What makes a good restaurant?

Amine M'Charrak
Computer Science and Engineering
University of California, San Diego
La Jolla, California
Email: amine.mcharrak@tum.de

Bhaskar Kumar Mishra
Computer Science and Engineering
University of California, San Diego
La Jolla, California
Email: bkmishra@ucsd.edu

Prashant Pandey
Computer Science and Engineering
University of California, San Diego
La Jolla, California
Email: p3pandey@ucsd.edu

***Abstract*—In the project we want to answer the question : *What makes a Good Restaurant* using techniques from Machine Learning and Natural Language Processing. One way to answer this question is to identify an extensive set of features associated with both the restaurants and the people visiting them. This again, can be accomplished either by using explicit features from users and businesses from the data as in Feature Based Models or inferring them implicitly from the data as in Latent Factor Models. Yet another approach might be to examine restaurants review text and formulate an understanding of Topics that tend to be associated with good restaurants and subsequently the bad ones. In this project, we implement these techniques for Yelp Restaurant Review Dataset. We compare a variety of Feature Based Models like Linear Regression, Ridge Regression , Neural Networks etc and a variety of Latent Factors Models like the classical Latent Factor Model (LFM) and Latent Dirichlet Allocation (LDA) amongst themselves and with each other.**

***Keywords*—Recommender system, Latent Factor Model, Latent Dirichlet Allocation, Text Mining**

I. Introduction

Review website such as Google Local, IMDB and Yelp, allow consumers to evaluate places they visited or movies they saw. Therefore, today many people use these platforms to decide weather they are going to visit a place they are interested in or not. As most of us do not have a lot of time to elaborate on our decisions, we tend to look at the reviews, more specifically we often just look at the star rating and perhaps a few of the text reviews in order to double check that this place is worth a visit. Thus, it can be stated, that the rating stars of the reviews for businesses and items heavily influence potential consumers during the decision making process. The drawback of these reviews is, that they are subjective. One person might give 5 stars and another person might give only 1 star even though they visited the business on the same day and the setup was the same. Other people tend to give always 5 star when they like something whereas some consumers use the full range from 1 to 5 stars.

Many a times when building models on rating predictions, we are not concerned about the accuracy of such systems in terms of minimising an error metric but instead, on getting a conspicuous understanding of these models so they can we can translate the learnings into clear set of action items for businesses in order to help them improve their offerings. For instance, we care about answering questions like whether the presence of *free WiFi* or *business Parking* makes the restaurant more desirable to people. In an attempt to achieve this we implement a bunch of different models and more importantly dissect them to identify statistically significant features and attributes that influence restaurant ratings. We use the dataset provided by Yelp for training and testing our models. In the following chapters we first introduce the dataset and its properties. After that, we perform an exploratory analysis of our data in order to determine important features and unusual trends within the dataset. We then explain which models we decided to use in order to solve the predictive task and how we implemented the models. Finally, we make conclusions based on our results of exploratory analysis but also the prediction task.

II. Literature Survey

A lot of work have been done on rating prediction tasks in the past. One of the most famous advancements in the field of rating prediction was the introduction of the latent factor model. This model was proposed during Netflix prize competition by Simon Funk [10] and it gained a lot of traction after the competition following which, many other variants of latent factor models were introduced.

Koren, the winner of Netflix prize competition, proposed a variant of the above model by adding user and item bias terms in the basic matrix factorization model. Later, he also introduced an algorithm called SVD++ that outperformed Netflix Recommender system [4][5][7]. SVD++ exploited both implicit and explicit feedback from the users to improve the rating predictions [4]. Latent Factor Models uncover latent features from the ratings discarding the reviews text.

Several topic modeling techniques have been developed to learn hidden topics from reviews text. Latent Dirichlet Allocation (LDA) is one of the common unsupervised learning algorithms to discover the hidden topics [9]. There are variants in topic modeling methods like Latent Semantic Analysis(LSA) and the Probabilistic Latent Semantic Analysis. Over the past decade, various other models have been proposed which use LDA for topic modeling and rating predictions. One of such methods involved averaging over all reviews ratings that contained the given topic to calculate the hidden topic rating [2]. This model won first prize in Yelp Dataset Challenge. One more study by Julian McAuley fuses latent rating dimensions and latent review dimensions to predict the star ratings more accurately. This model recorded an RMSE of 1.176 and won him the first prize on Yelp Dataset Challenge 1. We implement a bunch of these techniques for our rating prediction task.

III. Dataset and its properties

The Yelp dataset we are analyzing is downloaded from yelp website and it contains subsets of reviews, business and user data which can be used for educational and academic purposes. Since the data is available as both JSON and SQL file, we decided to work with the JSON format because we were familiar using it due to the previous homeworks and assignment 1. Each file is composed of a single object type, one JSON-object per-line. The dataset is split into three parts and has the following specifications:

TABLE I: Dataset Overview and Feature Examples

Yelp Dataset Overview		
Type	Entries	Predictors
business.json	156,000	”review_count” ”city” ”stars” ”categories”
review.json	4,700,000	”user_id” ”business_id” ”text”
user.json	1,100,000	”average_stars” ”review_count” ”business_id” ”yelping_since”

Out of these 4,700,00 reviews, roughly 2,900,000 are restaurant specific. There are 156,000 unique businesses of which 51,000 are restaurants. For our analysis purpose in the following chapters, we will be focusing solely on restaurants in Arizona. The dataset contains 10,000 restaurants of Arizona. Out of all reviews approximately 800,000 reviews have been given to restaurants in Arizona.

IV. Exploratory Analysis

A. Statistical Analysis

It is remarkable to observe that approximately 50% of the restaurants within the data came from only two states which are Ontario and Arizona. Figure 1 displays the percentage distribution of restaurants across different states. Moreover, we haave a look at what star rating distribution these bars of Figure 1 are actually containing. On a scale from 1 to 5 stars the hidden distribution is given in Figure 2. Next we are going to analyze the star rating distribution for restaurants which are contained within the business.json dataset. As we can see in Figure 3, the peak is around 3.5 and 4.0 stars with a stronger cutoff drop towards higher ratings and a smoother transition towards lower ratings. Furthermore, we want to see the amount of restaurant reviews for each type of star-rating as shown in Figure 4. We extracted these values from the reviews.json file, therefore it does only contain integer star values and not decimal star ratings as we see them in Figure 3. Now let us look at the number of restaurant reviews with different star ratings. To sum up, we can see, that there are only a few restaurants with an average rating of 1 or 5 stars. Around half of all restaurants have an average rating between 3.5 and 5.0 stars. We found out that there are only few restaurants with an average rating of either 1.0 or 5.0. Around 50% of all restaurants have an average

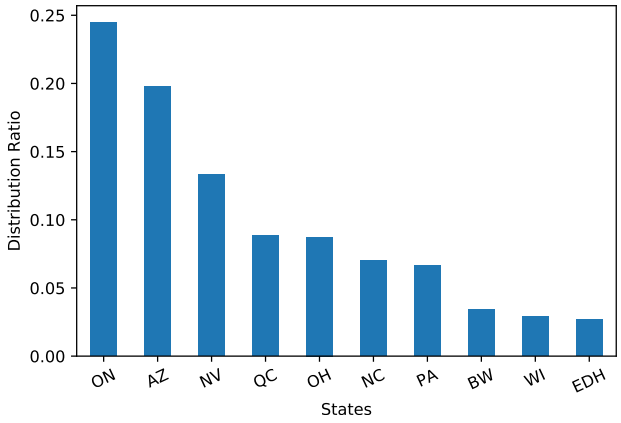

Fig. 1: Geographical Distribution of Restaurants per State

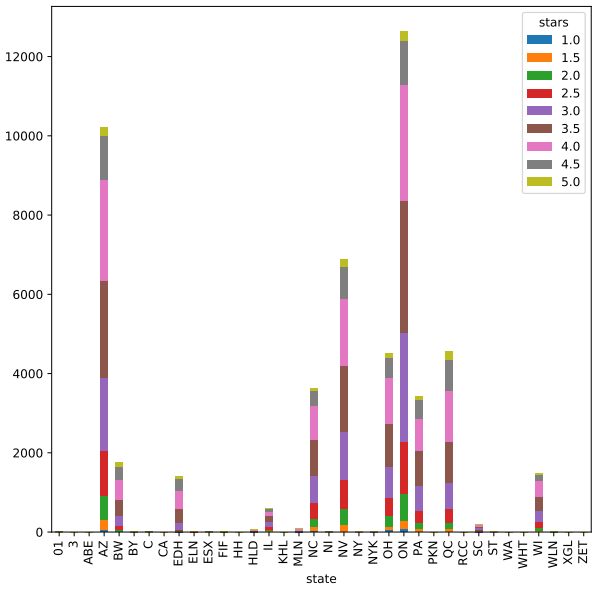

Fig. 2: Star Rating Distribution per State

rating of either 3.5 or 4. It is surprising to note that more than 70% of the reviewed restaurants belonged to either 4 star category or 5 star category. From Figure 3 and Figure 5 we can conclude that the restaurants with strong ratings have been more reviewed respectively visited by users even though they are present less in numbers compared to other star categories.

B. Time Series Analysis

From the restaurant reviews of Arizona, we found out, that the reviews are from a timespan between February 2005 and July 2017. We therefore tried to observe weekly, monthly and annual trends within the reviews and find out how the reviews and certain features behave over time. The first interesting property is that most of the restaurants tend

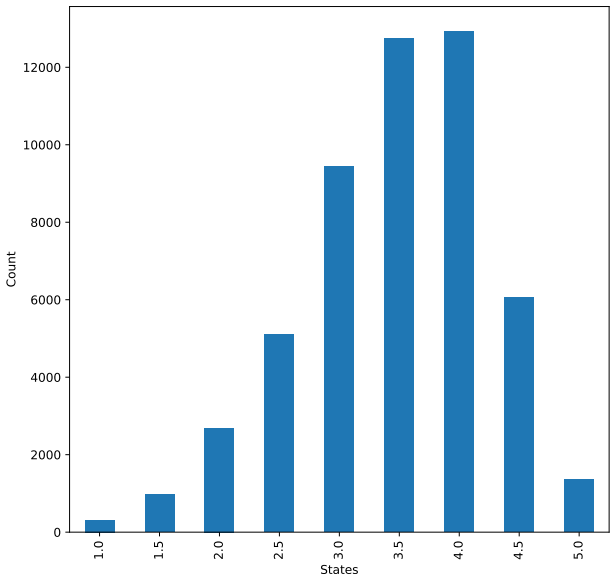

Fig. 3: Star Rating Distribution of Restaurants

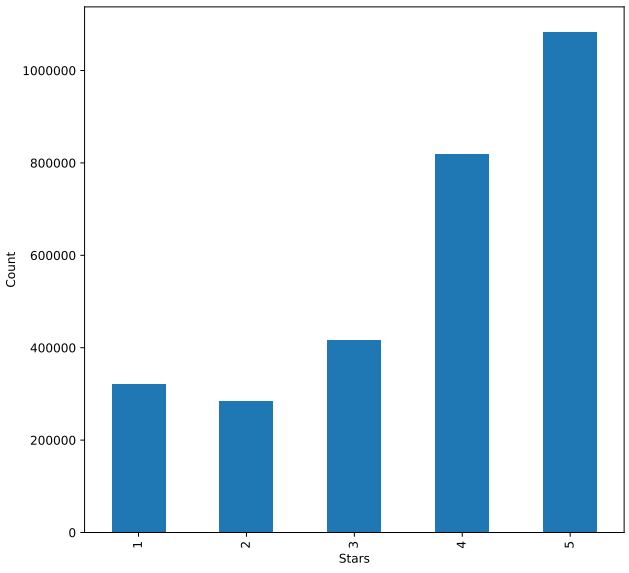

Fig. 4: Count of Reviews for each Star Category

to receive a maximum number of reviews every Sunday and a minimum number or reviews between Tuesday and Thursday as can be seen in Figure 5, where we can observe the weekly behaviour of certain features such as:

- *cool*
- *useful*
- *funny*
- *star rating*

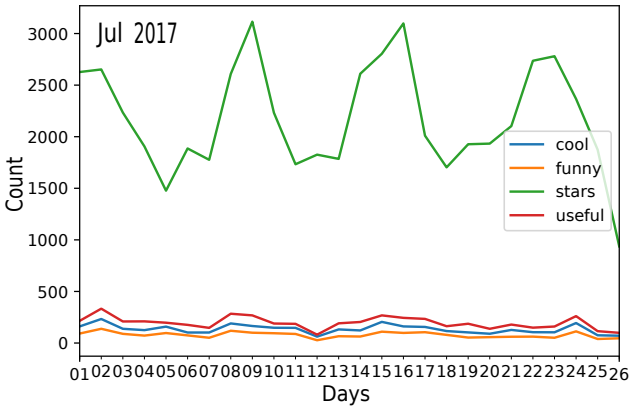

Fig. 5: Daily Feature Analysis

This is probably due to the fact that most customers are free on weekends and hence they prefer to visit restaurants in general on say Friday or Saturday and then commit a review on Sunday when they often rest and stay at home. Next we are going to look at the monthly analysis for the same set of predictors in Figure 6. Similarly, we found out, that restaurants receive a minimum number or reviews in the month of January and a maximum number of reviews in the month of April. Last but not least, we are going to zoom out even more

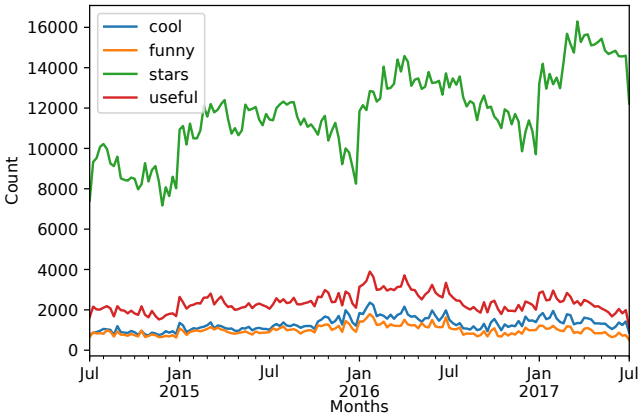

Fig. 6: Monthly Feature Analysis

in order to have an overview of what happened with the features since the dataset was being recorded in 2005. Figure 7 gives us a good idea where the trend is going to. We can observe that number of reviews in restaurants has increased significantly from 2005 to 2017; but we can also observe, that the feature *star rating* is more dominant than features like usefulness or coolness. One way to explain this is that

the popularity of Yelp has increased over time and that people tend to rate businesses but they do still keep back in reviewing other reviews and marking them as either funny, useful or cool because most of the features are not given anonymously.

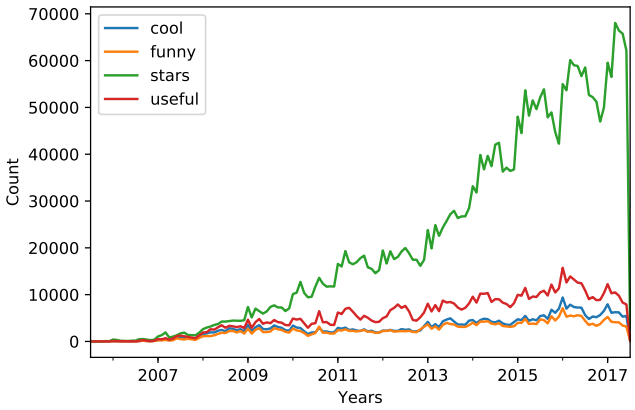

Fig. 7: Yearly Feature Analysis

V. Feature Based Models

A. Feature Selection

We used the review.json and business.json files for features selection. The review.json file contains filed like "user_id", "business_id", "review_text"etc. along with associated ratings. Furthermore, it consists of categorical field like "funny", "cool" and "useful". We however did not use these categorical fields in our models because most of these fields are either extremely skewed or very sparsely populated, meaning that many of these entries are null. In order to rule out the significance of these categorical variables, we performed χ^2 testing and as expected, we found out, that these features are highly insignificant at a type I error rate of 0.05. A lot of attributes from the business.json file we used in our rating prediction task. Most of these predictors we had to one-hot encode because they were categorical variables like "BusinessParking", "GoodForMealLunch", "Alcohol_beer_and_wine", "HasTV" or "WiFi_free". A few continuous valued predictors were also fed to these models like the "userAvgRating" or "userReviewCount". Overall, a total of 38 predictors from both review.json and business.json were used in our models in order to predict the rating. Figure 8 represents the significance order of these predictors based on the p-values obtained from *Linear Regression*. At a significance level of 0.05, we observed that user features like "userAvgRating" and "userReviewCount"were highly significant. Moreover, we found business features like "OutdoorSeating", "WiFi_free" and "NoiseLevel_quiet" to be highly significant predictors for the rating task.

B. Models

For all following feature based models below, the 200,000 reviews were randomly shuffled and split into a training set of 150,000 samples and a testing set of the remaining 50,000

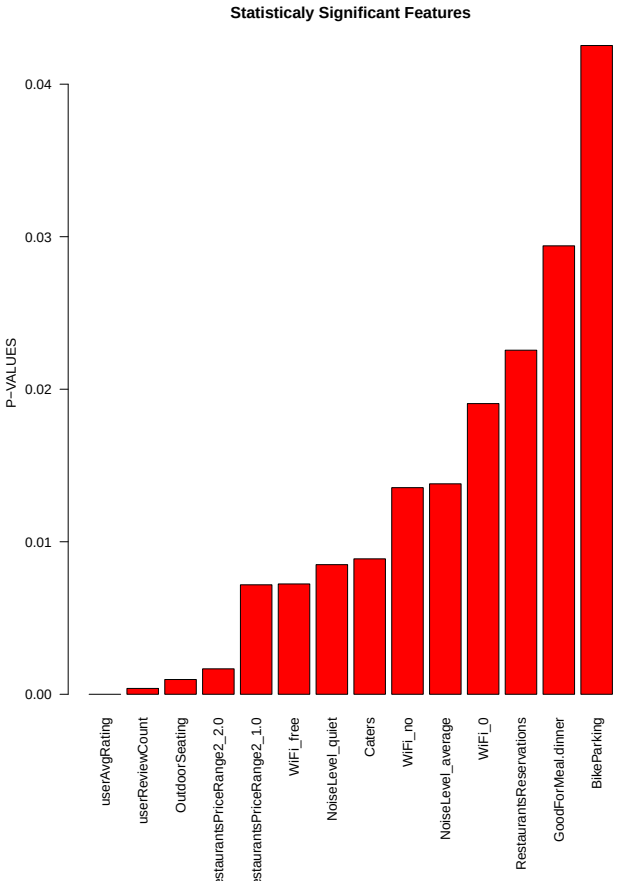

Fig. 8: Statistically Significant Features

testing samples. Next a *Grid Search* was used in order to tune the hyper-parameters for all these models. Finally we perform *mean normalization* and *feature scaling* on all or our models.

1) Linear Regression:
We used the *Ordinary Least Squares Regressor* from the Python numpy-package (scipy.linalg.lstsq) and set the "fit_intercept"-parameter to True. We obtained a *mean squared error* MSE of 1.2313 and a *root mean squared error* RMSE of 1.1096 on our test set for the Linear Regression Model.

2) Ridge Regression:
We used the standard Linear least squares with $L2$ regularization for our *Ridge Regression Model* with a regularization parameter lambda of 1.0 and the "fit_intercept"-parameter set to True again. We obtained a MSE of 1.2313 and a RMSE of 1.1096 on our test set for the Ridge Regression Model.

3) Support Vector Regression (SVR):
We used the standard based *Epsilon-Support Vector Regression Model* from the Python libsvm-package with a penalty parameter C for the error term of 1.0 and epsilon set to 0.1. Moreover, we choose the kernel type "rbf" and a polynomial kernel function of degree 3 for the fitting. We obtained an MSE of 1.2688 and an RMSE of 1.1264 on our test set for our Support Vector Regression.

4) Random Forest Regression:
We used the Python sklearn-package for the *Random For-*

est Model with number of trees in effect the parameter "n_estimators" set to 10. We chose the optimization metric as MSE with "min_samples_split" equal to 2 and the "bootstrap"-option set to True. With the Random Forest Regression we achieve an MSE of 1.4305 and an RMSE of 1.1960 on our test set.

5) AdaBoost Regression:
We use the *AdaBoost Regression Model* with the "base_estimator" variable set to "DecisionTreeRegressor", number of trees "n_estimators" set to 50 and a "learning_rate" of 1. We used the *Square Loss Function* for fitting. On our test set we get a MSE of 1.3581 and a RMSE of 1.1654 with the Random Forest Regression.

6) Neural Network Regression:
We also tried out a *Neural Net Regressor Model*. Figure 9 depicts the architecture of our Multilayer Perceptron Network using *resilient backpropagation* with one *hidden layer* consisting of 20 neurones. In order to avoid overfitting, we choose a drop-out rate of 50%. With this architecture we are able to obtain a MSE of 1.2182 and a RMSE of 1.1037 on our test set for our Neural Net Regression Model.

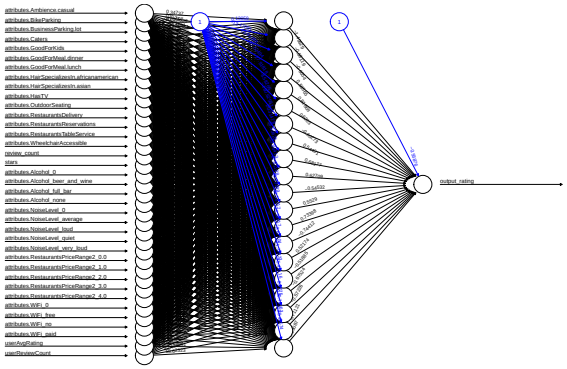

Fig. 9: Neural Netowork Regression Architecture

C. Comparison of Feature Based Models

Based on the RMSE obtained for the test set surprisingly Ridge Regression and Linear Regression outperformed more complicated models such as SVR (with a polynomial kernel of degree 3), Random Forests or AdaBoost. This might probably be attributed to the problem of *overfitting* when employing more complex nonlinear models. The bar plot in Figure 10 shows the RMSE values for a bunch of different linear and nonlinear regression models used in the previous subsection. Evidently, the neural networks outperforms all other models with the lowest RMSE of 1.1037 on the test set.

VI. Latent Factor Models

For restaurants to be successful, businesses must consider the tastes and preferences of the users and then try to improve on those attributes that users care about. In order to discover these latent attributes from the data implicitly we must consider the interactions between users and businesses.

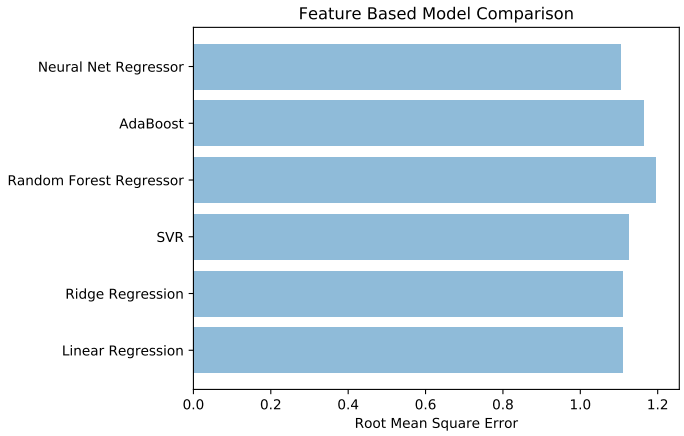

Fig. 10: Performance Evaluation of Feature Based Models

A. Word Clouds

We start by plotting word clouds of all 5 star reviews and 1 star reviews to check what positive and negative aspects of restaurants have been discussed by the users. Figure 11 shows the word cloud of all 5 star reviews. It can be seen that some most frequent words occurring in restaurant with 5 star ratings are a combination of positive adjectives, and some nouns like - "love", "great", "service", "time", "fresh", "food" , "best" , "delicious", "friendly" etc. Quite naturally, it can be inferred that properties like friendly staff, fresh and delicious foods or great service etc. have positive connotations and thereby are a strong indicator of higher ratings. Likewise, the word

Fig. 11: Word Cloud of all 5 Star Ratings

cloud of 1 star ratings as shown in Figure 12 gives us some insights about the attributes that tend to be associated with poor ratings like negative adjectives, and some nouns such as "never", "time", "service" and "food". It can be inferred that these restaurants lack in some of the properties like food quality, staff behaviour, punctuality or service quality. Thus they need to improve on these properties in order to receive higher ratings.

B. Classical Latent Factor Model

For prediction tasks of the latent factor models, we use 200,000 randomly shuffled reviews from Arizona as training

Fig. 12: Word Cloud of all 1 Star Ratings

dataset, 150,000 reviews as validation set and 50,000 reviews as testing set.

Classical Latent Factor Model:

C. Classical Latent Factor Model

Model 1:

The classical latent factor model predicts the rating of the restauarnts according to the equation given below.

$$\hat{R}_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u \gamma_i \tag{1}$$

Here $R_{u,i}$ denotes the the rating of restaurant i by user u whereas $\hat{R}_{u,i}$ stands for the rating predicted by the model. The global offset term is denoted by α and the bias parameters for restaurant i and user u are the variables β_i and β_u respectively. γ_i can be interpreted as the attributes of restauarant i, and γ_u can be interpreted as the preference of user u towards those attributes. γ_u and γ_i, therefore, incorporate the K-dimensional latent features for user u, and restaurant i respectively. This model uncovers the latent(hidden) features from ratings given by user u to a restaurant i. However, it completely ignores the reviews given by users. This model uses the following update rule to learn its parameters:

$$\alpha^t = \frac{\sum_{u,i} R_{u,i} - \beta_u^{t-1} - \beta_i^{t-1} - \gamma_u^{t-1}\gamma_i^{t-1}}{N_{train}}$$

$$\beta_u^{t+1} = \frac{\sum_{i \in I_u} R_{u,i} - \alpha^t - \beta_i^t - \gamma_u^t \gamma_i^t}{\lambda_u + |I_u|}$$

$$\beta_i^{t+2} = \frac{\sum_{u \in U_i} R_{u,i} - \alpha^{t+1} - \beta_u^{t+1} - \gamma_u^{t+1}\gamma_i^{t+1}}{\lambda_i + |U_i|}$$

$$\gamma_{u,k}^{t+3} = \frac{\sum_{i \in I_u} \gamma_{i,k}(R_{u,i} - \alpha^{t+2} - \beta_u^{t+2} - \beta_i^{t+2} - \sum_{j \neq k} \gamma_{u,j}^{t+2}\gamma_{i,j}^{t+2})}{\lambda_{\gamma_u} + \sum_{i \in I_u} \gamma_{i,k}^2}$$

$$\gamma_{i,k}^{t+4} = \frac{\sum_{u \in U_i} \gamma_{u,k}(R_{u,i} - \alpha^{t+3} - \beta_u^{t+3} - \beta_i^{t+3} - \sum_{j \neq k} \gamma_{u,j}^{t+3}\gamma_{i,j}^{t+3})}{\lambda_{\gamma_i} + \sum_{u \in U_i} \gamma_{u,k}^2}$$

D. Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is a generative statistical model for text, which is used as a topic modelling tool to discover hidden features from text reviews. In LDA, each document d_i is viewed as a collection of different topics, which is a K-dimensional stochastic vector. Each topic is a collection of a fraction of words that describes each topic with certain probability.

We used LDA on our entire training dataset to find out hidden topics from reviews [11]. We tried models with different values of latent factor(K), and finally we chose the value of K as 40. Table 2 shows few topics that we extracted by running LDA on the entire training data set. For example,the topic "Staff" is made up of words like order, "manager", "employee", "service" etc. Similarly, the topic "Sea Food" is made up of words like "sushi", "roll", "fish", "tuna", "salmon" etc. Each word inside a topic appears with a probability of it being associated with that topic.

TABLE II: Topics Extracted from Reviews by LDA

Topics	Words
Staff	Order, Manager, Service, Employee, Staff, Owner, Counter
Sea Food	Sushi, Roll, Fish, Tuna, Salmon, Fresh, Japanese
Menu	Menu, Wine, Dish, Item, Course, Meal, Restaurant
Cleanliness	Bathroom, Dirty, Floor, Parent, Clean, Coke, Movie
Service Time	Minute, Order, Time, Came, Table, Service, Ordered
Payment	Card, Dim, Cash, Sum, Credit, Stated, Continued
Group	Table, Room, Dining, Experience, Reservation, Group, Party
Desserts	Cream, Desserts, Chocolate, Ice, Cake, Sweet, Strawberry
Beer	Beer, Selection, Nacho, Tap, Brew, Bar, Local
Nighttime	Late, Night, Club, Hit, Open, Scottsdale, Miss
Food	Pizza, Italian, Pasta, Crust, Sauce, Cheese, Gluten

Lets take an example of a 1 star review given below:

"Absolutely hot garbage and overrated Please spend your hard earned money elsewhere. My room smelled like cat urine and the sinks and shower were slow draining, great place if you like cold showers in the morning time. I would rather stay in a Marriot Courtyard any day of the week."

The top 5 topics returned by this review are "Place", "Room", "Payment", "Service" and "Visit". It is clear from the topics, that in order to improve the ratings, the restaurant owner must try to improve attributes like room-condition, service and place of stay.

Model 2:

We used LDA to predict ratings of the restaurants according to the given equation:

$$\hat{R}_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u d_i \tag{2}$$

Where d_i denotes a K-dimensional stochastic vector for the restaurant i.

Where α is a global bias term, β_u is a user bias term and β_i is a restaurant bias term. The variable γ_i can be calculated by taking a list of all reviews for a particular restaurant and then using LDA to get a K-dimensional stochastic vector for

each restaurant d_i. Each element of d_i consists of probability of that topic associated with the reviews of the i_{th}-restaurant. The variable γ_u can be interpreted as the preference of user u towards those topics (attributes). This model uncovers the hidden features from reviews text at training time. The update rule for this model is same as that of the classical latent factor model except for the fact that here we dont update d_i after we initialize it with stochastic vector for each product.

Model 3:

We also experimented by combining γ_i and d_i from above 2 models in order to extract hidden attributes of restaurants from both rating dimension and review text dimension.

$$\hat{R}_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u(\gamma_i + d_i) \tag{3}$$

where the symbols have their usual meanings as defined in previous models. This model uses the following update rule for γ_u and γ_i:

$$\gamma_{u,k} = \frac{\sum_{i \in I_u} (\gamma_{i,k} + d_{i,k})(R_{u,i} - \alpha - \beta_u - \beta_i - \sum_{j \neq k} \gamma_{u,j}(\gamma_{i,j} + d_{i,j}))}{\lambda + \sum_{i \in I_u} \gamma_{i,k}^2 + \sum_{i \in I_u} d_{i,k}^2}$$

$$\gamma_{i,k} = \frac{\sum_{u \in U_i} \gamma_{u,k}(R_{u,i} - \alpha - \beta_u - \beta_i - \sum_{j \neq k} \gamma_{u,j}\gamma_{i,j} - \sum_{j \neq k} \gamma_{u,j} d_{i,j})}{\lambda + \sum_{u \in U_i} \gamma_{u,k}^2}$$

In all these models K is the number of latent features or topics when performing the Latent Dirichlet Allocation (LDA). The results for our different Latent Factor Models can be seen in Table 3. LFM gave a RMSE of 1.268, LDA performs with a RMSE of 1.284 and finally the mixture of both models in effect our third model leads to a RMSE of 1.286. We can clearly see, that the RMSE value of the simple Latent Factor Model is the best.

TABLE III: Comparison of Latent Models

Model Type	RMSE
Latent Factor Model (LFM)	1.268
Latent Dirichlet Allocation (LDA)	1.284
LFM + LDA	1.286

Interesting Findings:

We used LDA to find similarity between tastes of users in Arizona and users of Ontario. For this, we ran our model separately on 5 stars rated reviews of Arizona and reviews of Ontario and found the results as shown in Table 3, it displays the top 4 topics about which users discuss more when they give 5 star rating to a restaurant. The result shows that attributes like Service Time and Quality are more important to restaurants in Arizona, whereas attributes like Menu(variety of items), delicious food items are more important to restaurants in Ontario. It also shows that attributes like place, and staff behavior are important for both the states.

TABLE IV: Top 4 Topics Users Discussed in 5 Star Ratings

Arizona	Ontario
Place (25%)	Place (24%)
Service Time (9%)	Staff (11%)
Staff (6.6%)	Menu (6%)
Quality (4.6%)	Delicious Items (4.5%)

VII. Results and Overall Model Comparison

Our overall results can be seen in Table 5. The table reveals that Regression Models had better performance than Latent Factor Models. One of the reasons could be the selection of explicit fine grained features from the business.json file.

TABLE V: Overall Model Comparisons

Model Type	RMSE
Linear Regression	1.10962
Ridge Regression	1.10963
Support Vector Regression	1.12641
Random Forest Regression	1.19604
AdaBoost Regression	1.65393
Neural Net Regression	1.10372
Latent Factor Model (LFM)	1.268
Latent Dirichlet Allocation (LDA)	1.284
LFM + LDA	1.286

VIII. Conclusion

In this report, we have analyzed both Feature Based and Latent Factor models for predicting the ratings of restaurants. We observed that Feature Based Models gave slightly better results than Latent Factor models for our dataset. This can be attributed to the fact that we chose a relatively smaller subset (200,000 data points) of the entire data (5,000,000 data points) and therefore Latent Factor Models were unable to discover all the latent features for prediction. On the other hand, we were able to obtain very fine-grained predictors for our Feature Based Models from business.json and review.json which increased the predictive power of our Feature Based Models.

We also observed that amongst the Feature Based Models, simple Linear Models like Linear Regression and Ridge Regression gave better results than more complex nonlinear models like SVR (with a polynomial kernel), Random Forest and AdaBoost. This can be potentially explained by the fact that these nonlinear models might have been slightly *overfitting* the training data even with enforced regularization. In order to test this, we trained a Neural Net Model with a high dropout rate of 0.5 to counter the low bias problem associated with nonlinear models. We saw that the Neural Net model gave us the best results in term of MSE. Also, amongst the Latent Factor Models the results obtained were pretty close to each other with the basic version of Latent Factor Model outperforming the Latent Dirichlet Allocation and LDA + LFM mixture models by a small margin. Even though the feature based models outperformed the Latent Models, we observed

how LDA Models can harness the power of text reviews to help us identify the set of attributes/ Topics which are typically associated with good Restaurants and similarly for the bad ones.

IX. Future Work

One interesting follow-up of our analysis would be to see whether increasing the number of training data from 200,000 to 5,000,000 allows Latent Factor Models to outperform Feature Based models in terms of MSE. We would also be interested in extrapolating our work on both Feature Based and Latent Factor models to examine temporal effects.

References

[1] J.McAuley, Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text

[2] James Huang, Improving Restaurants by Extracting Subtopics from Yelp Reviews

[3] Jack Linshi, Personalizing Yelp Star Ratings: A Semantic Topic Modeling Approach

[4] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. Computer, 2009

[5] R. Bell, Y. Koren and C. Volinsky. The BellKor 2008 Solution to the Netflix Prize. 2008

[6] Zhou Tong , A Text Mining Research Based On LDA Topic Modelling

[7] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 426434. ACM, 2008

[8] J. McAuley and J. Leskovec. From amateurs to connoisseurs: model- ing the evolution of user expertise through online reviews. In Proceed- ings of the 22nd international conference on World Wide Web, pages 897908. International World Wide Web Conferences Steering Com- mittee, 2013

[9] Blei, David M. and Ng, Andrew Y. and Jordan, Michael I., Latent Dirchlet Allocation, J. Mach. Learn. Res., http://dl.acm.org/citation.cfm?id=944919.944937, 2003

[10] Netflix update: http://sifter.org/ simon/journal/20061211.html