Springboard Capstone Report: Finding Fake Reviews of Yelp

Howell Yu

Background & Purpose

Yelp is a multinational technology corporation whose product includes Yelp.com and Yelp mobile app and it focuses on publishing consumers' reviews about local businesses. Today, Yelp has become a common source to help consumers obtain general information and reviews of local businesses. As a result, it is crucial for local businesses to develop great star ratings and positive reviews as the consumers rely heavily on the information given by Yelp. However, customers' heavy dependency on Yelp also gives rise to an industry that lives in the dark - fake reviewers, who focuses on giving fake star ratings and positive comments to local business and therefore give unfair competitive advantages to low-rating businesses. I personally have friends who worked as a fake reviewer and told me how reviewer worked and therefore I come up with this idea to filter out possible businesses that might have hired fake reviewers, which Yelp can take a deeper look into.

Assumptions of Fake Reviews

1. Since the purpose of fake reviews is to increase the both the star ratings and quality of comments, during the month where fake reviewers are hired, there will be a significant increase in star rating.
2. Since fake reviewers may comment based on certain templates and for the same fake reviewer, he or she might have given out fake reviewers for a number of different business, it is highly likely that businesses which hired fake reviewers will have highly similar reviews.
3. Only a very small portion of businesses hired fake reviewers.

Explanation and Main Analysis

Coding language: Python
Libraries used: *Pandas, numpy, sklearn, scipy, json, copy, matplotlib*

1. **Initial Filter**
   Filter possible fake reviews based on the change of rating, the number of comments, the length of comments and create a data frame to store all the suspicious business id and corresponding information of their reviews.

   According to assumption 1, if there is a significant change in the reviews such as a sudden increase of the number of reviews or a noticeable increase in star ratings, it is highly like that the corresponding businesses hired a fake reviewer. Even though it is likely that some business increases their ratings and gain more positive reviews by redecorating their place or

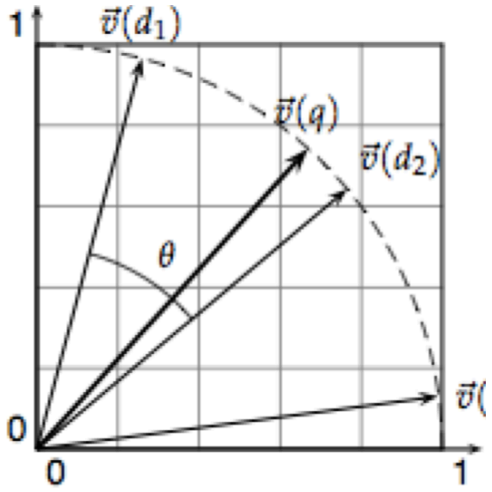
changing in products, we still regard them as suspicious. In this case, we filter business with more than 30 comments and whose monthly number of rating has increased more than 50% over the previous month and therefore obtain 70 suspicious business ids.

2. Analyzing the Comments (Bag-Of-Words Representation)

We can use Bag-Of-Words representation for the comments in the review and compute the cosine (correlation) between comments from different business id that was determined suspicious from the previous step.

For 2 comments d_1 and d_2, the cosine similarity of their vector representation $\overline{V}(d_1)$ and $\overline{V}(d_2)$ is:

$$\text{Cosine Similarity} = \frac{\overline{V}(d_1) \cdot \overline{V}(d_2)}{|\overline{V}(d_1)| \times |\overline{V}(d_2)|}$$



However, this method alone cannot provide us with a pleasant result since computing the cosine similarity has complexity $O(n^2)$. Thus, with 70 business ids, each of whom has on average more than 1000 comments, the total computation cost is really high.

To solve the problem of high computation cost, we could try clustering the comments.

3. Cluster the Comments

According to assumption 2, fake reviewers might follow a pattern or template when commenting for local business. Plus, it is unlikely that fake reviewers have been to many of the business where they post their fake reviews, which makes it harder for them to give detailed comments on the product. Instead, they may use descriptive fancy words in their comments. Accordingly, this gives room for the use of clustering method.

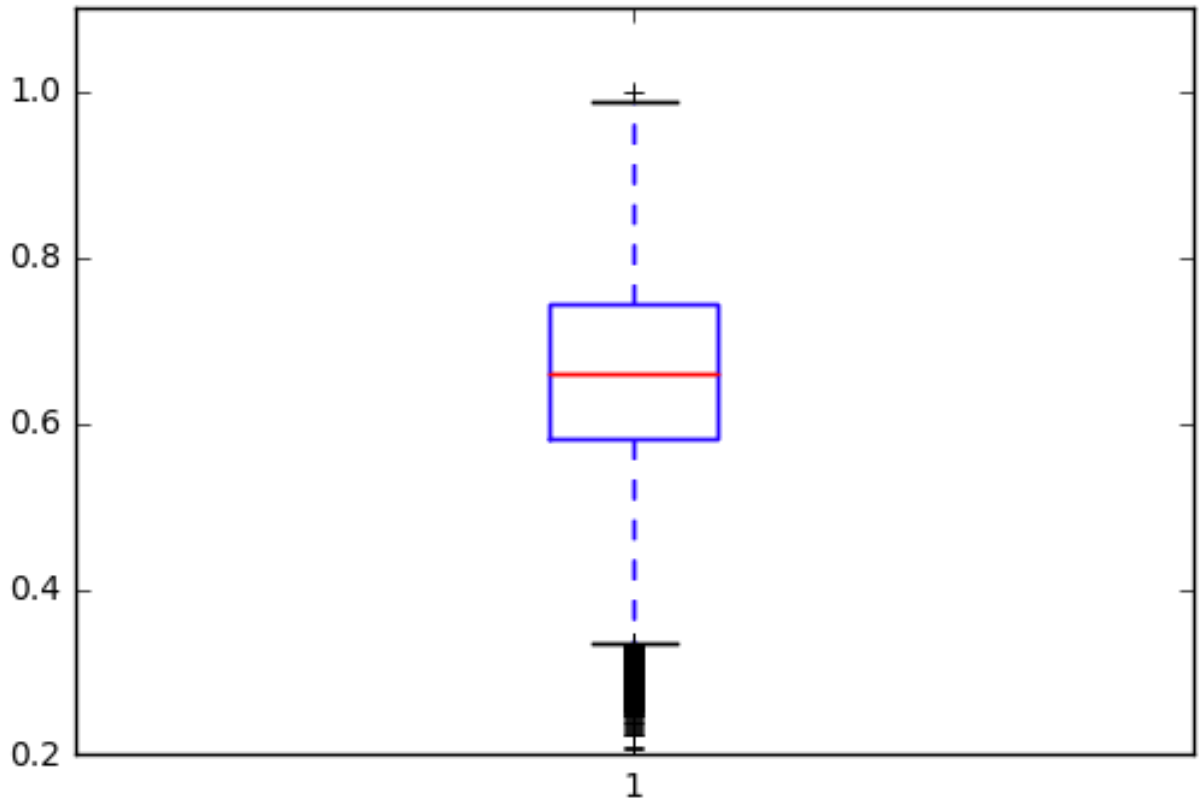
In this case, kmeans was used to cluster the vectorized comments and cosine similarity was

computed within each cluster in order to reduce the computation cost. Clusters of 10, 20, 50 and 200 were tested and a cluster of 200 was adopted due to the its minimum computation cost.

4. Determine the Threshold of High Similarity

Since there is no training set provided by Yelp, to be conservative, we use the 3rd quantile (0.744664) of the cosine similarity as the threshold. For businesses whose comments have cosine similarity higher than 0.744664 will be stored and recommended to Yelp as potential businesses that hired fake reviewers.

```
>>>  df_200_combined.angle.describe()
>>>  count    783601.000000
      mean          0.657639
      std           0.102723
      min           0.207543
      25%           0.580322
      50%           0.659148
      75%           0.744664
      max           1.000000
      Name: angle, dtype: float64
```



Recommendation

Suspicious business id and cosine score between their comments: final_suspects_detail.csv

Drawbacks and Improvement

1. Since Yelp did not provide training data for fake reviewers and it is unlikely to manually select fake reviews, it is difficult to do cross-validation and therefore suspicious reviewers cannot be determined as fake reviewers.
2. The method Bag-Of-Words does not consider the order of word when calculating the score between 2 reviews, nor does it consider the combination of words and therefore it is possible that 2 strings with high correlation may have completely different meanings. (*e.g.* "not very good" is similar to "very good" but they have very opposite meanings.) Therefore, we can try using Bag-Of-Words on expressions of several words, even though in this way the x matrix may become sparser.
3. Since the vectorized reviews are very sparse, it is computationally expensive to compute all the cosine score between every 2 reviews of different business id.