CAPP 30255 Project Midterm Report - FreshDash

Using Yelp reviews to optimize restaurant inspection

Ran Bi, Shambhavi Mohan, Minjia Zhu

I. Changes to Project Plan

There is no substantial change to our project plan. As we proceeded with the project, we had two in-depth meetings with the professor. Based on the feedback, we adopt more NLP libraries, decide on appropriate datasets to use, and refine our approach to the problem as described below.

Software / Libraries

We utilize more choices of NLP libraries than we listed out in the proposal that we've found to be useful.

- SpaCy is a newly adopted library we used to do tokenization, sentence recognition, part of speech tagging, lemmatization, dependency parsing, and named entity recognition.
- Gensim, among all Python NLP library, is one of the most well-optimized and specialized library to do topic modelling algorithm implementation

Datasets

Though Yelp Academic Dataset provides rich data of hundreds of cities, the volume of reviews per each city is limited. Moreover, different cities use different metrics in health inspection, and some cities do not publish their inspection records. Consequently, record linkage would be a very time-consuming and redundant task for our project. Therefore, we decided to use two readily-linked datasets, Seattle and Las Vegas, available from published academic studies, so that we can efficiently spend our time on sharpening NLP and machine learning skills.

Refine our approach

We were initially unsure about our two-stage machine learning approach: 1. Use NLP techniques to classify review texts; 2. Use the predicted classification as a feature to train the machine learning model. After discussion with the professor, we modified it as: 1. Use NLP techniques to extract text-based features from review texts; 2. Use text-based features together with other features to train the machine learning model. Additionally, we narrow down to adopt N-gram and LDA as main NLP methods to extract text features.

II. Related Works

Where Not to Eat? Improving Public Policy by Predicting Hygiene Inspections Using Online Reviews (Kang, Kuznetsova, Luca, Choi)

It is one the first works to use reviews as a predictive tool for assessing hygiene of restaurants. It uses SVM and SVR to predict restaurants with high unhygene scores (inspection health score >=50). One of the best things about the paper is that it uses first only individual features such as previous inspection scores, review counts, average rating, etc and shows that including review sentiment improves the prediction score to 82%.

Using Yelp Data to Predict Restaurant Closure (Michail Alifierakis)

In this blog report, the author uses yelp data to predict the probability of a restaurant closure. The author used both the yelp api and the google api to get details of all the restaurants that were open/closed as of

2013. The machine learning model used for this task was logistic regression optimised for precision of open restaurants. The final precision of open restaurants was 91%. This means that among the restaurants that are recognized as open by the model, 91% of them actually remained open. The remaining 9% are false positives. However, the model was not very good at predicting closure of restaurants. One of the suggestions was to use health inspection scores to improve the accuracy of predictions, which further cemented our decision to use both the yelp reviews and other features like previous inspection records to predict probability of a restaurant being a high health inspection scorer.

Predicting Restaurant Health Inspection Penalty Score from Yelp Reviews (Uppoor, Balakrishna)
This paper penalty score as a sum of minor, major and severe violators. This paper is also inspired by *Kang, Kuznetsov,etc* paper and uses ridge regression to come up with the best model to predict restaurant health inspection penalty score. Few things that we are going to use from this paper include the size of the word will be directly proportional to the the Ridge regression coefficient for that word. Hence, larger the word, higher the penalty. We will also consider global average penalty score as our base case.

III. Current Progress

Data Exploration

We use two datasets that integrate restaurant information on Yelp with health inspection result, one for Seattle and one for Las Vegas.

Seattle dataset is obtained from paper *Where Not to Eat? Improving Public Policy by Predicting Hygiene Inspections Using Online Reviews*. (available at http://cs.stonybrook.edu/~junkang/hygiene). The dataset contains Yelp reviews written for restaurants in Seattle over the period of 2006 to 2013, and public inspection records of the same period. The dataset contains more than 1,200 businesses, 13,299 inspections, and 162,310 reviews.

Las Vegas dataset is obtained from paper *Identifying Health-Violating Restaurants with Online Reviews* (available at https://sites.google.com/site/YelpRestaurantInspections/). The dataset integrates hygiene inspection records and Yelp reviews written for restaurants in Las Vegas over the period of 2005 to 2015. It contains more than 1,200 businesses, 17,000 inspections, 13,000 reviews.

We plan to use Seattle data as training and validation sets, and use Las Vegas data as test sets. Therefore, our data exploration focus on Seattle dataset. Following are some observations:

- Limited coverage of hygiene inspections geographically and seasonally. Geographically, more than 50% of the restaurants listed under Yelp did not have inspection records. The heatmap [Figure 1] of inspections also suggests unbalanced inspection efforts. Seasonally, though inspection instances increased from 700 in 2006 to 3,175 in 2012 [Figure 2], the pattern of inspection shows consistent seasonality over years – inspections were the most frequent in the first quarter, and the least frequent in the third quarter. [Figure 3]

- Review rating is not a good predictor of hygiene code violation. Following the convention in Kang et al. paper, we define an “inspection period” of each inspection record as the period of time starting from the day after the previous inspection to the day of the current inspection. If there is no previous inspection, then the period stretches to the past 6 months in time. Then we aggregate Yelp reviews generated during the inspection period by review counts and average rating. The histograms of review rating for those passed inspection and those with penalty score higher than 50 present very similar pattern. Additionally, we compute the Spearman's rank correlation coefficient for review rating and inspection penalty score at different cutoffs. The correlation seems insignificant and unstable over different cutoffs. [Figure 4]

- Review count is probably a weak predictor of hygiene code violation. Similar to review rating, we calculate the Spearman's rank correlation coefficient for review count during inspection period and inspection penalty score at different cutoffs. They seem to be positively correlated, with highest coefficient of 0.1 when cutoff is set at 30. [Figure 5]

Those observations justify the necessity of our project, i.e. using review content-based features to predict hygiene code violation.

Yelp Review Based Features - Top Frequently Occuring Keywords

In order to use yelp reviews as features, we decided to the top occuring keywords in all the reviews. To divide the review dataset, we assume that all reviews with an inspection penalty score of less than 50 are good reviews and those with a score of greater than 50 to be bad reviews.
We used the below methods to determine the keywords:

Ngrams

We first tried using unigrams to determine keywords related to good reviews and bad reviews. However, we find that unigrams is unable to take care of cases like "not great", etc. Hence we tried bigrams which resulted in better results. With bigrams we get sample keyword set for good reviews: ('much', 'better'), ('not', 'bad'), ('one', 'favorite'), ('little', 'bit'), ('late', 'night'), ('will', 'definitely) and for bad reviews as: ('long', 'time'), ('not', 'great'), ('go', 'wrong'). There are few overlaps between the two keyword sets, but we hope that the big sample feature set will help overcome the error caused by the overlap.

RAKE (Rapid Automatic Keyword Extraction):

Rake algorithm removes all stop words from the text and then creates an array of possible keywords. Then we find the frequency of the words and the degree associated with each word (the number of how many times a word is used by other keywords). Then the degree/frequency of each keyword is calculated and based on this score the best keywords are found. We find the words found using this method to be good too and we will create a set of features using RAKE too.

TF-IDF (work in progress)

Yelp Review Based Features - Topic Modeling

In order to mine hygiene information from Yelp reviews, we need to discern reviews commenting on health-related topics from other topics that are not useful in predicting hygiene code violation, for instance, cuisine authenticity, service quality, etc. Therefore, we use Latent Dirichlet Allocation (LDA) to achieve this goal by extracting the main latent topics from review texts.

We use review texts of Seattle restaurants as training texts and pre-process them with the following steps:

1. Tokenize the texts and remove punctuations using *gensim.utils.simple_preprocess* module;
2. Remove stop words, create bigrams and lemmatize tokens using gensim and spaCy libraries;
3. Create the corpuses and dictionaries required for LDA model training.

During model training phase, we underwent several iterations on appropriate training set before getting to a satisfactory modeling result. We started out with extracting topics from reviews with strong indicator of potential hygiene code violation, i.e. reviews with star rating less than 3 and inspection penalty score higher than 60. It turned out that the topic extracted were very close to each other, and it was hard to discern health specific items. We realized that the corpus itself was too narrow to extract meaningful subtopics.

Secondly, we decided to expand the training set to 162,310 reviews all together. The latent topics were well-distributed on the inter-topic distance map [Figure 6], but a close scrutiny of the subtopics revealed that they centered around restaurant type, service quality and comments on the food itself, without a clear subtopic on hygiene complaint.

Finally, we segmented reviews based on their star ratings in order to reveal underlying sentiment aspects of the reviews, which would be otherwise lost when all reviews were considered together. After some trial and error on different segmentation rules, we decided to separate all reviews into two groups: (1) positive reviews, i.e. reviews with 5-star rating (49,183 in total); and (2) negative reviews, i.e. reviews with 1-star rating (9,378 in total). Two LDA models were trained on respective review groups. The number of topics (k) was determined experimentally. We trained LDA models using k=20,30, and 40 and used coherence value to determine the optimal model. Interestingly, number of topics and coherence value correlated negatively for negative group [Figure 7] but positively for the positive group [Figure 8]. We tried k=15 and 20 respectively, and k=20 gave more human-interpretable results. So for now we used LDA model with k = 20.

Topic Modelling Results

Table 1 and 2 show the top words for selected topics. We have following observations:

1. It seems that LDA models well capture the negative words in 1-star reviews (e.g. depressing, hygiene, paper_thin, endless, grossly), and positive words in 5-star reviews (e.g. great, friendly, excellent, awesome).
2. In positive reviews, people rarely talk about cleanness of the restaurants, but more about dishes, service quality, and the environment. In negative reviews, people explicitly complain about hygiene of the restaurant.
3. Topics extracted from negative reviews seem less human-interpretable. Though the model successfully extracts hygiene-related topic, more work should be done to improve the distinction among topics and the weird downward sloping k-coherence score curve.

Figure 9 shows distribution of words for hygiene-related topic from LDA model trained on negative reviews.

Eventually we obtained 42 features for each review text, among them are 20 subtopics from LDA model trained on the positive reviews, 20 subtopics from the negative reviews, the dominant topic from the positive model, and the dominant topic from the negative model. The feature values for subtopic features are the percentage contribution of the corresponding subtopic. Since each inspection period contains multiple reviews, we need to figure out a reliable way to aggregate those features. For now, we use the percentage of dominant topic as aggregated feature. i.e. for each inspection period, there are 40 features (20 positive subtopics and 20 negative subtopics). The value of *positive (negative) topic i* is the count of reviews whose dominant positive (negative) topic is topic i divided by the total number of reviews in the inspection period. However, this aggregation method loses the information of the non-dominant subtopics that also contribute to the individual review text.

Machine Learning Model

We built a baseline machine learning model in the first place before including text-based features to filter the power of non-text-based features. Our suggested algorithm is expected to be better than the baseline. Put in the plain words, we want to know how well can predict the inspection violation without Yelp reviews, and how NLP model of Yelp review can facilitate the prediction power of machine learning model.

Non-text-based Features

The non-text-based features in the current baseline machine learning model includes average historical inspection penalty score, last inspection penalty scores, cuisines types, district (based on zip code), review count of the restaurant and average review rating.

Data Labeling

Determine the right label (hygiene and non-hygiene) is vital and disputable due to several reasons. First and most importantly, the inspection penalty score is asymmetrically clustered at low penalty level [Figure 10], which brings the problem of imbalanced data. Setting the threshold at high penalty score leads to much less sample labeled as "non-hygiene" compared to "hygiene" ones. There are two general ways to deal with imbalanced data: 1) resample the data 2) keep the data but change the evaluation metric. We can't choose the formal due the natural limitation in the availability of datasets, thus we use 5-fold cross validation with appropriate performance measurement to evaluate the machine learning model (See details in Model Evaluation).

Secondly, we found it ambiguous to cut the threshold of hygiene and non-hygiene by human intuition. Therefore, we let the machine decides for us: loop over different threshold cutoff of [10,20,30,40,50] of inspection penalty and choose the threshold that gives the best performance.

However, we are still looking for better way to determine the label. The potential drawback of using simple cutoff is that penalty scores is cumulative. A relative high penalty score may not necessarily indicate indicate severe health issues. For example, a restaurant may violate "proper labeling" or "broken windows" for several times and get a total penalty score of 40. However, these violations seem minorly correlated to health issues and thus are very unlikely to be mentioned by the reviewers. Therefore, we plan to merge a more detailed inspection record to focus on restaurant with severe violation as they are exactly the set of inspector and customer need to pay attention to

Model

We have built a machine learning pipeline that allows us to feed in different parameter tuning grids and multiple machine learning models. At the current stage, due to the time and computing power limitation, we only tuned our baseline model with Naive Bayes and Logistics Regression (with L1 and L2 penalty) with 5-fold cross validation. Naive Bayes is one of out pilot models because it will converge quicker than discriminative model and doesn't need parameter tuning. Although we haven't brought text feature in the baseline model, Naive Bayes is assumed to perform better when working with text classification.

Model Evaluation

In the proposal, we proposed to use Area Under the Receiver Operating Characteristic curve (AUC-ROC) as our key validation metric. However, we are taking out words back because ROC curve mainly represents the specificity of the model at different cut-off levels, which is not the best interests of inspectors. Standing in the shoes of inspectors, the question that the evaluation metric should answer is that of the all restaurants labeled "Non-hygiene", how many are actually "Non-hygiene"? Therefore, we have strong intention to favor precision score:

$$\text{Precision} = \frac{= \text{ \# of Correctly Predicted Non} - \text{Hygiene}}{\text{\# of Total Predicted Non} - \text{Hygiene}}$$

In summary, the evaluation metric has Mean Squared Errors, AUC-ROC, Precision at 1%, 2%, 5%, 10% and 20%, as well as the average precision for 5-fold cross validation.

Results

Although the machine learning model has very small grid size and also is not well tuned. It still generates some rough results.

1. As we increase the threshold, the precision goes down in most cases. [Figure 11]

2. Logistics outperforms Naive Bayes. Probably because we use Gaussian Naive Bayes while have a lot of binary features [Table 3]

3. Adding the text-based features (Topic Modelling) improve the precision rate of Naive Bayes, but has minor influence on Logistic Regression

IV. After-Midterm Plan

We plan to achieve the following goals in the rest of the quarter:

Improve review text-based features.

- Refine LDA model training by setting multiple passes and observing document converging speed.
- Experiment different ways to aggregate features generated from several individual review texts to one inspection period.
- Add more structural features such as length of the review, number of sentences, average sentence length,etc.
- Add percentages of POS tags in each review (low priority)

Increase non-text-based features

- Improve inspection history-based features. Right now, we rely heavily on inspection penalty score. Though it is used in multiple papers, we think the score adds unnecessary complexity and ambiguity to the machine learning model training. We plan to experiment using other inspection outcome scores that can be obtained from Seattle Open Data Portal.
- Add restaurant-based features: 1) Location (zip code, or smaller clusters based on address); (2) Cuisine type.

Fine tune machine learning model.

- Conduct recursive feature elimination or tree-based feature selection.
- Experiment with more model + features group combination to find the best performing one.
- Test our model on Las Vegas data obtained from Yelp Academic Dataset.
- If time permit, consider dealing with review bias, and issues related to inspection timing.
- If time permit, implement the inspector simulator proposed in Joaristi et al. paper.

V. Supplement File

Member	Task	Time spent	Lines of code
Ran Bi	Find appropriate datasets, data exploration, LDA modeling and feature generation	10 ~ 15 hours per week.	375 lines.
Minjia Zhu	Data cleaning and exploration, Feature Generation, Machine Learning Pipeline, Documentation	10 ~ 15 hours per week.	Haven't count it. Code is in Jupyter Notebook

VI. Appendix

Figure 1. Inspection heatmap 2006-2012

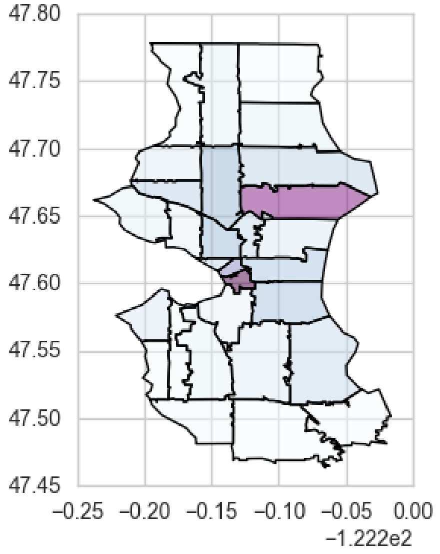


Figure 2. Annual total inspection instances

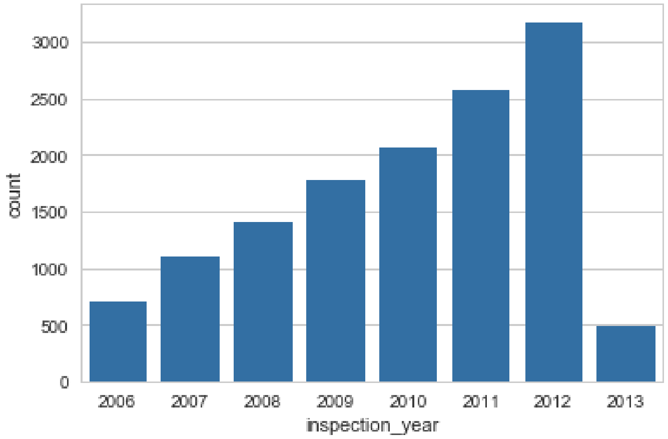


Figure 3. Monthly total inspection instances

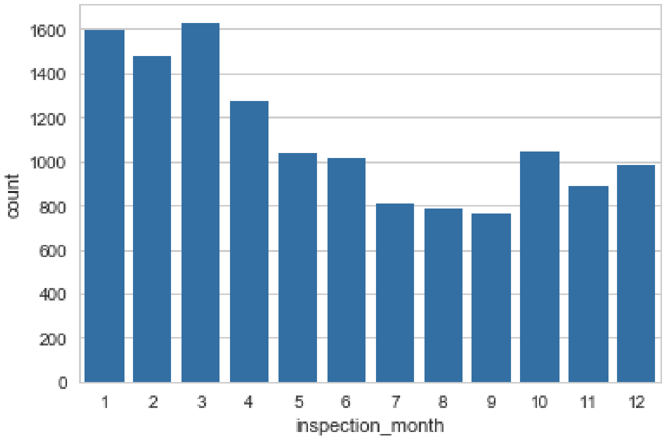


Figure 4. Spearman's coefficients of review average ratings and inspection penalty scores

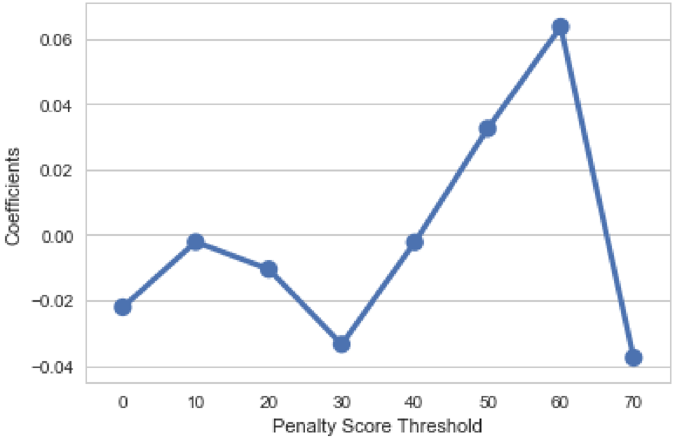


Figure 5. Spearman's coefficients of review count and inspection penalty scores

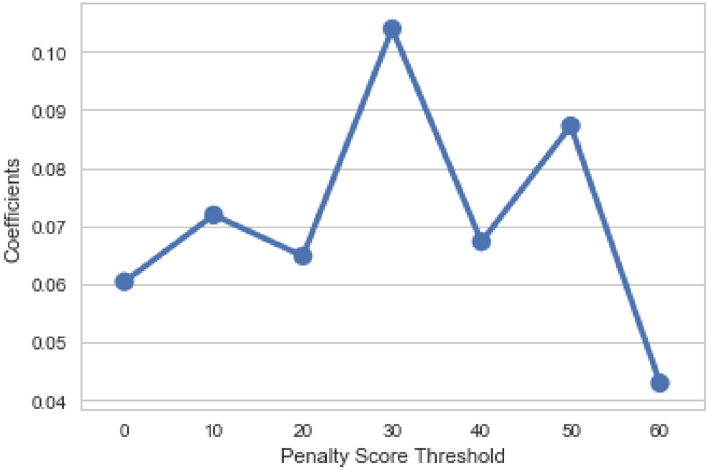


Figure 6. LDA model trained on all reviews together

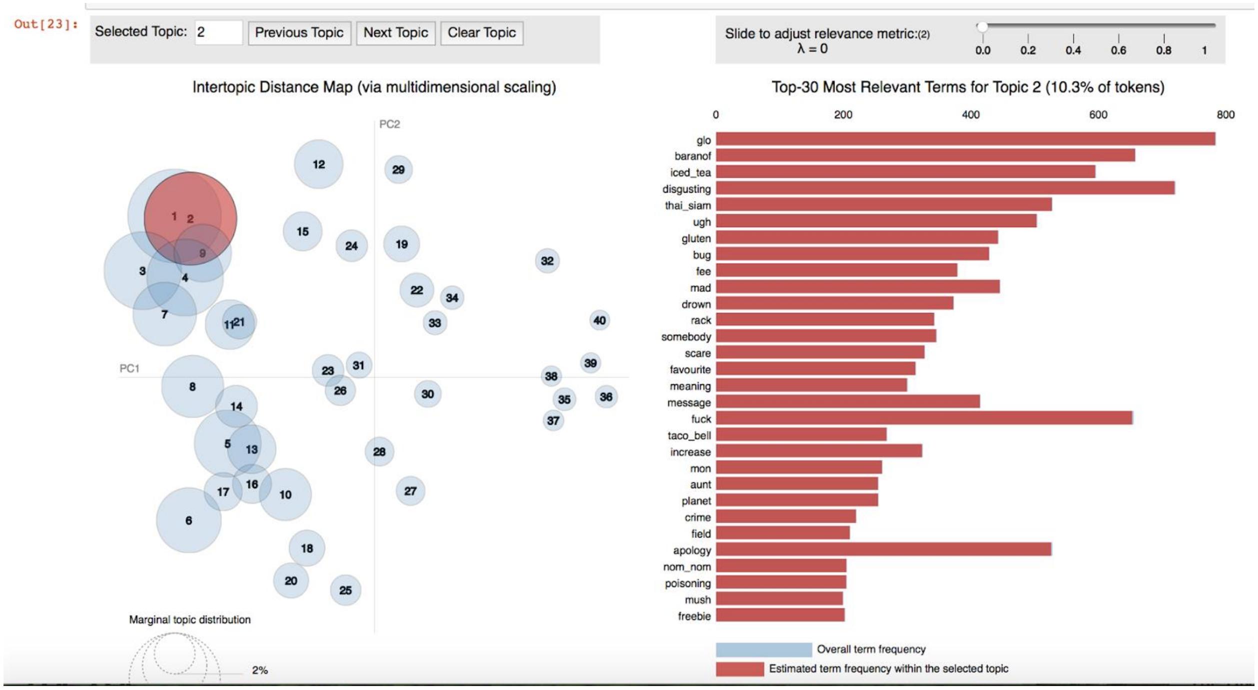


Figure 7. Number of topics vs Coherence score for LDA model trained on 1-star reviews

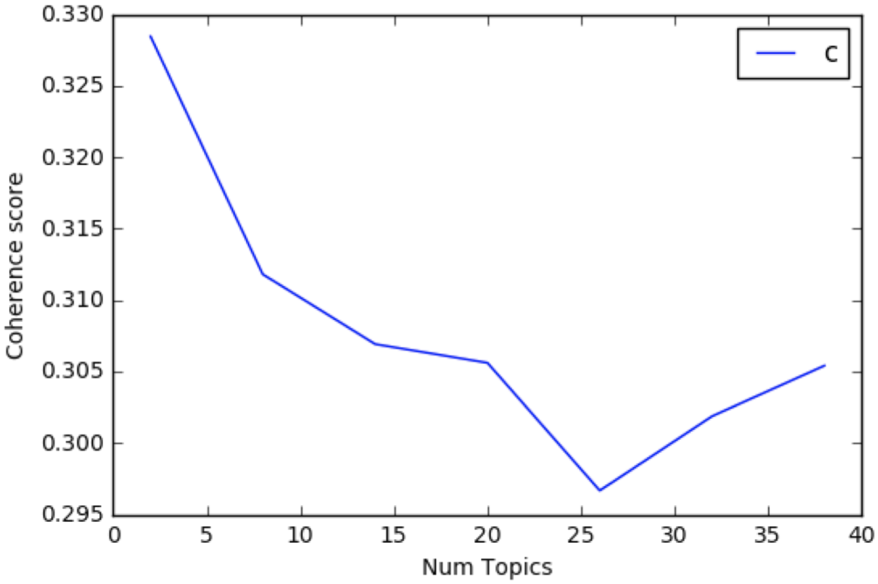


Figure 8. Number of topics vs Coherence score for LDA model trained on 5-star reviews

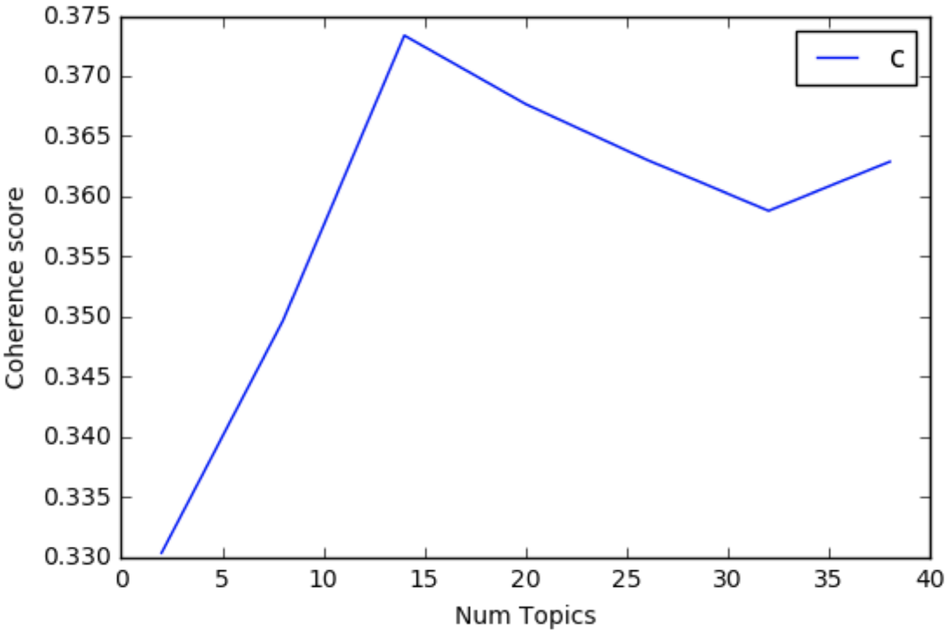
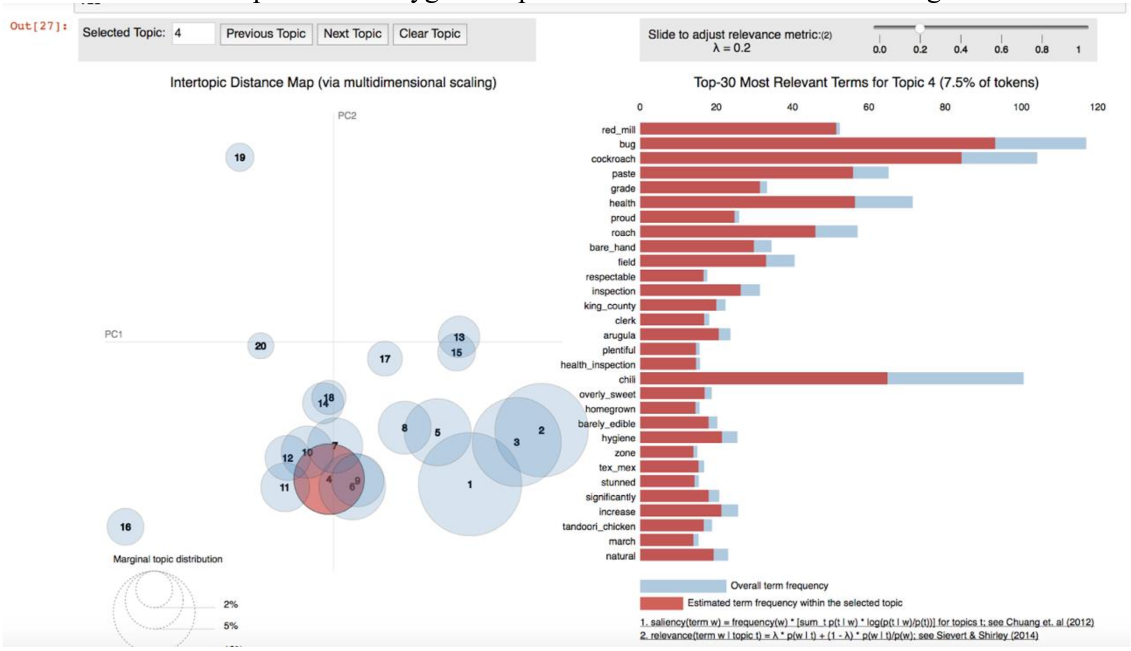


Figure 9. distribution of top words for hygiene topic from LDA model trained on negative reviews


Out[27]:

Selected Topic: 4

Slide to adjust relevance metric:(2)
λ = 0.2

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)

2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 10. Distribution of inspection Penalty Score

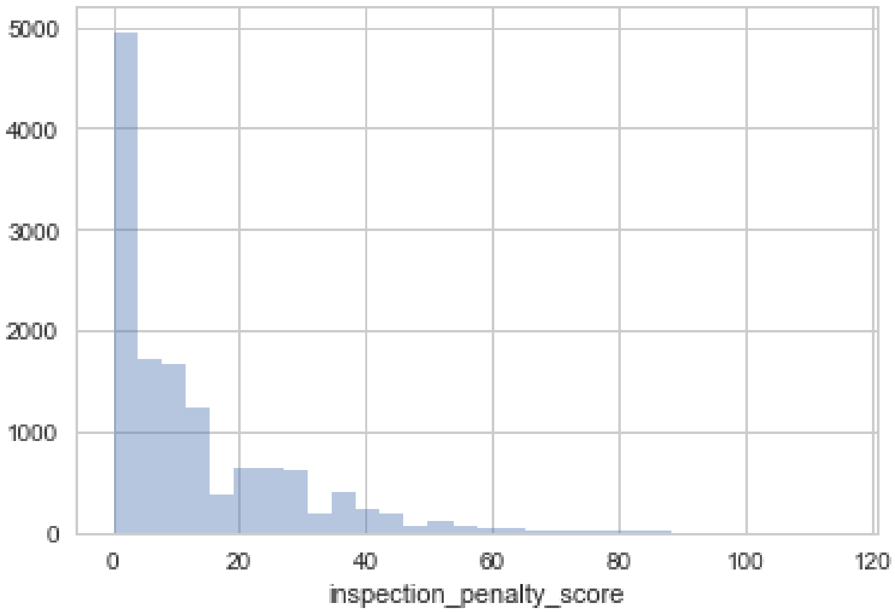


Figure 11. Trend of Penalty Score Thresholds & Precision

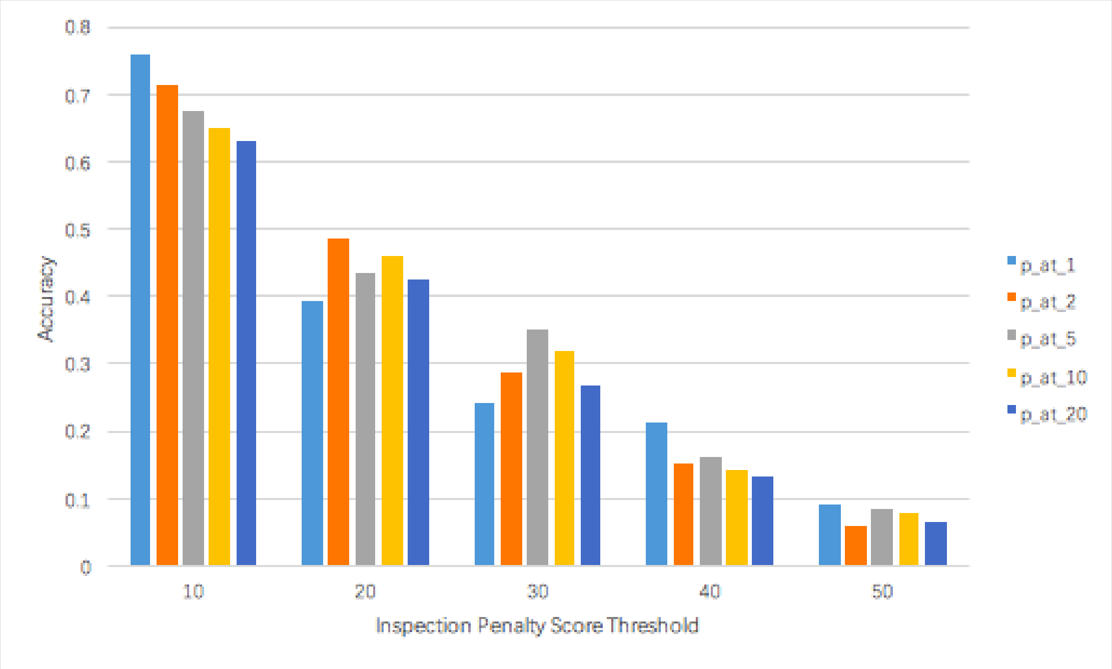


Table 1. Selected topics and top words (λ = 0.2) from LDA model trained on positive reviews

Topic	**Words**
Service	Seat, wait, minute, arrive, waiting, busy, line, seating, pricing…
General Praises	Great, friendly, excellent, awesome, amazing, reasonable, delicious…
Deli Food	Sandwich, burger, deli, turkey, lettuce, vessel, meat
Pub	Beer, pub, hipster, bagel, bar, pretzel, beer_selection, bloody_mary…
Asian Food	Noodle, Chinese, dumpling, raman, rice, katsu, bubble_tea, hot_pot…
Pizza	Pizza, crust, topping, thin, mozzarella, Hawaiian, calzone, slice, pesto
Coffee Shop	Coffee_shop, sweet_tooth, study, espresso, peanut_butter, fried, foam, barista, hospitable…

Table 2. Selected topics and top words (λ = 0.2) from LDA model trained on negative reviews

Hygiene	Bug, cockroach, roach, health, bare_hand, inspection, barely_eadible,…
General Complaint	Mislead, unaware, paper_thin, endless, depressing, grossly, scrub,…
Mexican Food	Mexican, taco, rice, burrito, taco_bell, roti, corn_tortilla,…
Pizza	Pizza, crust, gelato, topping, gluten_free, English_muffin, pie, oven,…

Table 3. Evaluation Metric - Baseline LR & NB Comparison (Threshold = 10)

model	**parameters**	**MSE**	**auc-roc**	**p_at_1**	**p_at_2**	**p_at_5**	**p_at_10**	**p_at_20**	**avg_p_cv**
Logistic Regression	{'C': 0.1, 'penalty': 'l1'}	0.39	65.2%	63.6%	65.2%	68.1%	68.1%	64.5%	60.1%
	{'C': 1, 'penalty': 'l1'}	0.39	64.9%	72.7%	72.7%	66.3%	65.7%	63.0%	59.5%
	{'C': 10, 'penalty': 'l2'}	0.39	64.8%	72.7%	71.2%	68.1%	64.8%	62.9%	59.2%
	{'C': 1, 'penalty': 'l2'}	0.39	64.8%	72.7%	71.2%	68.1%	64.8%	63.2%	59.2%
	{'C': 0.1, 'penalty': 'l2'}	0.39	65.0%	75.8%	69.7%	66.9%	65.7%	64.4%	59.2%
	{'C': 0.001, 'penalty': 'l2'}	0.38	64.6%	54.5%	63.6%	65.7%	62.7%	63.2%	59.2%
	{'C': 10, 'penalty': 'l1'}	0.39	64.8%	75.8%	71.2%	67.5%	65.1%	62.9%	59.2%
	{'C': 0.001, 'penalty': 'l1'}	0.38	64.4%	57.6%	60.6%	62.7%	62.3%	64.1%	59.1%
	{'C': 1e-05, 'penalty': 'l2'}	0.38	64.7%	57.6%	57.6%	62.0%	61.1%	63.3%	58.9%
NB	{}	0.46	60.3%	6.1%	39.4%	56.6%	56.3%	59.4%	54.7%

Table 4. Evaluation Metric - Adding Topic Modelling Feature to Baseline

model		LR									NB
auc-roc	Baseline	65.20%	64.90%	64.80%	64.80%	65.00%	64.60%	64.80%	64.40%	64.70%	60.30%
	Baseline+ Topic	65.25%	64.82%	64.71%	64.70%	64.98%	64.82%	64.50%	64.65%	64.75%	61.57%
p_at_1	Baseline	63.60%	72.70%	72.70%	72.70%	75.80%	54.50%	75.80%	57.60%	57.60%	6.10%
	Baseline+ Topic	63.64%	72.73%	72.73%	75.76%	75.76%	54.55%	75.76%	57.58%	54.55%	75.76%
p_at_2	Baseline	65.20%	72.70%	71.20%	71.20%	69.70%	63.60%	71.20%	60.60%	57.60%	39.40%
	Baseline+ Topic	62.12%	71.21%	71.21%	72.73%	69.70%	60.61%	69.70%	63.64%	57.58%	74.24%
p_at_5	Baseline	68.10%	66.30%	68.10%	68.10%	66.90%	65.70%	67.50%	62.70%	62.00%	56.60%
	Baseline+ Topic	69.28%	69.28%	68.67%	68.07%	67.47%	66.87%	68.07%	59.64%	61.45%	65.06%
p_at_10	Baseline	68.10%	65.70%	64.80%	64.80%	65.70%	62.70%	65.10%	62.30%	61.10%	56.30%
	Baseline+ Topic	68.37%	66.87%	66.87%	67.47%	68.07%	62.95%	65.66%	62.35%	61.14%	63.55%
p_at_20	Baseline	64.50%	63.00%	62.90%	63.20%	64.40%	63.20%	62.90%	64.10%	63.30%	59.40%
	Baseline+ Topic	63.91%	63.46%	63.61%	64.36%	64.21%	62.56%	63.91%	63.01%	62.86%	62.71%
cv_avg_precision	Baseline	60.10%	59.50%	59.20%	59.20%	59.20%	59.20%	59.20%	59.10%	58.90%	54.70%
	Baseline+ Topic	60.25%	59.64%	59.36%	59.36%	59.32%	59.39%	59.19%	59.23%	59.01%	57.01%