Text Mining on Yelp Data

Bhanu Renukuntla
A53094633
brenukun@eng.ucsd.edu

Vikas Sakaray
A53090014
vsakaray@eng.ucsd.edu

Varun Bhatt
A53093209
vdbhatt@eng.ucsd.edu

ABSTRACT

In this paper we investigate a variety of regression and classification models to predict rating from given review text. It was found that Ridge regression with TFIDF unigram features worked the best in terms of Mean Squared Error. We investigate the model performance on data from different cities and conclude that our model is robust to changes in rating distribution. Also, we show that the text based models should be used with caution across regions since they are sensitive to language. Additionally, it is observed that it may be better to use different models for different time periods (seasons or years). Visualizing the word clouds give valuable insights about region, culture as well as its correlation to seasons and ratings.

Keywords: Rating Prediction, Sentiment Analysis, Text Mining, SVM, Ridge Regression, Gradient Boosted Regressor, PCA

1. INTRODUCTION

In the age of big data, businesses turn to data driven approaches to take business decisions. These usually involve prediction, data analytics and market estimation tasks. Understanding the user behavior and catering to their needs has been one of the most crucial goals of the any business. In the the past decade with the bloom in the ecommerce and breakthroughs in machine learning, businesses have now turned to machine learning to meet their demands as well as grow their business. This has become indispensable for companies like Yelp, one of the most popular business rating and review website, which solely rely on online user interaction. Yelp dataset is very diverse and we would like to explore the dataset to formulate and solve interesting predictive tasks inline with the Yelp dataset challenge 2017.

2. LITERATURE REVIEW

In the past decade, large-scale recommendation datasets from Netflix [1], Amazon [2], and other similar data including that released yearly by Yelp as a part of an open challenge have been extensively studied. Attempts have been made to gauge information from the review text by predicting user preferences with regard to various aspects of the business. The purpose behind most of the studies has been to predict the rating of a business/product and further use the hidden user and product/business features to create recommendation systems. Another similar dataset which has been studied in past is Yahoo! Music [3] which models music ratings and recommendations. To this end, a lot of models have been proposed and used successfully to solve the predictive and recommendation tasks. Collaborative Filtering methods utilize user feedback to infer relations between users, between items, and ultimately relate users to items they like [4]. Context based filtering methods characterize items based on textual attributes, cultural information, social tags and other kinds of web-based annotations [5]. Hidden topics obtain highly interpretable textual labels for latent rating dimensions, which helps to 'justify' ratings with text. There have been several other well-established methods like Matrix factorization, Latent Factor Modeling, and Human Annotation Methods [3, 6-7].

Our aim being to gauge information from the review text for prediction purposes, we focus on the models related to text mining. The impact of text derived information has been previously studied at both word and sentence level, with the help of the topic information on various datasets. Some simple models include using n-grams from the text corpus to generate features. Latent topic modeling is very widely used as an unsupervised model for clustering and classification in machine learning areas. LDA (Latent Dirichlet allocation) is a common method of unsupervised learning to discover hidden topics. It assumes that there are latent variables that reflect the thematic structure of the documents [8]. Another common

topic modeling method is probabilistic latent semantic analysis (PLSI) [9]. HFT (Hidden Factors Topics) combines latent rating dimensions and latent review text dimensions, which results in more interpretable topics and more accurate rating predictions at the same time [10].

The most popularly used TFIDF technique (Term Frequency–Inverse Document Frequency) is intended to reflect how important a word is to a document in a collection or corpus. It is used as a weighing factor in information retrieval and text mining. The TFIDF value increases proportionally to the number of times a word appears in the reviews but is offset by the frequency of the words in the corpus, which helps to adjust for the fact that some words appear more frequently in general. Words with high TFIDF numbers imply a strong relationship with the reviews they appear in, suggesting that if that word were to appear in a review, the review could be of interest to the user [11-12].

3. DATASET EXPLORATION

3.1 Dataset Selection

We use a portion of the dataset from the ninth round of Yelp data challenge which includes data from over 29 countries and 878 cities all over the globe. The data is segregated into 6 files and statistics are listed in Table 1.

Type	#Records
Business	144072
Reviews	4153150
Users	1000000
Tips	946600
Checkins	125000
Pictures	200000
Cities	878
Countries	29

Table 1: Raw DataSet

For the purpose of the project, we limit ourselves to data from 4 cities from 4 different countries: Pittsburgh, Montreal, Edinburgh and Stuttgart. Also, we use only business and review data. This leaves us with 309,746 reviews from 15,616 businesses for sentiment analysis and prediction tasks. See Table 2 for further details. Note that the reviews from Stuttgart are predominantly in German , while a good portion of those from Montreal are in French although most reviews are in English.

City	#Reviews	#Business Records
Pittsburgh	143118	5275
Montreal	96510	4785
Edinburgh	45482	3601
Stuttgart	24636	1955

Table 2. Data used

```
{
    "review_id":"encrypted review id",
    "user_id":"encrypted user id",
    "business_id":"encrypted business id",
    "stars":star rating, rounded to half-stars,
    "date":"date formatted like 2009-12-19",
    "text":"review text",
    "useful":number of useful votes received,
    "funny":number of funny votes received,
    "cool": number of cool review votes received,
    "type": "review"
}
```

Figure 1. Sample review json

3.2 Rating Distribution

To get a better understanding and develop insights to formulate predictive tasks with the data, we performed an elaborate exploratory analysis.

Fig. 2 shows the distribution of review star rating over the complete dataset. The distribution is rather skewed, and there are far more reviews with high ratings than those which received a lower rating.

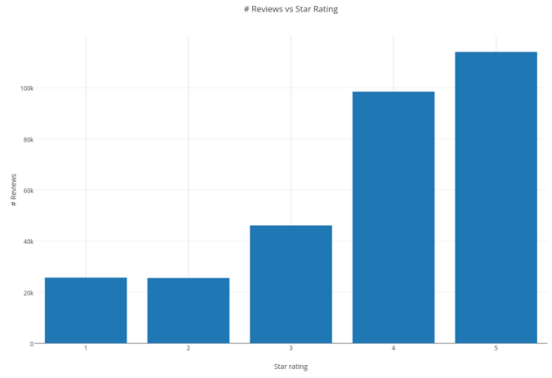

Figure 2. Number of Reviews vs Star Rating

3.3 Temporal Analysis

3.3.1 Over the years

From Figure 3, we can note a monotonic increasing trend in the number of reviews on Yelp over the years since it was launched. However, the biggest two jumps are in years 2010 and 2015. Yelp launched it's mobile app in 2008 and expanded in Europe in 2014. These could be the reasons for the sudden jump in Yelp reviews during the following years.

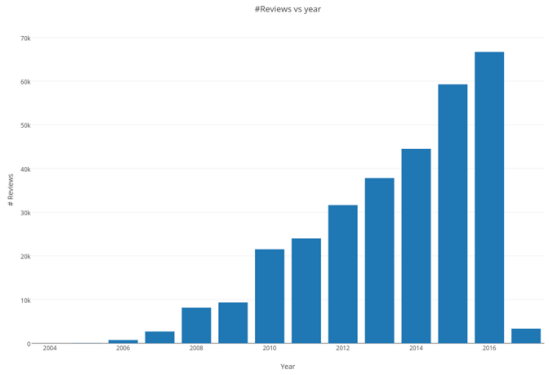

Figure 3. Plot of number of reviews vs year.

As for the average ratings, there seem to be a higher ratings during the 2000s as compared to those in the following decade. (Fig. 4)

3.3.2 Over the seasons

The temporal distribution over the months can give insights about variation in reviews and ratings according to season i.e. Fall, Winter, Spring and Summer. This is can be a crucial factor for certain kinds of businesses.

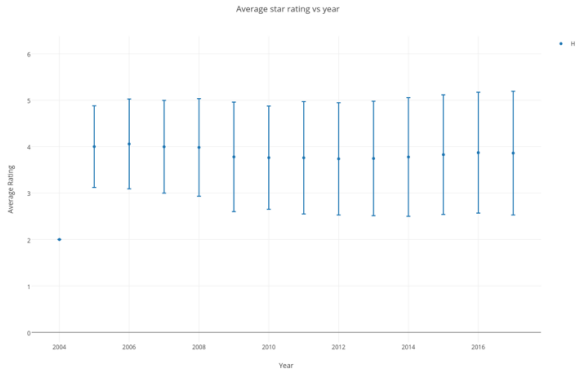

Figure 4. Plot of average user rating vs year.

Fig 5. shows the number of reviews for different months. It's interesting to note the increasing trend from spring through summer, with July having the highest reviews. However, the number of reviews are lowest for the Winter months, with January being an exception which we believe is due to the holiday season.

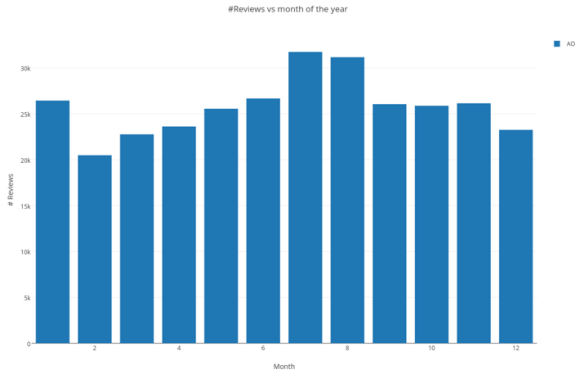

Figure 5. Plot of number of reviews vs month.

The average star rating distribution over the months in plotted in Fig. 6. Once again, we see relatively higher ratings during the holiday season as well as the months building up to it. November has the highest average star rating of 3.835, whereas September has the lowest (3.779).

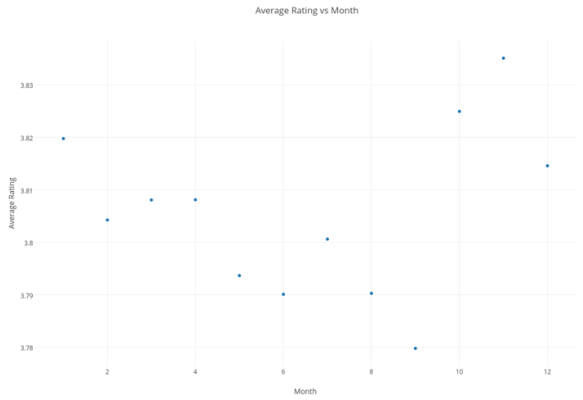

Figure 6. Average Rating vs Month

3.4 Geographical Analysis

Our dataset is based on four different cities from four different counties. This motivated us to analyze how sentiments of people change across different cities which we can further utilize to answer certain question specific to a geographical/cultural region.

Fig. 7 helps in understanding the distribution of star ratings within each city and explain the variation in average ratings in the previous plot. Grouping the reviews based on cities, we see that Edinburgh has relatively higher proportion of positive ratings and Pittsburgh has the highest

proportion of negative ratings. In turn, Pittsburgh has the lowest average rating, and Edinburgh the lowest.

3.5 Wordclouds

We use wordclouds to visualize the sentiments from the reviews. The word clouds based on word frequency for review texts with ratings 1 and 5 have been shown in Fig. 8. These represent the most positive and most negative sentiments in our review corpus. While those generated using popular unigrams aren't very informative, the wordclouds based on bigrams and trigrams reflect the reviewer sentiment clearly. Bigrams such as "10 minutes", "15 minutes" and "20 minutes" illustrate dissatisfaction among reviewers who had longer waiting times.

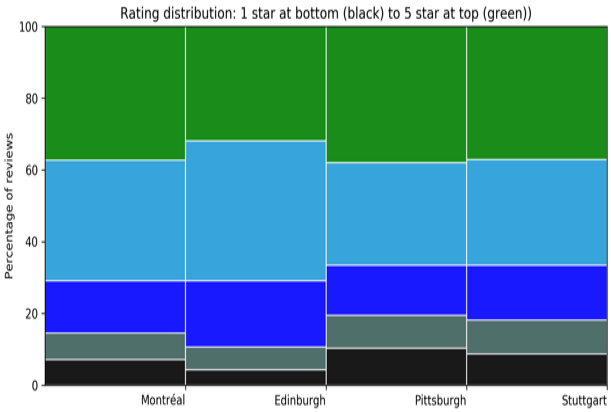

Figure 7. Star rating plot across cities

(a) Rating 1 Unigrams (c) Rating 1 Bigrams (e) Rating 1 Trigrams

(b) Rating 5 Unigrams (d) Rating 5 Bigrams (f) Rating 5 Trigrams

Figure 8. Wordclouds for 1 and 5 star reviews

Some other popular bigrams with negative sentiments are "go back", "will never", "even though". Reviews for rating 5 clearly talk about the positive aspects of the business: "highly recommend", "really good", "come back" and "love place". Many popular trigrams seem to stem from the most popular bigrams. However, certain trigrams such as "give zero stars", "worst service ever" manage to capture more extreme negative sentiments.

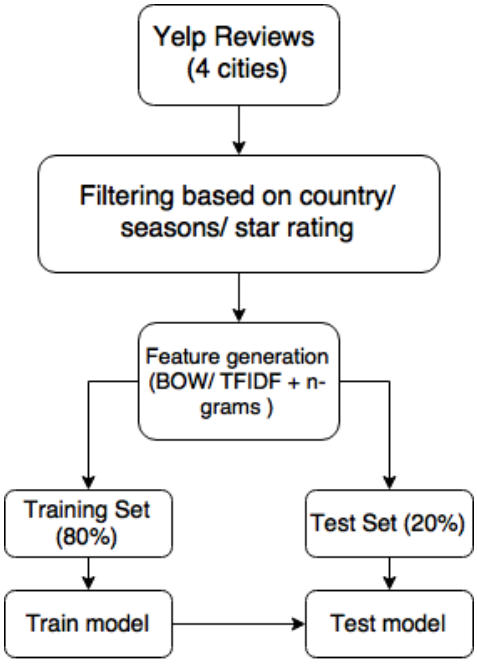

Figure 9. Flowchart of our workflow

4. PREDICTIVE TASKS

Our primary goal is to predict the rating using the given review information. From the tasks carried out in the data exploration section, we figured that the there's a considerable variation in the average rating for different months of the year. In fact, there's a significant difference in the average rating for reviews before 2010 and 2011 onwards.

Also, the distribution of ratings is not the same for the 4 cities, with a close agreement only for Montreal and Stuttgart. Another task which seemed intuitive to us was to investigate if we could employ dimensionality reduction methods on certain features in the review data. Specifically, the cool, funny and useful votes of a review could possibly contain redundant information. This warrants the usage of dimensionality reduction procedures. Further we are interested in the task of clustering of restaurant categories from the business records. Motivated by these observations, we consider the following predictive tasks:

1. Rating prediction using review text
2. Comparing cultures across countries
3. Geo-temporal sentiment analysis
4. Temporal sentiment analysis
5. Dimensionality reduction

All of these can be cast as machine learning problems. This is discussed in detail in the next sections.

5. METHODS

5.1 Features

First the review text is preprocessed by removing capitalization, punctuation and stop words. This is then used to generate features to train the models. The following features were considered:

1. Bag of words (BOW) based on top 1000 unigrams
2. BOW based on top 1000 bigrams
3. BOW based on top 1000 unigrams+bigrams
4. TFIDF based on top 1000 unigrams
5. TFIDF based on top 1000 bigrams
6. TFIDF based on top 1000 unigrams+bigrams

For the dimensionality reduction task, we use cool, useful and funny features of only those reviews which have at least one non-zero entry.

5.2 Model and Evaluation

The rating prediction task can be solved using regression as well as classification techniques. We considered a variety of models (Ridge Regression, Gradient Boosted Regressor, Neural Network Regressor, SVM Classifier, Random Forests) and have picked those which gave the best results for this report.

We hypothesize that regression based methods are better suited for our task than classification. The intuition behind this is wrongly classifying a 5 star review as 1 star would contribute more to the error than the regression estimate.

5.2.1 Models

0. Baseline

We use the average rating of reviews in the training set as the baseline rating predictor.

1. Regression

a. Ridge Regression

This is a linear model and the estimated rating is a linear transformation of the feature. An additional regularization term is introduced to the loss function at the time of training to account for model complexity and reduce overfitting. The

data in the training set is first filtered appropriately based on the relevant task (geographical/ temporal) using the features listed in the previous section. The value of the regularization parameter is selected using 5-fold validation. We observed that with all models, training using TFIDF based features seemed to give better results. Additionally, unigram based features gave the lowest MSE.

b. Gradient Boosted Regression

The Ridge Regression model gave a phenomenal improvement over the baseline. However, we didn't stop here but decided to look further for a better model. Ensemble based methods (which use a collection of estimators to predict the output) in general seem to give good results. Gradient Boosted Regression (GBR), a flexible non-parametric statistical learning technique, is one such method. Although we expected GBR to give a more accurate prediction, our results did not reflect this.

2. Classification

a. Support Vector Machine Classifier

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. Based on the features generated using the review text, the SVM classifier predicts the rating by classifying it into one of 5 classes.

Evaluation Metric

Mean Squared Error (MSE): The machine learning models used for the above tasks are trained to minimize the Mean Squared Error.

3. Dimensionality Reduction

a. Principal Component Analysis (PCA)

Principal component analysis uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance (also the largest eigenvalue) and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are an uncorrelated orthogonal basis set. PCA can be used for dimensionality reduction by discarding the components along directions with low variance (those along eigenvectors associated with smallest eigenvalues).

Evaluation Metric

Percentage Variance Explained: This is the fraction of variance explained by the eigenvectors that are used for dimensionality reduction.

b. Hierarchical Clustering

We consider the task of clustering the business category tags. For this task we only consider the businesses which are tagged as 'Restaurants'. There are about ~48K business records and 607 unique tags. To solve this problem, we define a feature vector representation for each tag which will help us in defining a distance measure in 'tag' space. First we populate a symmetric matrix where rows and columns both represent tags and the element in location (i,j) is the number of records in which both 'tag-i' and 'tag-j' occur together. Now each row can be considered as feature vector for the corresponding tag. Since the number of occurrences of different tags is much skewed, we need to standardize the components of the feature vector. By design, this feature representation works well for bottom up clustering methods. Hence we use agglomerative clustering techniques, more specifically, we use ward's method (for evaluation) to solve this task.

6. RESULTS AND DISCUSSION

We train the models described in the previous section using various features and report the best model. The data is split into 70 : 20 : 10 sets for training, validation and test respectively for every city. We also maintain a similar rating distribution for each of the sets. For each model we using 5-fold cross validation to select model parameters wherever necessary. We select the best model based on validation MSE and use this model for all the remaining experiments.

6.1 Rating Prediction

The problem of rating prediction based on review text is considered in this task. We perform our experiments on each city independently and report the validation MSE (see Table 3). We pose this task as classification problem and as a regression problem. We hypothesize that regression based approaches work better than classification and we experiment primarily using regression based models. Our hypothesis is justified and backed up by our results, Ridge

Model	Feature	Edinburgh Validation MSE	Stuttgart Validation MSE	Montreal Validation MSE	Pittsburgh Validation MSE
Baseline		1.128	1.632	1.438	1.749
Ridge	BOW+Unigram	0.744	1.022	0.832	0.927
	BOW+Bigram	0.923	1.464	1.120	1.226
	TFIDF+Unigram	**0.668**	**0.865**	**0.710**	**0.759**
	TFIDF+Bigram	0.906	1.131	1.047	1.137
GBR	TFIDF+Unigram	0.865	1.219	0.977	1.099
	TFIDF+Bigram	1.06	1.537	1.301	1.529
SVM	BOW+Unigram	0.928	1.351	0.973	1.042
	TFIDF+Unigram	0.909	1.227	0.915	0.936
		Test MSE	**Test MSE**	**Test MSE**	**Test MSE**
Ridge	TFIDF+Unigram	0.667	0.864	0.720	0.767

Table 3. Models and MSE for different cities

Regressor, Gradient Boosted Regressor(GBR) works better than SVM classifier. It can be observed that every model outperforms the baseline quite comfortably. We found that TFIDF features work better than vanilla Bag Of Words features and to our surprise, Ridge Regressor outperformed Gradient Boosted Regressor. Ridge Regressor with TFIDF unigram features is the best model.

6.2 Comparing Cultures Across Cities

Table 3 compares MSE across different cities. As each city is very different i.e., different country, different language and different culture, we train every city independently and compare MSE on validation set. Another reason for doing this is that rating distribution is different for each city. Note that baseline for Pittsburgh is quite high compared to other cities. But we observe that Ridge Regressor with TFIDF unigram outperforms other models across cities. Contrary to our belief, we observe that the relative performances of different models is same across different datasets (which have different rating distribution). This is an indicator of generalizability of the model.

Finally, we train the best model on Pittsburgh and calculate MSE on other cities. As we are performing text based analysis, language plays a pivotal role in model's performance on new test corpora much alike how it plays a role as a part of people's culture. This is reinforced in Table 4. where it is clear that our model generalizes very well for English speaking cities but performs poorly on Stuttgart. It performs reasonably well on Montreal since only a few reviews are French.

City	Baseline	MSE Pittsburgh	Best MSE
Edinburgh	1.128	0.698	0.668
Montreal	1.438	0.769	0.710
Stuttgart	1.632	**1.580**	0.865

Table 4. Model comparison

Word clouds for different cities have been shown in Figure 10. We observe a few French words in wordclouds for Montreal. Similarly, German words are dominant for Stuttgart. Also, note that the bigram 'good coffee' is very dominant in Montreal but not in other cities. We observe similar bigrams for other word clouds, not shown here, which is indicative of different culture. For all the experiments here on, we use Ridge Regressor with TFIDF unigram features as our

model to report performance on test set.

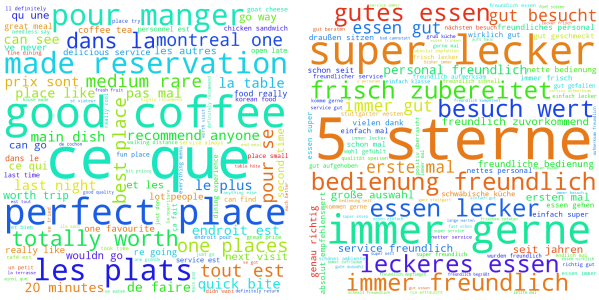

Figure 10. 5 star wordclouds Montreal & Stuttgart

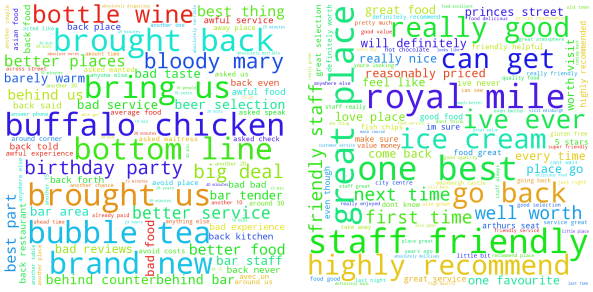

Figure 11. 5 star Spring Pittsburgh (left) & Summer Edinburgh (right)

6.3 Geo-temporal Sentiment Analysis

In this section we compare the performance of the models trained independently across seasons/ quarters for each city. We observe a similar trend for all four cities: MSE of Summer, Winter is lower than other two and MSE for Spring, Fall are quite close. We observe that the MSE is quite variable across different seasons. (Table 5)

City	Winter	Spring	Fall	Sum
Pittsburgh	0.746	0.780	0.769	0.756
Montreal	0.686	0.732	0.733	0.706
Edinburgh	0.653	0.670	0.672	0.659
Stuttgart	0.825	0.873	0.881	0.882

Table 5. MSE across seasons

Wordclouds in Fig. 11 reflect Geo specific attributes over seasons for 2 cities. Specifically, in Summer-Edinburgh people seem to enjoy ice cream. On the other hand in Spring-Pittsburgh there seems to be a jolly food and drinks atmosphere (wine, chicken and what not).

6.4 Temporal Sentiment Analysis

We observed that the rating distribution is quite different for reviews in 2000s and reviews in 2010s. Note that the baselines (variance in this case) for 2000s and 2010s are also quite different. These reasons motivated us to train different models for review in 2000s and reviews in 2010s. Table 6 shows MSE's for both.

Year	Baseline MSE	Best MSE
2000s	1.181	0.747
2010s	1.599	0.729

Table 6. MSE for 2 decades

6.5 Dimensionality Reduction

6.5.1 PCA

Each review, apart from the text, also has a different votes named 'useful', 'cool' and 'funny'. We believe these votes have a high degree of correlation and therefore we could try dimensionality reduction i.e., replace them with a weighted average of the three votes. This motivated us to investigate the dimensionality reduction aspect through PCA. We found that the 92.3% of the variance is explained by the first component of PCA and the corresponding eigenvector is [0.660, 0.612, 0.436]. Very high explained variance justifies the replacement of the 3 kinds of votes with a weighted average of votes.

6.5.2 Hierarchical Clustering

As we can see from Fig. 12, the algorithm we used is able to get the low level clustering right. For example, it groups most of Asian cuisines together. We can observe that most of the bars, pubs are together. Although the clades are right, but the relation between the clades is not appropriate at many places.

7. SUMMARY

In this paper we investigated a variety of regression and classification models to predict rating from given review text. It was found that Ridge regression with TFIDF unigram features worked the best (lowest MSE) across 4 cities we considered.We investigated the model performance on data from different cities and concluded that our is model robust to changes in rating distribution. Also, we showed that the text based models should be used with caution across

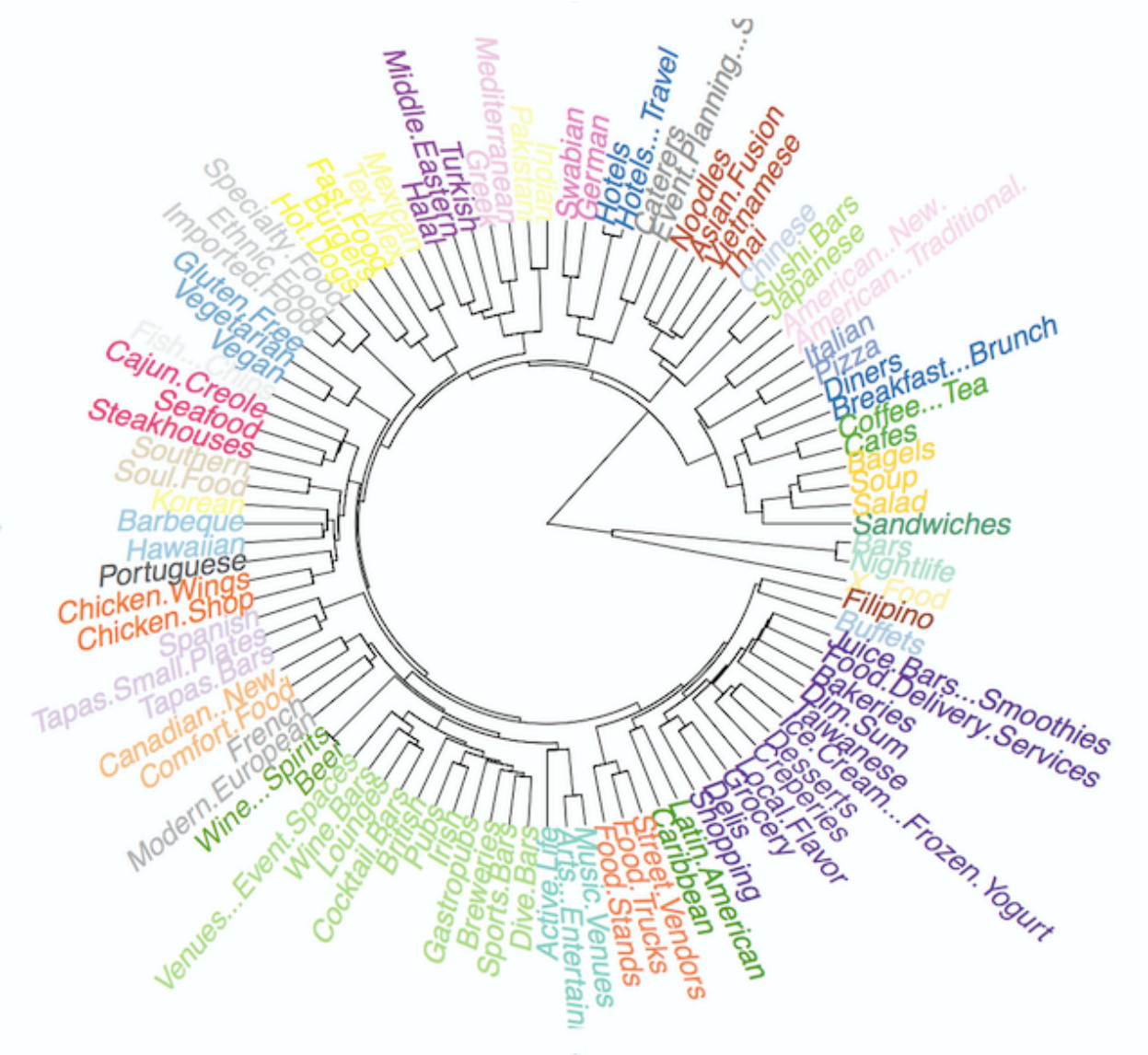

Figure 12. Hierarchical Clustering of Restaurant categories

regions since they are sensitive to language. Additionally, it is observed that it may be better to use different models for different time periods (seasons or years). Visualizing the word clouds gave us valuable insights about region, culture as well as its correlation to seasons and ratings.

8. REFERENCES

1. J. Bennett and S. Lanning. “The netflix prize.” In Proceedings of KDD cup and workshop, volume 2007, page 35, 2007
2. G. Linden, B. Smith, and J. York. “Amazon. Com recommendations: Item-to-item collaborative filtering.” Internet Computing, IEEE, 7(1):76{80, 2003
3. K. Noam, G. Dror, and Y. Koren. "Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy." Proceedings of the fifth ACM conference on Recommender systems. ACM, 2011
4. M. Bamshad, X. Jin, and Y. Zhou. "Semantically enhanced collaborative filtering on the web." Web Mining: From Web to Semantic Web. Springer Berlin Heidelberg, 2004. 57-76
5. P. Lamere. “Social tagging and music information retrieval.” Journal of New Music Research, 37(2):101–114, 2008
6. K. Yehuda, R. Bell, and C. Volinsky. "Matrix factorization techniques for

recommender systems." Computer 42.8 (2009)
7. D. Agarwal and B. Chen. "Regression-based latent factor models." Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009
8. D. M. Blei, A. Y. Ng, and M. I. Jordan. “Latent dirichlet allocation.” the Journal of machine Learning research, 3:993-1022, 2003.
9. T. Hofmann. “Probabilistic latent semantic analysis.” In Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence, pages 289-296. Morgan Kaufmann Publishers Inc., 1999
10. J. McAuley and J. Leskovec. “Hidden factors and hidden topics: understanding rating dimensions with review text.” In Proceedings of the 7th ACM conference on Recommender systems, pages 165–172. ACM, 2013
11. T. Joachims. “A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization.” No. CMU-CS-96-118. Carnegie-mellon univ Pittsburgh, Pa, Dept of Computer Science, 1996
12. J. Ramos "Using tf-idf to determine word relevance in document queries." Proceedings of the first instructional conference on machine learning. 2003