**What Makes a Good restaurant**

**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 restaurant 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 restaurant’s 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 Models (LFM) and Latent Dirichlet Allocation (LDA) amongst themselves and with each other.

1. **Introduction**

Recommender Systems are integral to a variety of services today. The sheer amount of data available in the form of peer contributed reviews and ratings from social platforms like Yelp, Zomato and the likes have provided a huge impetus to the performance of these systems. But many a times, 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 begin by briefly discussing the existing Literature on these types of Modelling followed by Exploratory Data Analysis to better understand the properties of our dataset.

1. **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’s 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]. One more study by J.McAuley fuses latent rating dimensions and latent review dimensions to predict the star ratings more accurately. This model recorded an RMSE of 1.176 on Yelp Challenge Data. We implement a bunch of these techniques for our rating prediction task.

**Data Statistics and Exploratory Analysis**

We have used the Yelp ratings and review dataset from Yelp’s website for our analysis. The dataset contains around 4.7 Million reviews, 1.1 Millions unique users data, and around 150K unique business data. The data are available in json format. Out of total businesses around 51K are restaurants.

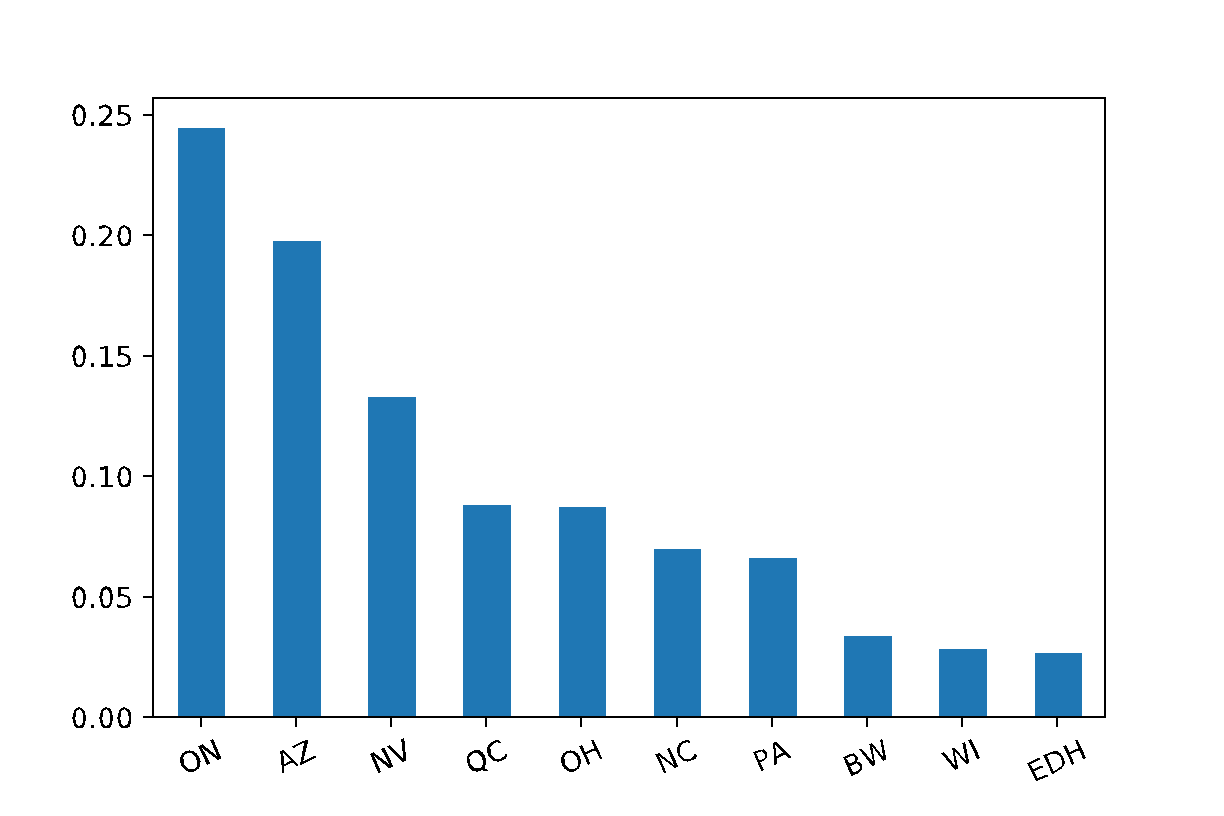


Fig 1 : Geographical Distribution of Restaurants

It was bit weird to observe that approximately 50% of the restaurant's data came from just 2 states – Ontario and Arizona. Fig 1 above displays the percentage distribution of restaurants across different states.

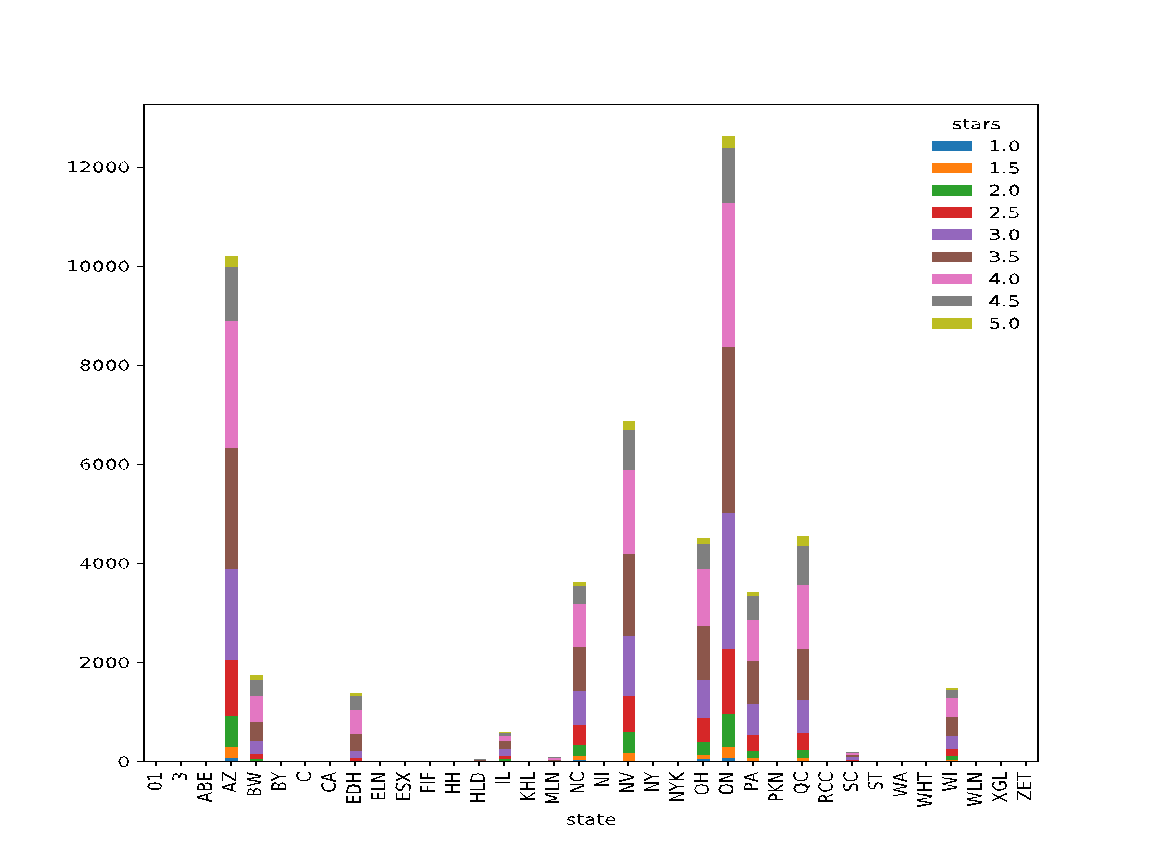


Fig 2.a : Geographical distribution of restaurants with different star rating

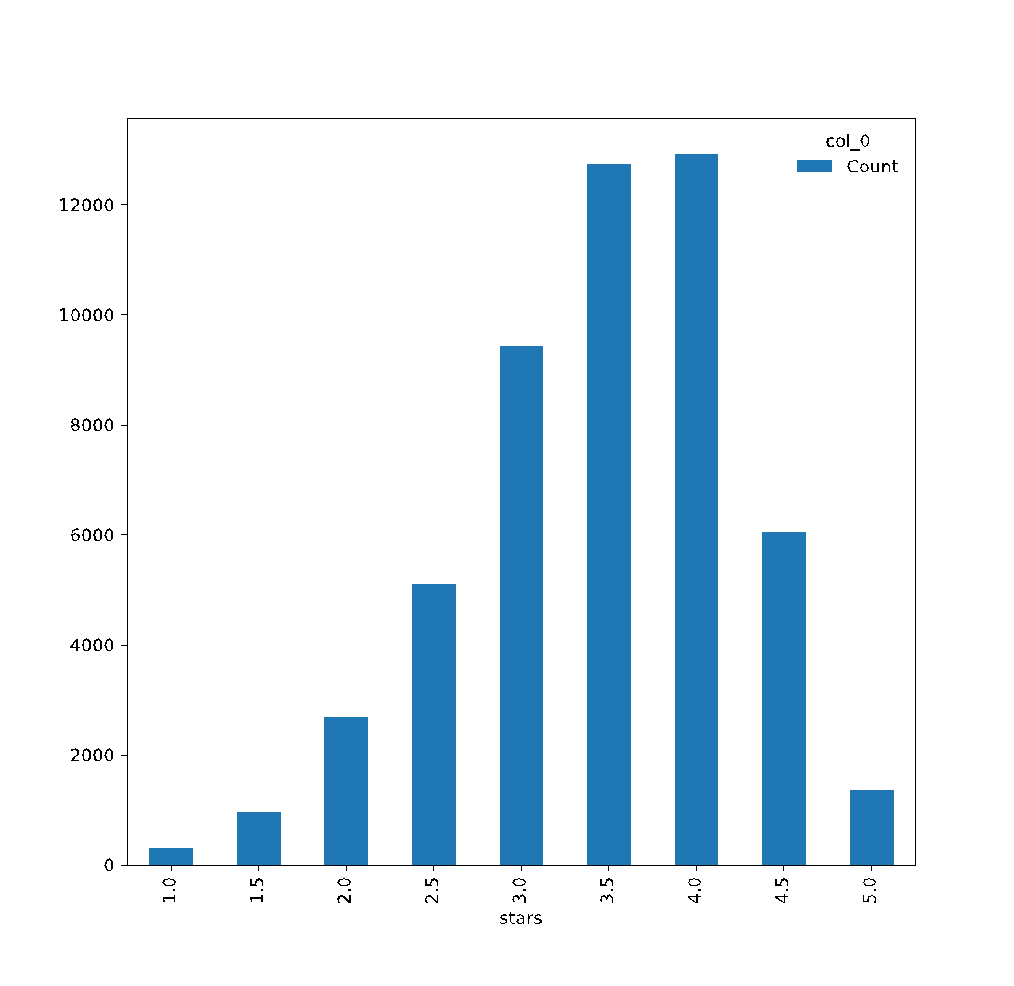


Fig 2.b : Number of restaurants with different star rating

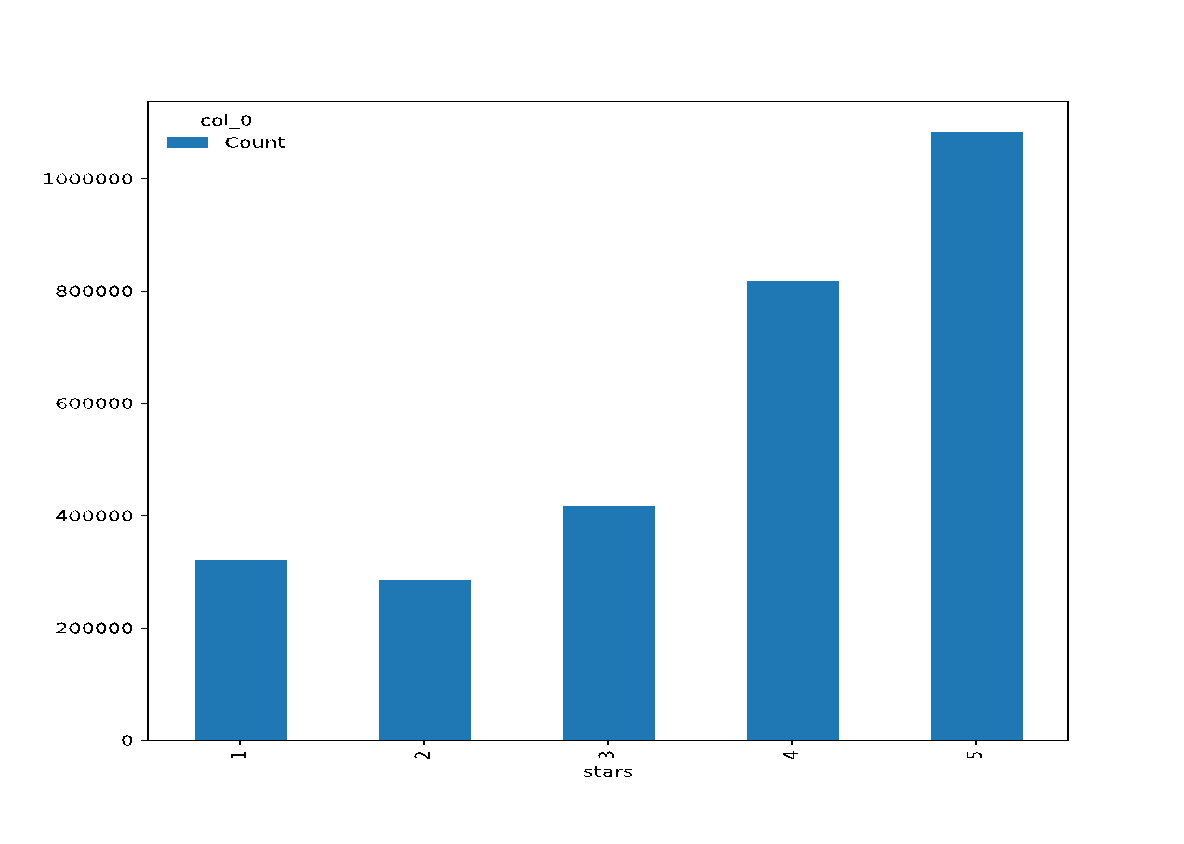


Fig 2.c : Number of Reviews with different star rating

For our purpose of our analysis, we drilled down to only the restaurants in Arizona. The dataset contained around 10k restaurants in the state of Arizona. Out of the total restaurants, approximately 800k reviews have been given to restaurants in Arizona. Stars ratings in the review is an indicator of a positive feedback on these restaurants. We found that there are only a few restaurants distributed geographically, with average rating of either 1 or 5. Around 50% of the total restaurants have average rating of either 3.5 or 4. Fig 2.a displays the geographical distribution various star ratings. We also plotted the distribution of review rating and were surprised to notice that more than 70% of the reviewed restaurants belonged to either 4 star category or 5 star category. It indicates that the restaurants with good ratings have been reviewed (or visited) more by users even though were less in numbers compared to other businesses in the dataset. Fig 2.c displays the plot of number of reviews with different star ratings.

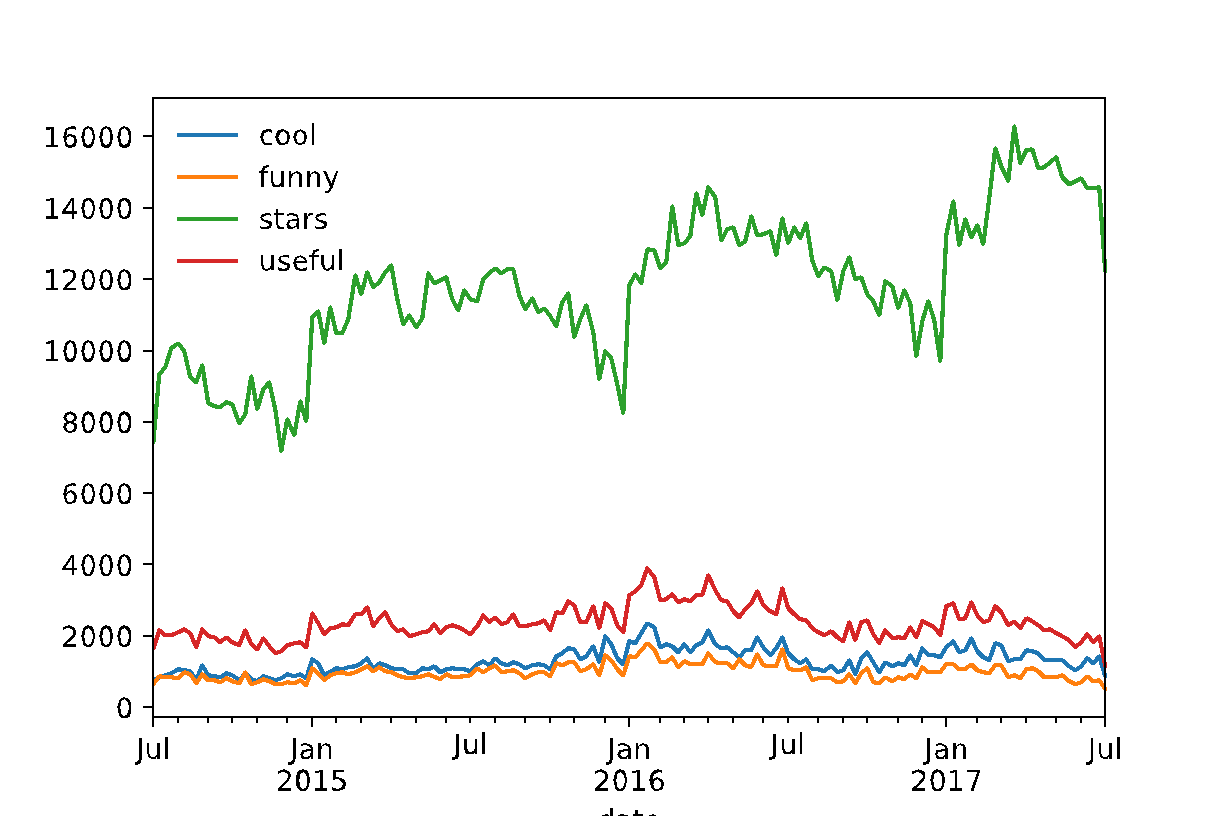


Fig 3.b: Time Series Analysis (Monthly Analysis)

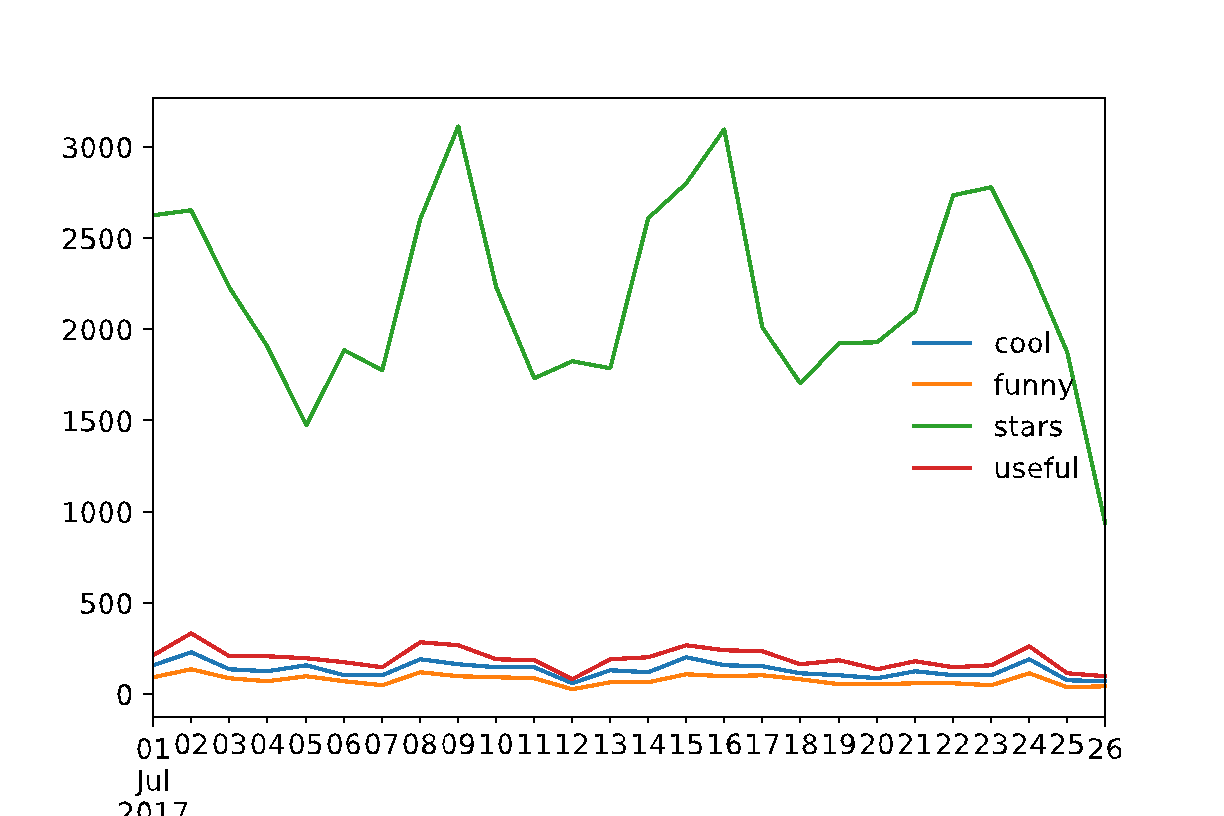


Fig3.a: Time Series Analysis (Weekly Analysis)

**III.1 Time Series Analysis**

We found that the restaurants in Arizona have been reviewed from February 2005 to July 2017. We tried to find weekly, monthly, and annual trends of reviews given to restaurants.

We found that most of the restaurants received maximum number of reviews every Sunday, and minimum number of reviews during Tuesday –Thursday. It might be explained by the fact that users are most likely to find time for reviews over the weekends. . Fig 3.a shows the weekly trend. Similarly, we found restaurants received minimum number of reviews in the month of January and maximum number of reviews in the month of April. Fig 3.b shows the monthly trend in number of reviews. From fig 3.c we observe that number of reviews in restaurants have increased significantly from 2005 to 2017. Increasing popularity of Yelp over time could be one of the reasons for this change.

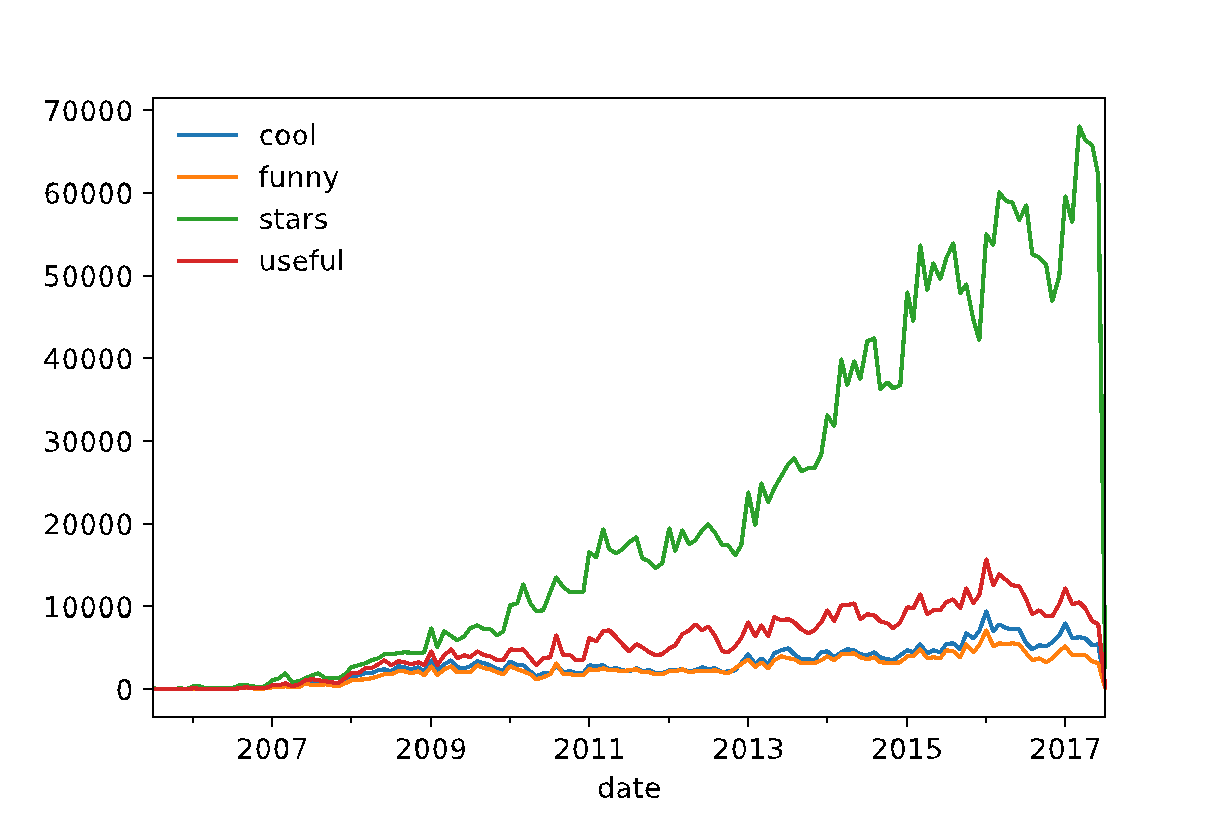


Fig3.c: Time Series Analysis (Yearly Analysis)

We plotted the star rating vs total number of reviews a restaurant has received and we got the distribution as shown in fig 4.

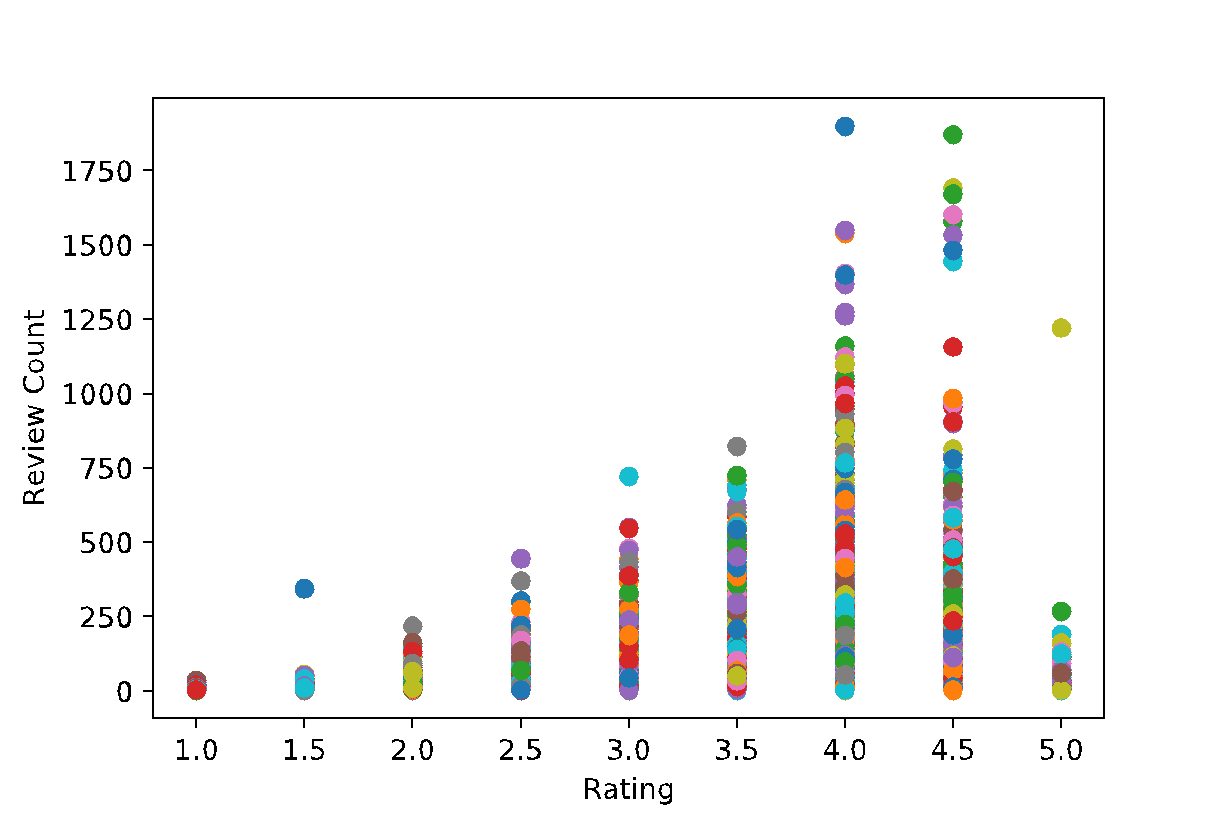


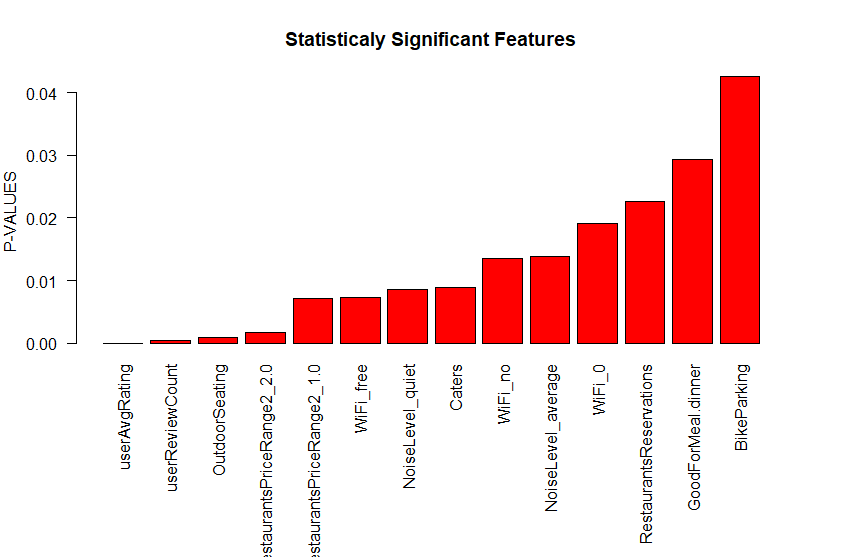
Fig 4 Rating VS Review Count

1. **Feature Based Models**

**IV.1 Feature Selection :**

We used the review.json and business.json files for feature selection. The review .json  contains  fields like “user\_id” , “business\_id”, “review\_text” etc. along with the associated ratings. It also consists of some categorical fields like “funny”, “cool”, “useful” etc. We however didn’t use these categorical fields in our Models because most of these fields were either extremely skewed or very  sparsely populated with many null entries. In order to rule out the significance of these categorical variables , we performed chi-square testing and as expected, we found these features to be highly insignificant at a type I error rate of 0.05.

A lot of attributes from the  business.json file were also used in our prediction. Most of these predictors were One Hot Encodings of several categorical variables like "BusinessParking", "GoodForMeal.lunch", "Alcohol\_beer\_and\_wine", "HasTV", WiFi\_free etc. A few continuous valued predictors were also fed to these models like the "userAvgRating" , "userReviewCount" etc. Overall, a total of 38 predictors were  from both review.json and business.json were used in our models. The chart below represents the order of Significance 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 . We also found Business features like “OutdoorSeating”, “WiFi\_free”, “NoiseLevel\_quiet” to be highly significant predictors of the ratings.



**IV.2 Models :**

The 200K data was randomly split into 150K training and 50K testing sets for all the Feature Based Models below. Grid Search was used to tune the hyper-parameters of all these Models.

We performed mean normalisation and feature scaling for all our Models.

**IV.2. 1. Linear Regression:**

We used the Ordinary Least Squares (scipy.linalg.lstsq)  Regressor with fit\_intercept parameter  set to True. We obtained an MSE of 1.23127597354 and an RMSE of 1.1096287548271544 on our test set for our Linear Regression Model.

**IV.2. 2. Ridge Regression :**

We used the standard Linear least squares with l2 regularization for our Ridge Regression Model with regularisation parameter lambda = 1.0 and fit\_intercept parameter set to True.

We obtained an MSE of 1.23127931759  and an RMSE of 1.109630261660905 on our test set for our Ridge Regression Model.

**IV.2. 3. Support Vector Regression (SVR):**

We used the standard libsvm based Epsilon-Support Vector Regression Model with  C set to 1 and epsilon set to 0.1. Rbf kernel and degree 3 polynomial was used for the fitting.

We obtained an MSE of 1.26881663173 and an RMSE of 1.1264176098262875 on our test set for our Support Vector Regression

**IV.2. 4. Random Forest Regression :**

We used the sklearn Random Forest Model with no. of trees i.e. n\_estimators  as 10 . We chose the optimisation metric as mse with minimum split i.e min\_samples\_split  set to 2 and bootstrap set to True. We obtained an MSE of 1.43051313003 and an RMSE of 1.1960406055105643 on our test set for our Random Forest Regression.

**IV.2. 5. AdaBoost Regression :**

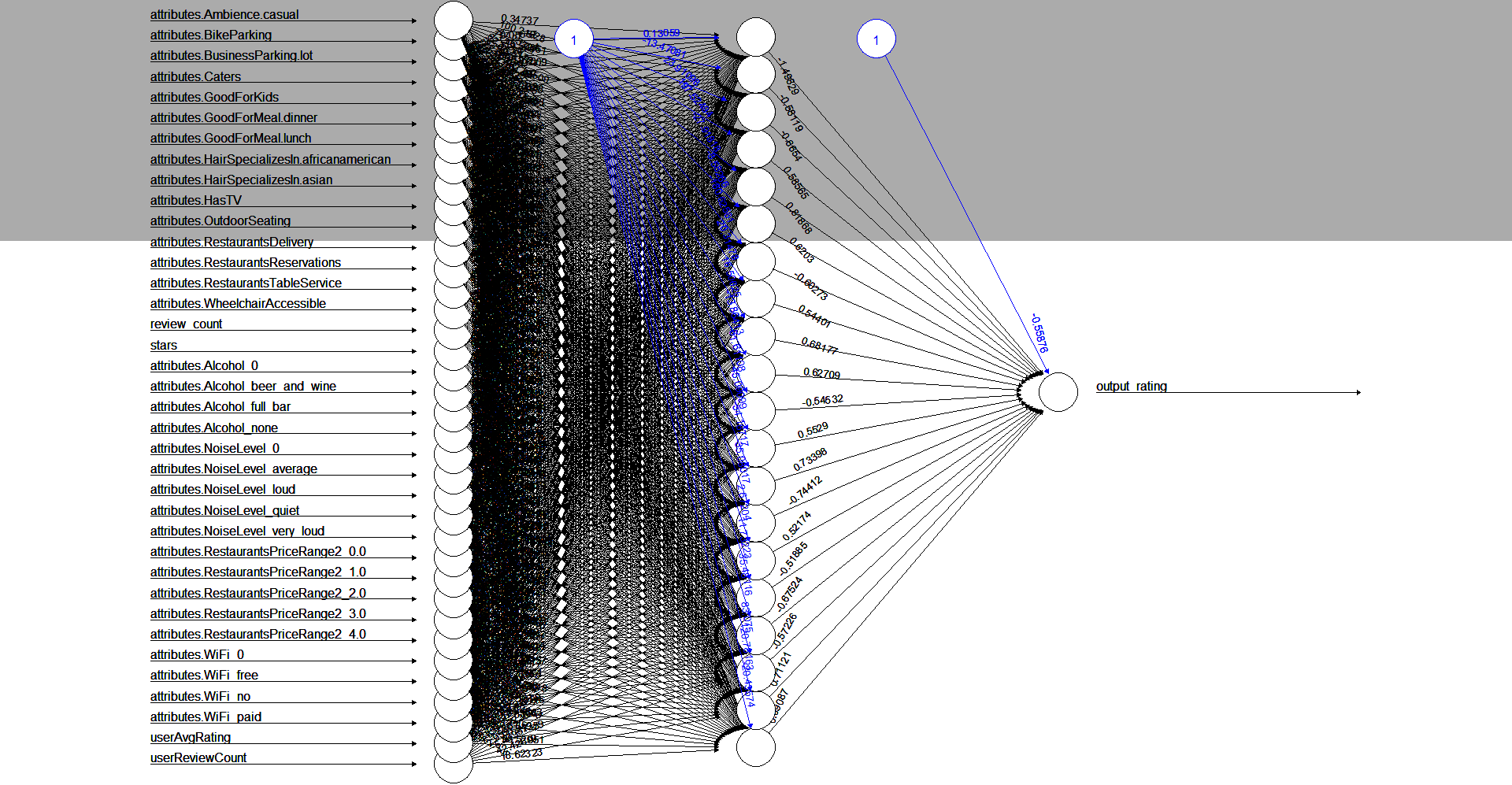
We used the AdaBoost Regression model with base\_estimator set to DecisionTreeRegressor, no. of trees i.e. n\_estimators set to 50 and a learning\_rate of 1. We used the square loss function for the fitting.We obtained an MSE of 1.35814161741 and an RMSE of 1.1653933316308949 on our test set for our Random Forest Regression.

**IV.2. 6. Neural Net Regression :**

We also tried a Neural Net Regressor on the Model. The Plot below depicts the Neural Net

Architecture using resilient backpropagation with 1 hidden layer consisting of 20 neurons.

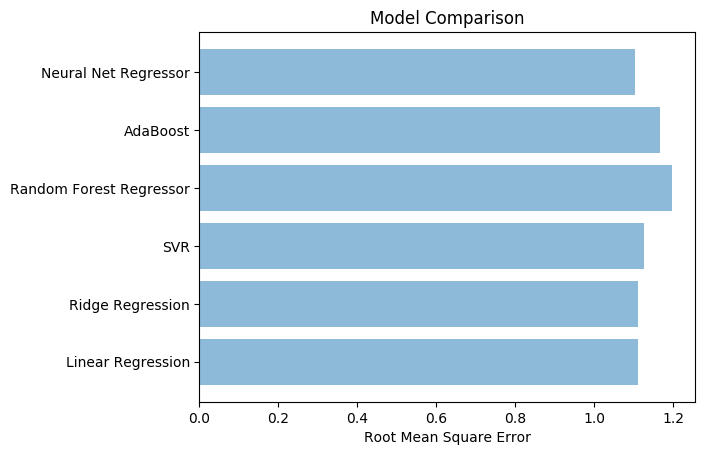
In order to avoid overfitting, we used a drop-out rate of 50%. We obtained an MSE of 1.21821283799  and an RMSE of 1.1037267949948337 on our test set for our Neural Net Regression.



**IV.3 Comparison of Feature Based Models :**

Surprisingly Ridge Regression and Linear Regression outperformed more complicated Nonlinear Models like SVR (with a polynomial kernel of degree 3), Random Forests and AdaBoost based on the RMSE obtained on the Test set.This might probably be attributed to the problem of Over-Fitting of Nonlinear Models.

This bar plot below shows the Root Mean Squared Error (RMSE) obtained by trying a bunch of different Linear and Nonlinear Models like Linear Regression, SVR, Ridge Regression, Neural Nets as described above.  As evident from the Bar Plot above, Neural Net outperformed every other model with an RMSE of 1.1037267949948337 on the test set.



1. **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.

**V. 1 Word Clouds**

We start by plotting wordcloud of all 5 star reviews and 1 star reviews to check what positive and negative aspects of restaurants have been discussed by the users. Fig 5.1 shows the wordcloud of all 5 stars 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 staffs, fresh and delicious foods, great service etc have positive connotations and thereby are a strong indicator of higher ratings.



Fig 5.1: Word Cloud of all 5 Star Ratings

Likewise , the wordcloud of 1 star ratings as shown in Fig 5.2 gives us some insights about the attributes that tends to be associated with poor ratings like negative adjectives, and some nouns such as “never”, ”time”, “service”, ”food” etc. It can be inferred that these restaurants lack in some of the properties like food quality, staffs behavior, punctuality, service quality etc, and they need to improve on these properties in order to receive higher ratings.



Fig 5.2: Word Cloud of all 1 Star Ratings

**V.2 Models**

For prediction tasks, we used 200k randomly shuffled reviews data from Arizona as training data, 150k reviews data as validation data, and next 50k reviews data as test data.

**V.2.1 Classical Latent Factor Model:**

The basic latent factor predicts the rating of the restauarnts according to the given equation.

*<Equation Here>*

<Explanation>

Where <alpha> is a global bias term, <beta\_u> is user bias term, and <beta\_i> is a restaurant bias term. <Gamma\_i> can be interpreted as the attributes of restauarant i, and <Gamma\_u> can be interpreted as the preference of user ‘u’ towards those attributes.

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.

**V.2.2 Latent Dirichlet Allocation (LDA)**

Latent Dirichlet Allocation (LDA) is a generative statistical model for text, which is used as a topic model to discover hidden features from text reviews. In LDA, each document ‘di’ 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 1 shows few topics that we extracted by running LDA on the entire training data set. For example, *Staff* topic 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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Staff | Service Time | Sea Food | Menu | Cleanliness |
| order | minute | Sushi | menu | bathroom |
| manager | order | roll | Wine | dirty |
| service | Time | fish | restaurant | floor |
| employee | came | tuna | dish | parent |
| staff | table | salmon | item | movie |
| owner | service | fresh | course | coke |
| counter | ordered | Japanese | meal | clean |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Payment | Group | Desserts | Food (2) | Beer | Night  Time |
| card | table | cream | pizza | beer | late |
| dim | room | desserts | italian | selection | night |
| cash | dining | chocolate | pasta | nacho | club |
| sum | experience | ice | crust | tap | hit |
| credit | reservation | cake | sauce | brew | open |
| stated | group | sweet | cheese | bar | scottsdale |
| continued | party | strawberry | gluten | local | miss |

Table 1: Topics Extracted From Reviews by LDA

Let’s 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”, “Visit”. It’s clear from the topics that in order to improve the ratings, the restaurant owner must try to improve attributes like Room condition, Service, staying place etc.

**Model 2**:

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

<Equation here>

Where <alpha> is a global bias term, <beta\_u> is user bias term, and <beta\_i> is a restaurant bias term. <Gamma\_i> can be calculated by taking list of all reviews for a particular restaurants and then using LDA to get a K-dimensional stochastic vector per restaurant(<di>). Each element of <di> consists of probability of that topic associated with the reviews of <ith> restaurant . <Gamma\_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 don’t update <di> after we initialize it with stochastic vector for each product.

**Model 3 :**

We also experimented by combining <gamma\_i> and <di> from above 2 models in order to extract hidden attributes of restaurants from both rating dimension and review text dimension.

<Equation here>

where the symbols have their usual meanings.

We used the following update rule for <gama\_i> and <gamma\_u>:

**V.2.3 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 2. The table 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.

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

**V.3. Comparison of Latent Models :**

|  |  |
| --- | --- |
| **Models** | **RMSE** |
| Latent Factor Model (LFM) | 1.268 |
| Latent Dirchlet Allocation (LDA) | 1.284 |
| LFM + LDA | 1.286 |

1. **Overall Results and Comparison across Feature Based and Latent Factor Models**

|  |  |
| --- | --- |
| Models | 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 Dirchlet Allocation (LDA) | 1.284 |
| LFM + LDA | 1.286 |

Table 3: Overall Model Comparisons

**<We used \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ for training \_\_\_\_\_\_\_\_\_\_\_for validation, and \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ for test. >**

The table shows that Regression Models showed better performance than Latent Factor Models. One of the reasons could be the selection of explicit fine grained features from business.json file.

**VII. Conclusion :**

In this report, we have analysed 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 (200K data points ) of the entire data (5 million 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 regularisation. In order to test this, we trained a Neural Net Model with a high dropout rate of 50%  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.

**VIII. Future Work**

One interesting follow-up of our analysis would be see whether increasing the number of Training Data from 150K to 5 million would allow 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.

**IX. 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*, (8):30–37, 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 collab- orative filtering model. In *Proceedings of the 14th ACM SIGKDD in- ternational conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[8] J. 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 897–908. International World Wide Web Conferences Steering Com- mittee, 2013.

[9] Blei, Latent DirchLet Allocation

[10] Netflix update: Try this at home. http://sifter.org/ simon/journal/20061211.html.

[11] https://anaconda.org/anaconda/gensim