



DC-DS-14 Final Project

Amy Ghatge

Project Outline

➤ What is your project about?

- Yelp Dataset Challenge - https://www.yelp.com/dataset_challenge
- Yelp ratings allow a user to rate a restaurant between a 1 – 4
 - Predict user restaurant ratings based on reviews
 - Based on correct model – what are the things most important to restaurant go-ers – what aspects of dining and food do high ratings and low ratings focus on?

➤ What is its history?

- Dataset challenge in it's 8th round
- Many research papers already published on this dataset - https://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_ImprovingRestaurants.pdf

Your Next Review Awaits



Fudge Street Cafe

[Read our review guidelines](#)



Yay! I'm a fan.

Saved

Cute place in a very small town. We stopped in for dessert - ice cream, chocolate cake and coffee. The service was really friendly. The space and decor are interesting as the restaurant is a converted grocery warehouse. For example the tables are made from old refrigerator doors. Stop by if you happen to be in Covington!

★ Checked out this place recently? What was your experience like?

Share your review on ☐ Facebook

[Cancel](#)

[Post Review](#)

Project Summary

- Review topics analysis – what's important to people when they rate restaurants?
- Based on a review – suggest a star rating in an effort to standardize ratings
- Here is the data:

	business_id	date	review_id	stars	text	type	user_id	votes
0	5UmKMjUEUNdYWqANhGckJw	2012-08-01	Ya85v4eqdd6k9Od8HbQjyA	4	Mr Hoagie is an institution. Walking in, it do...	review	PUFPaY9KxDacGqfsorJp3Q	{u'funny': 0, u'useful': 0, u'cool': 0}
1	5UmKMjUEUNdYWqANhGckJw	2014-02-13	KPvLNJ21_4wbYNctrOwWdQ	5	Excellent food. Superb customer service. I mis...	review	lu6AxdBYGR4A0wspR9BYHA	{u'funny': 0, u'useful': 0, u'cool': 0}

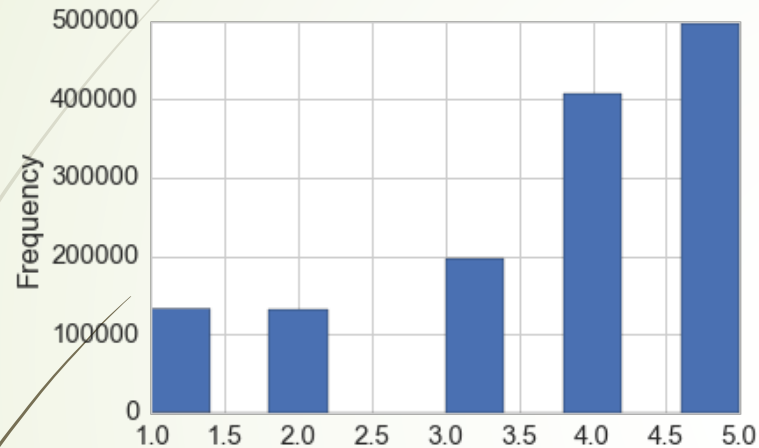
attributes	business_id	categories	city	full_address	hours	latitude	longitude	name
{u'Take-out': True, u'Drive-Thru': False, u'Out...	5UmKMjUEUNdYWqANhGckJw	[Fast Food, Restaurants]	Dravosburg	4734 Lebanon Church Rd\nDravosburg, PA 15034	{u'Tuesday': {u'close': u'21:00', u'open': u'1...	40.354327	-79.900706	Mr Hoagie

neighborhoods	open	review_count	stars	state	type	Restaurant
	True	4	4.5	PA	business	1.0

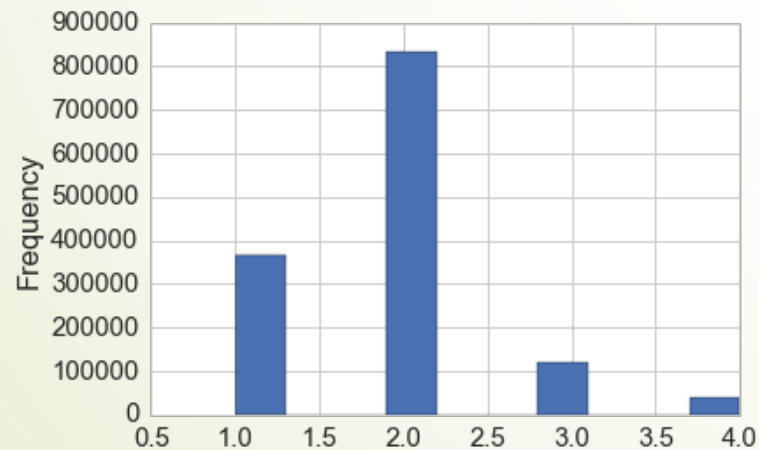
- Lots of features but for a first project I decided to use **price** and **topic** analysis of the reviews
 - Other features that could be used
 - Ambiance
 - Offer take out
 - Parking
 - Delivery
 - Tags (ie. Mexican, Chinese, Brunch, etc)

Exploratory Insight

Stars Distribution by Users



Price Distribution by Restaurant



Correlation

	stars_review	review_count	open	stars_business	price
stars_review	1.000000	0.073063	0.048143	0.398845	0.022634
review_count	0.073063	1.000000	0.127398	0.167174	0.213545
open	0.048143	0.127398	1.000000	0.110348	0.002237
stars_business	0.398845	0.167174	0.110348	1.000000	0.047419
price	0.022634	0.213545	0.002237	0.047419	1.000000

Describe

	stars_review	review_count	stars_business	price
count	1.363242e+06	1.363242e+06	1.363242e+06	1.356627e+06
mean	3.736939e+00	4.108197e+02	3.729141e+00	1.874950e+00
std	1.302318e+00	7.273045e+02	5.570203e-01	6.755791e-01

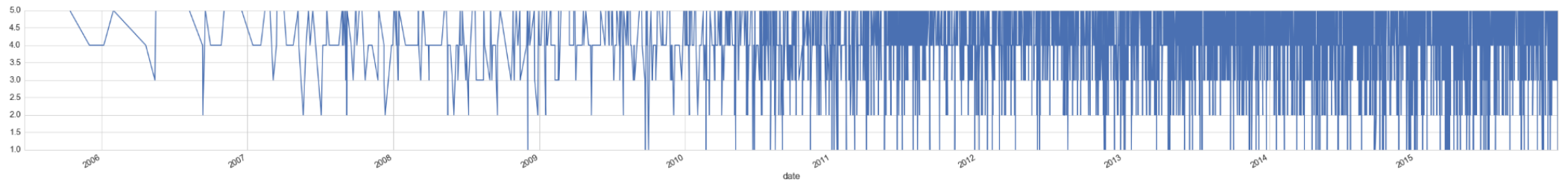
Exploratory Insight

Time Series

```
In [33]: df_res1 = df_working[df_working.business_id == "4bEjOyTaDG24SY5TxsaUNQ"]
```

```
In [34]: df_res1.stars_review.plot(figsize=(50, 5))
```

```
Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x1a38f90d0>
```



```
In [37]: df_res1.stars_review.resample('AS').mean().autocorr(lag=1)
```

```
Out[37]: 0.74345342509923995
```

Modeling

Optimal Number of Topics – measure perplexity

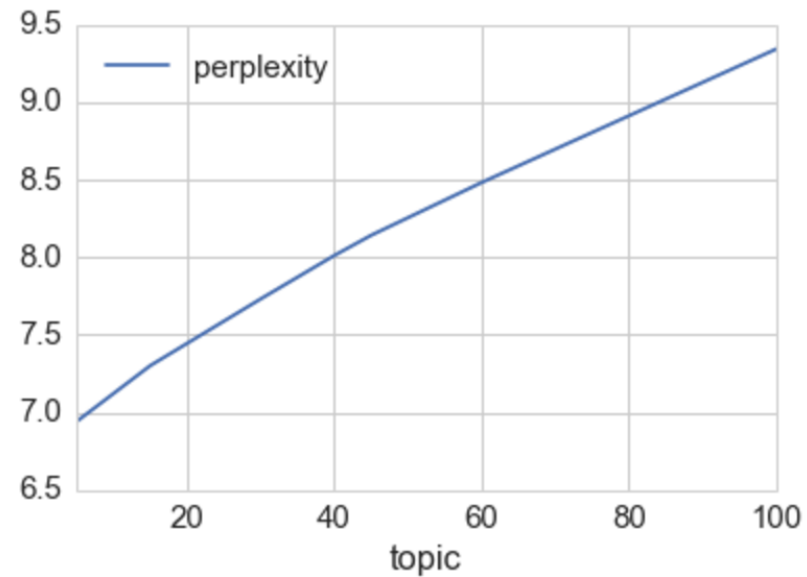
```
X_train, X_test, y_train, y_test = cross_validation.train_test_split(X, y, test_size=0.20)
```

```
vectorizer_sample = TfidfVectorizer(max_df=.8, min_df=.01, ngram_range=(1,2), stop_words='english', lowercase=True)  
v_sample = vectorizer_sample.fit_transform(X_train.text)
```

```
id2word_sample = dict(enumerate(vectorizer_sample.get_feature_names()))
```

```
corpus = Sparse2Corpus(v_sample, documents_columns=False)
```

```
lda_model_45 = LdaModel(corpus=corpus, id2word=id2word_sample, num_topics=45)
```



Modeling

- LDA with 45 topics –
 - max_df=.8
 - min_df=.01
 - ngram_range=(1,2)
 - stop_words='english'

- Topics – Name and Sentiment


```
Topic: 0
(0, u'0.021*looked + 0.018*decided + 0.013*looked like + 0.013*like + 0.013*decided try + 0.012*try + 0.009*good +
0.009*place + 0.009*just + 0.008*food + 0.008*stick + 0.008*got + 0.008*saw + 0.008*ordered + 0.007*didn')
()
Topic: 1
(1, u'0.024*horrible + 0.024*food + 0.023*terrible + 0.022*worst + 0.019*service + 0.018*awful + 0.017*rude + 0.017
*quality food + 0.016*quality + 0.016*bad + 0.014*poor + 0.014*money + 0.013*mediocre + 0.011*place + 0.011*spend')
()
Topic: 2
(2, u'0.020*line + 0.019*long + 0.017*wait + 0.015*waiting + 0.014*time + 0.012*20 + 0.011*food + 0.011*minutes +
0.011*long time + 0.011*20 minutes + 0.010*order + 0.009*40 + 0.009*ready + 0.008*people + 0.008*understand')
()
Topic: 3
(3, u'0.047*brunch + 0.019*fruit + 0.018*sunday + 0.014*desert + 0.014*orange + 0.013*breakfast + 0.009*good + 0.00
8*great + 0.007*place + 0.007*food + 0.007*menu + 0.006*fresh + 0.006*coffee + 0.006*toast + 0.006*like')
()
Topic: 4
```

Modeling

➤ Random Forest Classifier

- `n_estimators=200`
- `max_depth=30`

```
X_train, X_test, y_train, y_test = cross_validation.train_test_split(X1, y, test_size=0.4, random_state=42)
clf = ensemble.RandomForestClassifier(n_estimators=200, max_depth=30)
clf.fit(X_train, y_train)
```

Results

- Baseline: 0.66
 - Accuracy: 0.776
 - ROC AUC: 0.806
 - Avg Topic Score for Restaurants with 3 or below
 - Avg Topic Score for Restaurants with a 4 or above
- 

Conclusion Slides

- Was I able to predict a user's star rating from their text review?
- Was I able to determine what restaurant go-ers who rate highly care about vs. those who give low ratings?

Bad (rating ≤ 3)

	Score
11	0.076968
12	0.033886
14	0.033196
26	0.052744
31	0.103038

Good (rating > 3)

	Score
8	0.030686
43	0.032329

Topic Score > 0.03

Topic #	Sentiment	General Topic	Word Distribution
8	Good	Great Service	0.101*great + 0.059*great food + 0.049*great service
11	Bad	Bad Service and Food	0.019*terrible + 0.018*food + 0.018*horrible + 0.016*service
12	Neutral	Greetings - Service	0.013*greeted + 0.012*walked + 0.011*looked
14	Bad	Slow Service – Took too long to get food	0.022*line + 0.022*wait + 0.018*minutes + 0.017*20 + 0.015*food + 0.015*20 minutes + 0.014*waiting + 0.013*long + 0.013*time + 0.012*slow
26	Bad	Food Quality - bland	0.016*bland + 0.015*food + 0.013*okay + 0.012*good + 0.011*just + 0.011*place + 0.010*average + 0.010*disappointing + 0.010*pretty + 0.010*expectations + 0.009*mediocre
31	?	Minutes, Ordered, Waited	0.021*minutes + 0.013*order + 0.012*food + 0.012*came
43	Neutral	Breakfast!	0.047*breakfast + 0.036*coffee + 0.028*eggs + 0.023*pancakes

- Things I wanted to try but ran out of time – More features (price), Predict actual rating (1-5)...didn't work,



Next Steps Slides

- Lots of features to test out
 - Does the city make a difference?
 - Do tags help create a better model?
 - Does the trendiness of a restaurant help it's ratings?
- What would be the next two or three things you want to try? What impact might they have?
 - With more data over time you could really consult restaurants on what matters most to restaurant go-ers
 - Next Yelp challenge involves machine learning with images – lots of analysis there

Acknowledgements



- Thank you Alex and John for all your help!
- And Stack Overflow!