




SENTIMENT ANALYSIS USING YELP REVIEWS

George (Xiaochi) Li, Li Du
Natural Language Processing Final Project
Dec 7, 2018



The Dataset



5,996,996 reviews



188,593 businesses



280,992 pictures



10 metropolitan areas

Data available in six JSON files: business, review, user, checkin, tip, photos

review.json

```
{
  // string, 22 character unique review id
  "review_id": "zdSx_SD6obEhz9VrW9uAWA",

  // string, 22 character unique user id, maps to the user in user.json
  "user_id": "Ha3iJu77Cx1rFm-vQRs_8g",

  // string, 22 character business id, maps to business in business.json
  "business_id": "tnhfDv5I18EaGSXZGiuQGg",

  // integer, star rating
  "stars": 4,

  // string, date formatted YYYY-MM-DD
  "date": "2016-03-09",

  // string, the review itself
  "text": "Great place to hang out after work: the prices are decent, and",

  // integer, number of useful votes received
  "useful": 0,

  // integer, number of funny votes received
  "funny": 0,

  // integer, number of cool votes received
  "cool": 0
}
```

Number of stars corresponding to the review

Review text stored as string

Business.json

```
// string, 22 character unique string business id
"business_id": "tnhfDv5I18EaGSXZGiuQGg",

// an array of strings of business categories
"categories": [
  "Mexican",
  "Burgers",
  "Gastropubs"
],
```

Motivation

- The Yelp dataset provides abundant information that can be helpful to robust analysis and modeling
- Large amount of textual data to evaluate multiple natural language processing techniques
- Star levels associated with each review allows supervised learning and provides ways to evaluate the natural language processing techniques

Objectives

- Implement and evaluate text vectorization approaches
- Develop supervised learning model to predict the star levels based on the reviews
- Evaluate the model and vectorizer

Preprocessing & data preparation

- The size of the original review dataset is **3654k**, which is too large for analyze in a reasonable time.

We sorted the “categories” field of business json and selected the most common categories, and we created two variables for restaurant categories:

1. (idA)Cuisine:American(Traditional)(1), Italian(2), Mexican(3), Chinese(4), American(New)(5), Japanese(6), Other(0)
 2. (idB)Type: Nightlife(1), Fast Food(2), Breakfast &Brunch(3), Coffee& Tea(4), Cafes(5), Others(0)
- We filtered out the review set that (idA!=0 or idB!=0), the size of filtered dataset is **921k**, which is still too large. (It takes 4h47min to tokenize and remove stop word on AWS)
 - We selected the review that has at least 2 “useful”
and random sample 76% of them.

921172/921172 [4:47:34<00:00, 53.39it/s]

Useful 8 Funny 5 Cool 4

Finally, we reduced the size of the dataset to **144k** (It takes 24 min to tokenize and remove stop words)

Original texts

"Love their chicken and waffle! The service was great. \nTook one star out because I just can't believe with this size of restaurant they only 2 restroom one for woman and one for men so if the restaurant is busy then the wait for the restroom is going to be longggg."

Processed texts

'love chicken waffle service great took one star
ca believe size restaurant restroom one woman
one men restaurant busy wait restroom going
longggg'

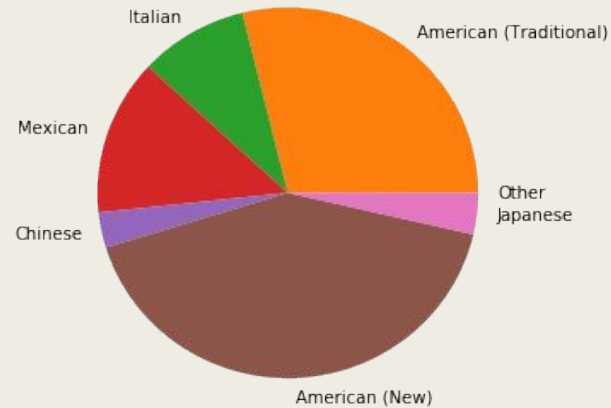
,

Exploring the dataset...

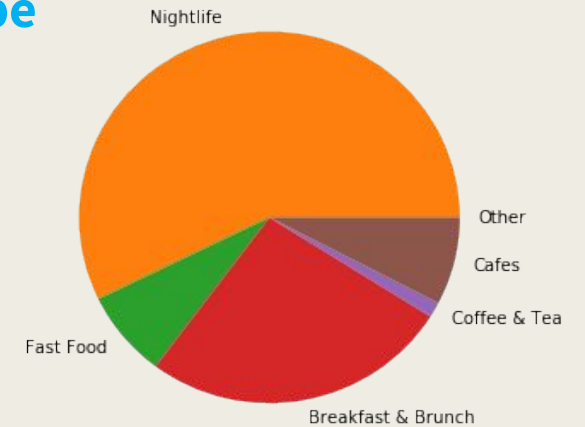
Stars



Cuisine



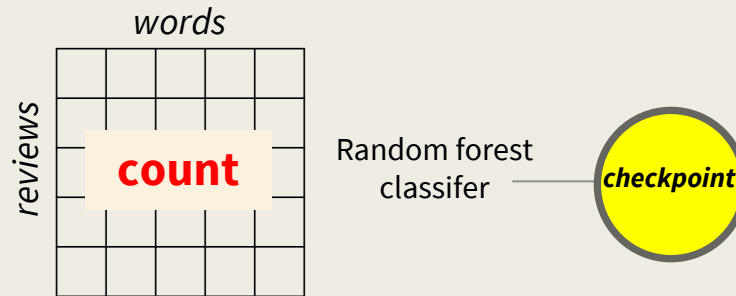
Type



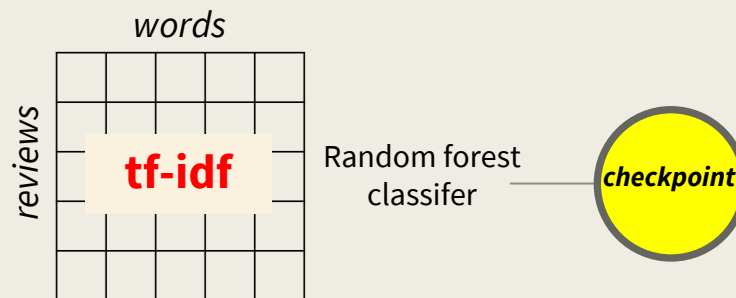
- The five star categories are relatively balanced although 4 and 5 reviews consist of more than 50% of the entire dataset
- American food (traditional and new) and nightlife restaurants are the majority

Bag of words models

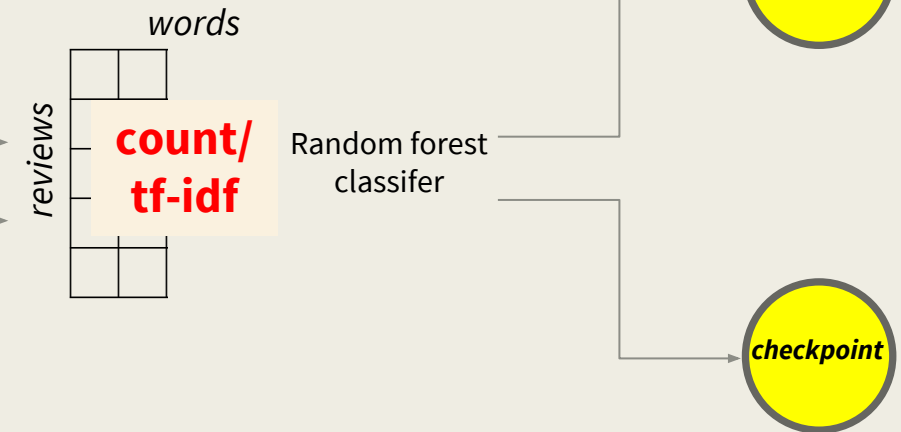
Count vectorization



Tf-idf vectorization



Truncated singular value decomposition (aka. Latent semantic analysis)

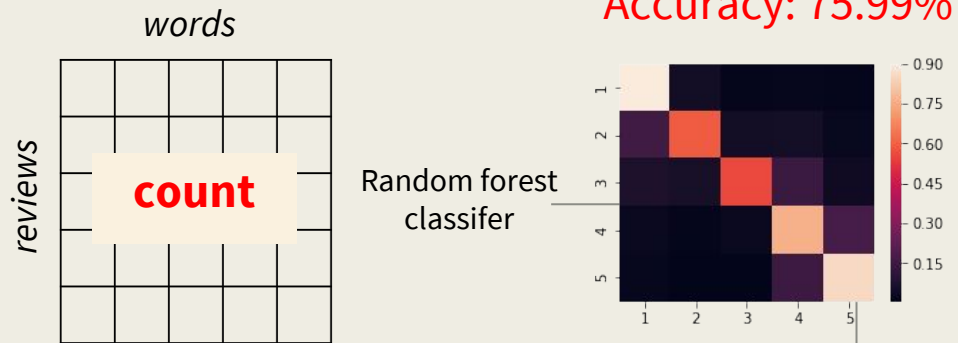


- Minimum document frequency = 1%, maximum document frequency = 5%
- Only keep the top 5000 features ordered by frequency
- Consider both single words and bi-grams

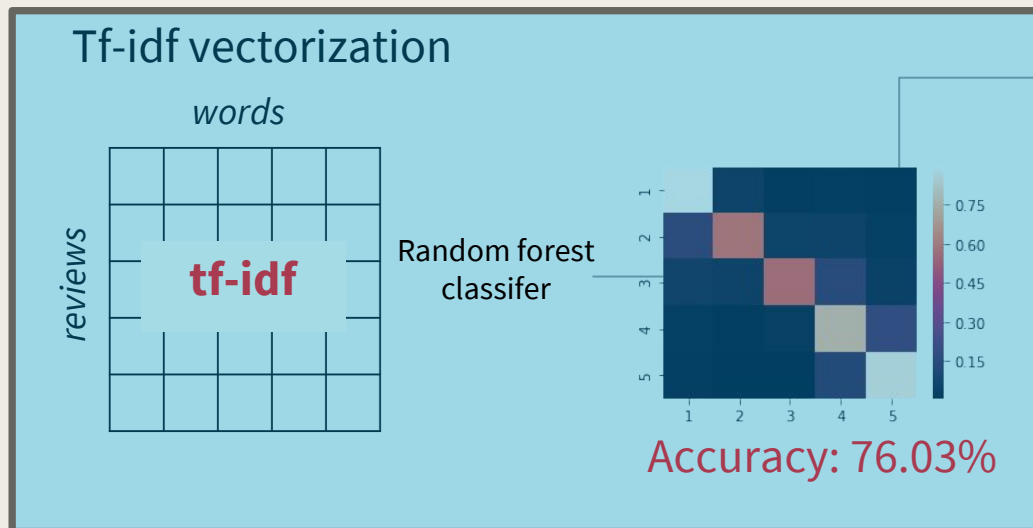
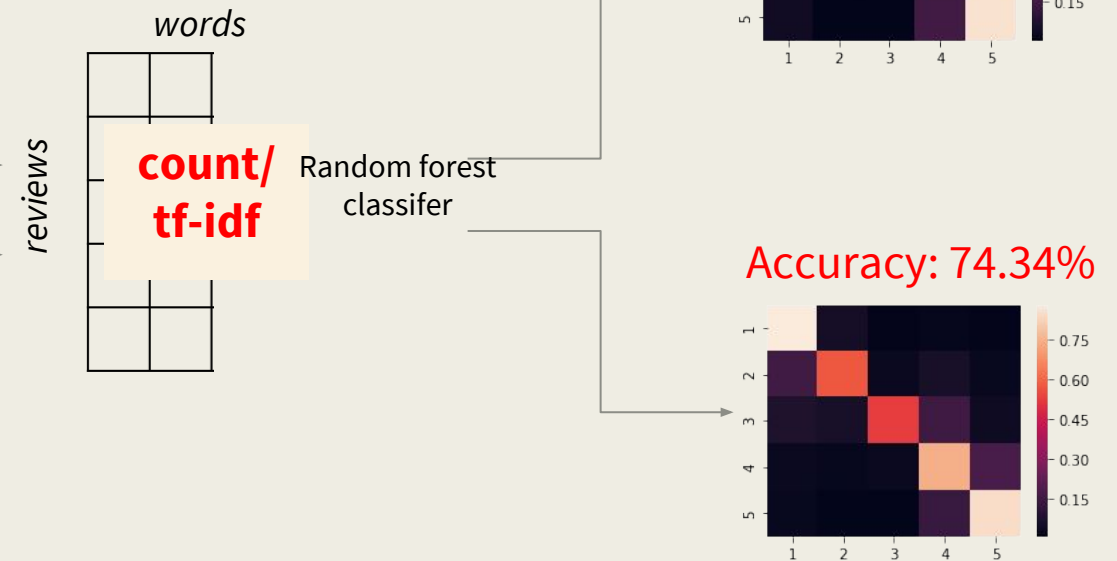
- Output dimension was set to 100

Bag of words models

Count vectorization



Truncated singular value decomposition (aka. Latent semantic analysis)



Tf-idf vectorization has a tiny winning margin

TF-IDF Vectorization (or any evaluated cases)



- Reviews with strong feelings are easier to predict, especially the strong “likes” and strong “dislikes”
- 2 and 3 star are hardest to predict, likely due to the lack of words with strong emotions.

What words are determining the prediction performance?

Words **MOST** important to
the sentiment prediction



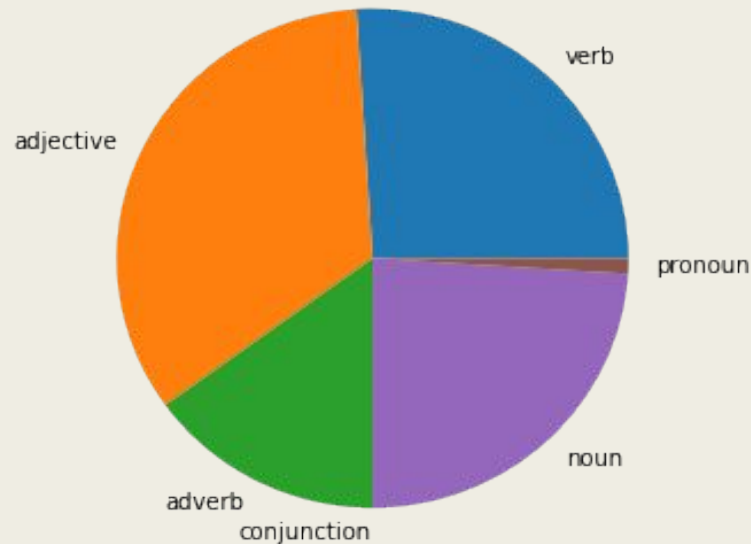
Words **LEAST** important to the sentiment prediction



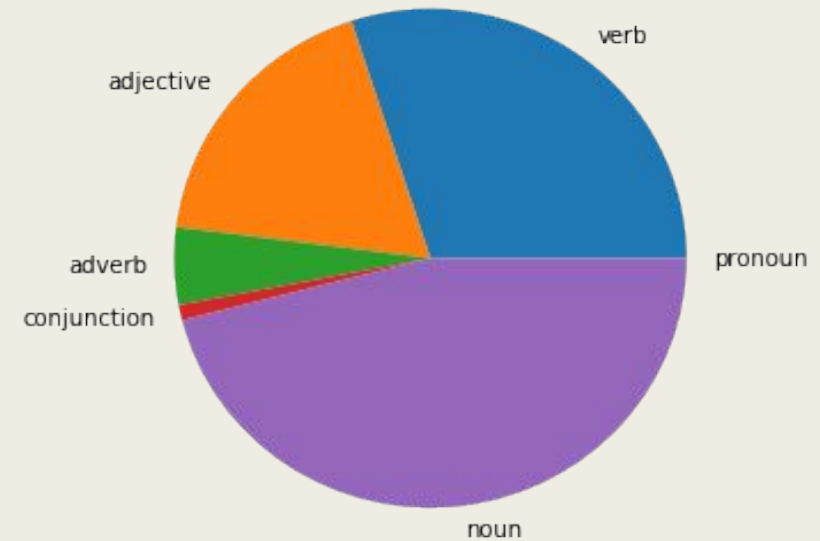
Size of the words represents relative importance: most import and least important are shown with the largest sizes

What words are determining the prediction performance?

Words **MOST** important to the sentiment prediction



Words **LEAST** important to the sentiment prediction



Note: Part of speech tagging was done manually and may subject to minor errors

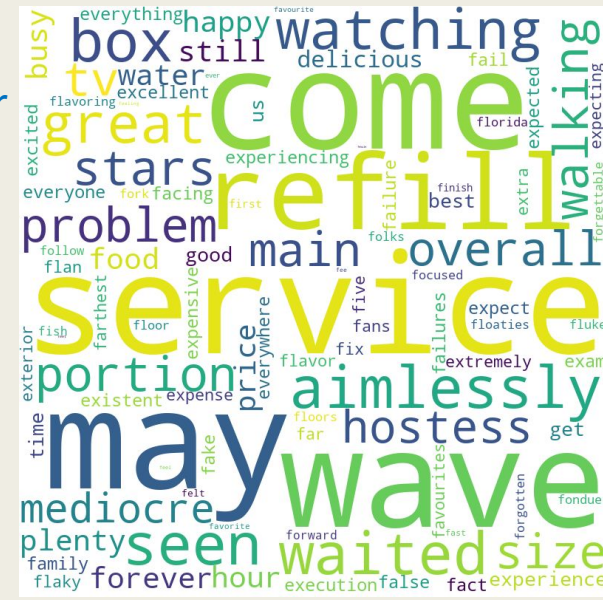
Where the model fails (tragically)...

- These reviews tend to shorter
 - *Reviews with sentiment correctly predicted: 545 characters/review*
 - *Reviews with 1 star predicted as 5 star: 411 characters/review*
 - *Reviews with 5 star predicted as 1 star: 464 characters/review*
- These reviews tend to be a bit ambiguous and may lack strong emotional words

True: 5 star
Predicted :1 star

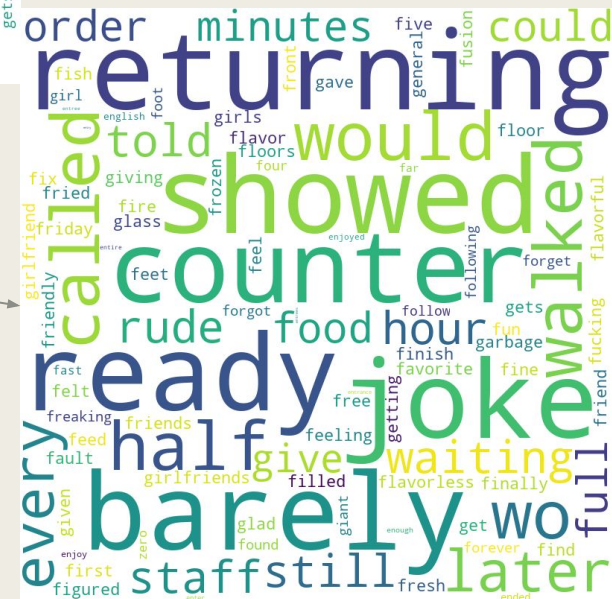
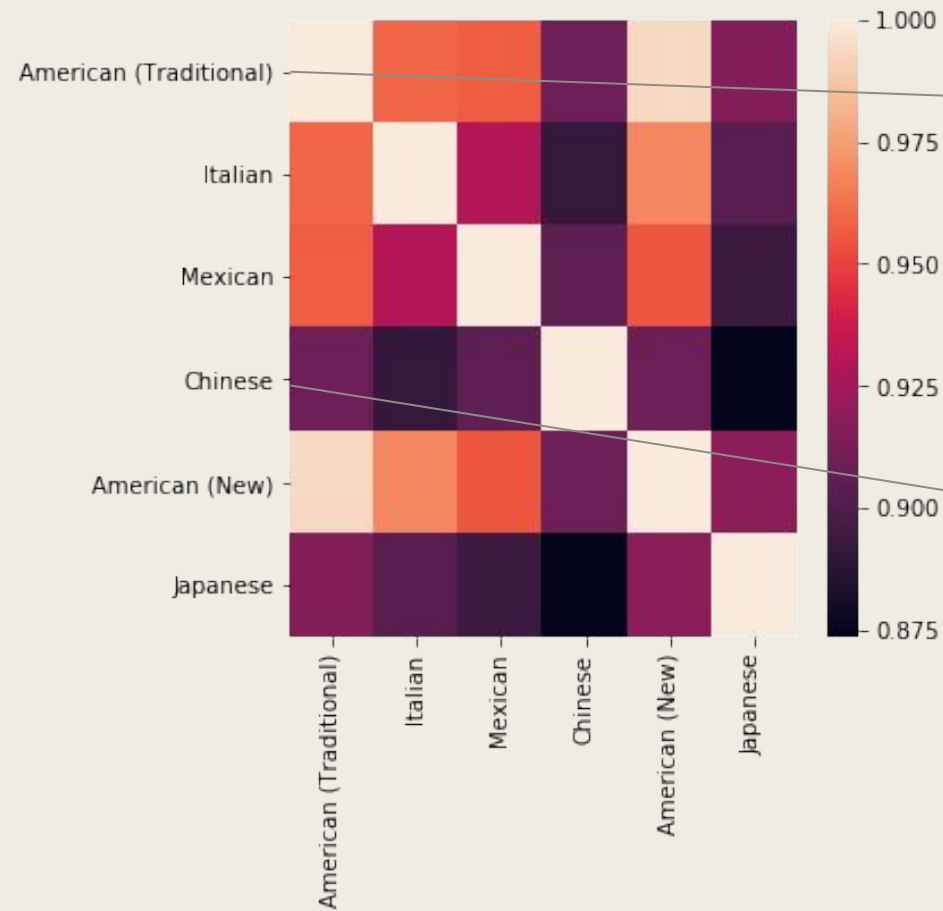


True: 1 star
Predicted :5 star



How different are the categories?

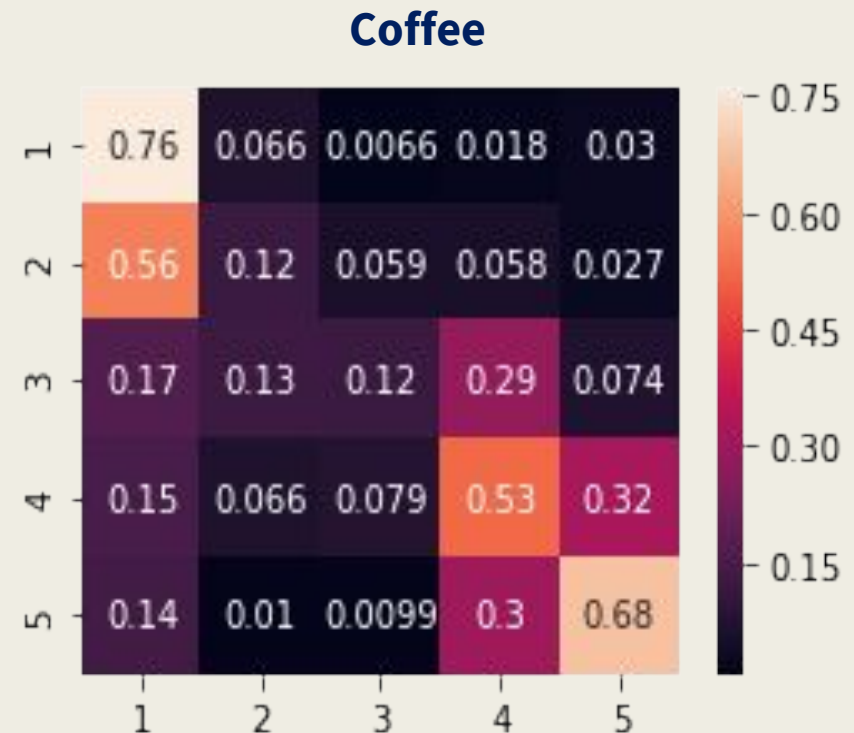
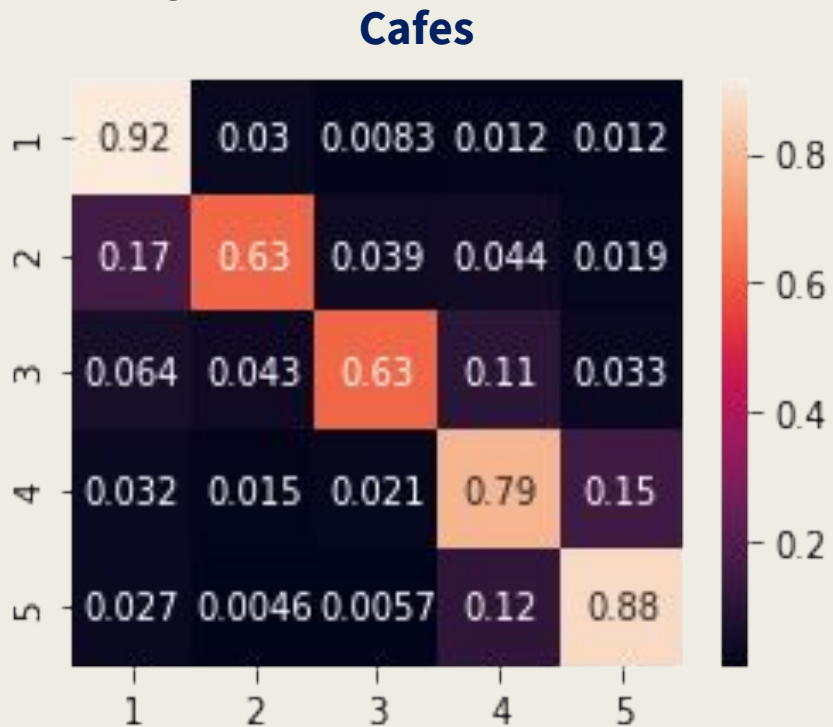
Cosine similarity of the tf-idf vectors of 1 star reviews among different cuisines



Are model trained on one category predictive to others?

It depends ...

A model trained using reviews for “Nightlife” restaurants performs well with “Cafes” but not any other categories



Conclusions

- Count vectorization, tfidf vectorization and the application of LSA yielded similar results.
- Strong sentiments are easier to predict whereas the model performs less ideally for reviews with moderate sentiments
- Adjectives and adverbs with strong sentiments are most important in the modeling
- The performance of models trained on one category likely depends on the similarity between the training set and the target set
- Need to optimize the algorithms and process so that the modeling can be implemented in reasonable time