ratings. Yet, these ratings still contain valuable information on how certain individuals value a particular restaurant's traits. With the plethora of online reviews and ratings available for countless restaurants alongside an equally diverse collection of information profiles on those particular consumers and restaurants

reviewed, one can observe some commonality in why a user may rate one restaurant above another. While deducing this information is a complex task, doing so allows for better predicting how a user might rate a restaurant in relation after considering their informational profiles.

II. PROBLEM STATEMENT

Even with the large amount of information readily available for use, the task of predicting how a user would rate a restaurant is still difficult. However, by finding a way to make use of this wealth of information, recommender systems could recommend restaurants a target user would enjoy with much greater accuracy. This would not only improve the accuracy of specific, targeted advertisements, but also benefit the users simply looking for a restaurant they would actually enjoy. So then, the goal of this project is to explore methods in predicting how a user might rate a restaurant, with consideration of informational profiles on both the user and the restaurant they are to attend.

III. GENERAL APPROACH

In approaching this problem, there are three major considerations to be made. First, one must consider how to aggregate all the available data on both the users and restaurants together. In orderly to effectively assess what factors might have lead to a particular rating, restaurants and user information profiles must be composed. These can be joined together for any given review, to provide a full picture of the review, reviewer, and reviewee.

The second major consideration is what information can be derived from common features between the user and restaurant. From the available information, one can create many composite relationships. For example, the user smoking and the restaurant allowing smoking, or the user preferring formal dress and the restaurant requiring it, or the user's budget matching the restaurant's prices, and so on. However, additional complex features and relationships can be derived from these simpler attributes.

The last major consideration is what model to approach this problem with. Many features are bound to have a significantly greater impact than others, and thus warrant a weighting system. Likewise, exploration and experimentation is necessary to determine whether a cluster-centric approach might be less preferable than a regression-centric approach, or perhaps a combination of methods is preferable to single methodology, and so on. So, much experimentation needs to be done on the performance of various models, to determine what methods are the most effective at predicting what a user's final rating might be.

IV. DATA PREPROCESSING

The first step to creating a model was creating the training set that would teach the model. To create this training set, a large table needed to be created, with one entry for each rating specified in the training data. Each entry of this table contains information on the user who gave the review, and the restaurant to which it was given. But before this common table could be created, all of the provided data files needed to be aggregated into a common informational profile for each user, and another for each restaurant.

First, the component files for generating the restaurant profiling table were gathered. The files are listed as follows:

- chefmozaccepts.csv - Contains information on which payment methods each restaurant accepts
- chefmozcuisine.csv - Contains list of cuisines offered by each restaurant
- chefmozhours4.csv - Lists operating days and hours for each restaurant
- chefmozparking.csv - Contains information on parking services provided by each restaurant
- geoplaces.csv - Contains various information on restaurant location, services, and general style

After cleaning the files, replacing missing values, and removing unnecessary features, the restaurant profile was created with the features shown in (TABLE I). Some features, such as alcohol and smoking, were converted into an ordinal ranking, while payment methods were converted into a series of boolean features. Cuisine was also passed along as a list, and later processed during feature selection. The restaurant's placeID was passed along for later use as an identifier when joining the restaurant profiles to the relevant reviews.

TABLE I
RESTAURANT PROFILE INFORMATION

Boolean Features	Continuous Features	Ordinal Features
Accepts Payment-type Free / Paid / No Parking Formal Dress Code Franchise Open Area Quiet Ambience	Latitude Longitude Weekday Hours Saturday Hours Sunday Hours	Alcohol Accessibility Pricing Bracket Service Level Smoking Accessibility

TABLE II
USERPROFILE INFORMATION

Boolean Features	Continuous Features	Nominal Features	Ordinal Features
Age Formal Dress Preference Married Prefers Quiet Ambience Smokes Uses Payment-type	Latitude Longitude Weight	Interest Personality	Budget Drinking Level Transportation Level

Next, the component files for generating the user profiling table were gathered. The files are listed as follows:

- userpayment.csv - Contains information on which payment methods each user uses
- usercuisine.csv - Contains list of cuisines preferred by each user
- userprofile.csv - Contains large amount of personal information on each user

After processing missing values and removing unnecessary information, the user informational profile was created with the features shown in (TABLE II). As with some of the restaurant features, ordinal features were derived based on magnitudes from existing information. Cuisine was again passed as a list, to be processed during feature selection. Interest and personality were left as nominal, due to the difficulty to "rank" their values.

After obtaining these two profiling tables, these values were joined against the training entries selected from "rating_final.csv", which also added ordinal features for the sub-ratings on both food and service. The entries were joined by the restaurant and user IDs, creating the desired "primary information" table to contain all relevant info for both the user and restaurant involved in each rating.

V. FEATURE SELECTION

After completing the aggregation of all data into the primary information table, many similar features were combined into "matching" features. These features were either boolean in cases of boolean matching between the user and restaurant, or treated instead as a score, where higher values indicated increasing agreement between the user's preference and the restaurant's respective accommodation. Some additional features were also synthesized from on combinations of features or were computed from existing information. The additionally-created features are listed as follows:

- Alcohol Match - True if the user's stance on drinking matched the restaurant's providing of alcohol
- Alcohol Score - Score based on how closely amount of alcohol provided matched user's preference
- Average Hours - Average length of time location was open across all days in operation
- Cuisine Score - Based on overlap between user's preferred dishes and restaurant's menu
- Days Open - Number of days a restaurant operated
- Dress Match - True if user's preference of formal dress matched restaurant's dress policy
- Noise Match - True if the environment matched the activity / noise levels preferred by the user
- Parking Score - Based on how well parking setup matched user's level of transportation
- Payment Score - Based on how many of the user's payment methods the restaurant accepted
- Price Score - Based on how well user's budgets matched the restaurant's prices
- Proximity - A logarithmic score indicative of how close the restaurant was the to user
- Smoker - If the user was a smoker, regardless of restaurant's policy
- Subrating - Average of service rating and food rating provided for by user for some restaurant
- Smoking Score - Score based on how closely smoking accommodations matched user's smoking habits

After all these features were generated, the process of feature selection began. After extensive experimentation and pruning, only highly-correlative and useful features were retained; the rest were removed. See (TABLE III) for a full list of all the features selected for the training and applying the model.

TABLE III
FINALIZED LIST OF SELECTED FEATURES

User Features	Restaurant Features	Extra Features	Synthesized Features
Age Activity Married Personality	Accessibility Franchise Open Area Services	Sub-rating: Food Sub-rating: Service	Alcohol Score Cuisine Score Days Open Dress Match Noise Match Proximity Smoker Smoking Score

VI. MODEL SELECTION

The most important part of this project was selecting the type of model to use. While many different factors went into selecting the correct model, much experimentation was necessary to determine which models worked best with which settings. The first batch of tests focused on testing models with great categorical diversity, but quickly showed that clustering algorithms and tree-based approaches generally performed significantly worse than the rest. See the below chart (Fig. 1) for a comparison on scores for model accuracy, MSE and RMSE.

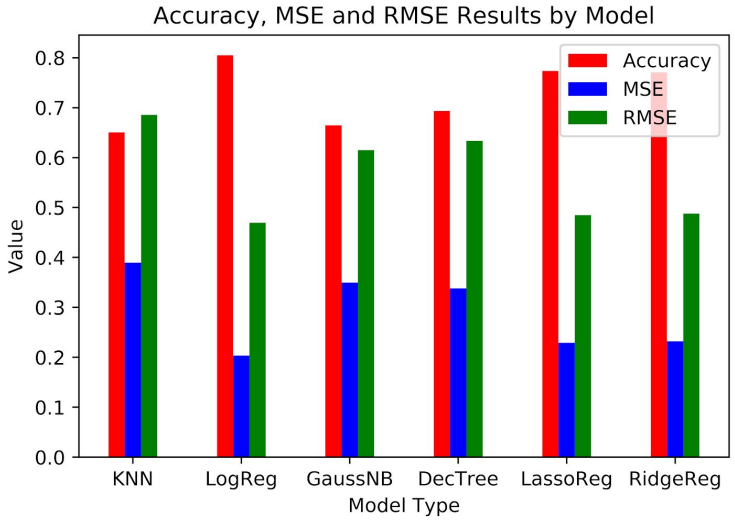

Fig. 1 - Performance of various models

From the first wave of results, linear regression models were shown to be highly effective compared to the alternatives. Three noteworthy candidates with relatively similar performances were Linear, Lasso and Ridge Regressions, with some of those results being show in the table above (Fig. 1). Seeing clustering perform as poorly as it did came as a surprise, given how the algorithms had performed in prior coursework. Despite attuning parameters in a variety of ways, the performances on outside of the linear models were lackluster, and left much to be desired. After exhaustively trying many linear models, the clear best choice was the Logistic Regression. After further parameter attuning, the results stood clearly above all other candidates. The specific parameter configuration included the use of 'solver', 'multi_class', and 'C'. The best solver was ‘newton-cg’, which was the option for handling multiclass problems with multinomial loss. The multi_class setting was set to ‘multinomial’, setting the loss to be the minimized loss fit across the entire probability distribution, even if the data is binary. Lastly, the inverse regularization strength parameter (‘C’) was set to 1.5.

VII. EXPERIMENTAL RESULTS

As previously discussed, the Logistic Regression approach proved to be the most effective. After completing the parameter fine-tuning and selecting the best set of attributes to run the model with, the primary run was completed. The training data was synthesized, processed, and fed into the model, and the model was then applied to the testing data. Show below are the results for the execution of this model using the testing data (Accuracy, RMSE and MSE):

```
Execution complete. The performance of LOGISTIC REGRESSION is as follows:
Accuracy: 0.8051575931232091
RMSE: 0.4697130746454017
MAE: 0.2034383954154728
```

As shown above, the accuracy of the model rests at roughly 80.52%, with an RMSE of 0.470, and an MAE of 0.20. These results are fairly promising, given how drastic the increase in performance was given the original unattuned linear models. Modifying the feature selections and fine-tuning the model parameters led to a significant increase of effectiveness; roughly a 3% accuracy increase at minimum, compared to the second-best linear model. Likewise, given the high accuracy, we see expectedly low MAE (mean average error) and RMSE (root mean average error) values; this relationship is maintained with all three accuracy measurements. Shown below (Fig. 2) is a depiction of the predicted values versus the true values; in general, the model expected users to rate closely to their true values fairly often, though when the model misclassified, it typically assumed the user rated more highly than they did. The predictions are shown in red, with the true values being shown in blue.

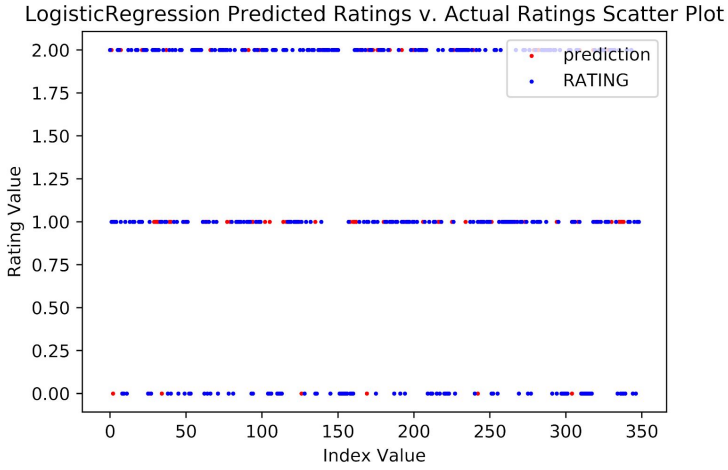

Fig. 2

VIII. CONCLUSION

In conclusion, after analyzing and preprocessing this restaurant data, extracting numerous features, and testing a wide variety of training models, Logistic Regression was chosen as the best method. As discussed earlier, the final results notably impressive, as the final model performed with an accuracy of nearly 80.52%. Despite testing a variety of models with an additional variety of parameter setups, the Logistic Regression remained the best performer. Naturally, there still exists plenty of room for the improvement of this model; if given more time, further model exploration and additional feature synthesis could be performed to potentially refine the performance further. Overall, the final model performed fairly well, earning an impressive level of accuracy, while also keeping run times reasonably low.