Restaurant Recommendations in Las Vegas: An Alternative approach.

Using Data Science to create Useful recommendations

Goal

Create a system which users input restaurants they like, and receive a list of interesting and informative recommendations.

➔ **Useful for travelers.**
Las Vegas has an extensive tourism industry.

➔ **Capable of making recommendations with sparse data**
Give people a reason to care.

➔ **Intelligent**
Provides user with information that is difficult to find by traditional means.

The Data Set

4 GB, **679540** REVIEWS for **4093** RESTAURANTS from **260014** USERS

- **Restaurant** information:

  [Alcohol, GoodForKids, Parking...]

  - Attributes contained practical information about the restaurant environment and features.

  [Pizza, Seafood, Steakhouse...]

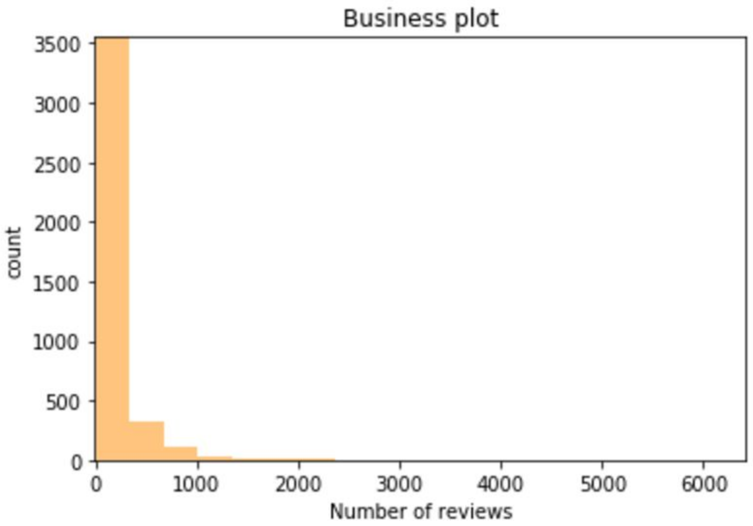
  - Category included information about cuisine, ambience, and other subjective information.

- **Users:**
  - Ratings for restaurants
  - Total number of reviews each user generated
  - Review text data

Highly Sparse Matrix

Median number of reviews each restaurant received:

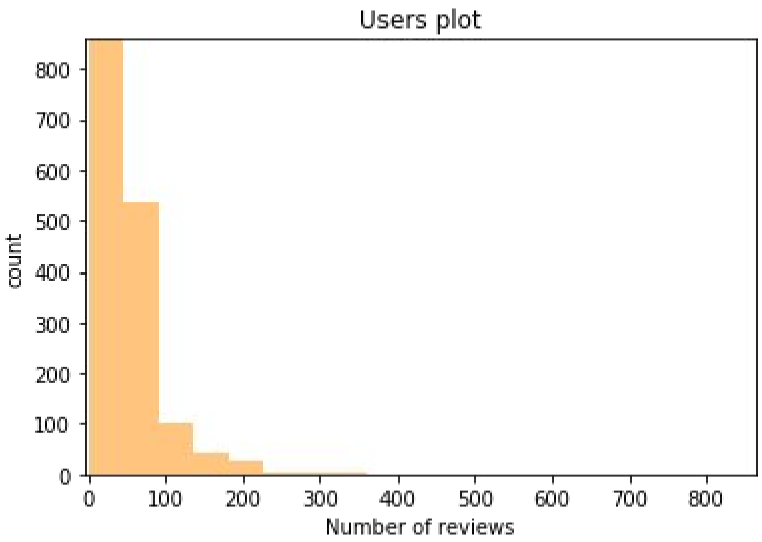
50



Few Active Users

Median number of reviews each user generated:

1

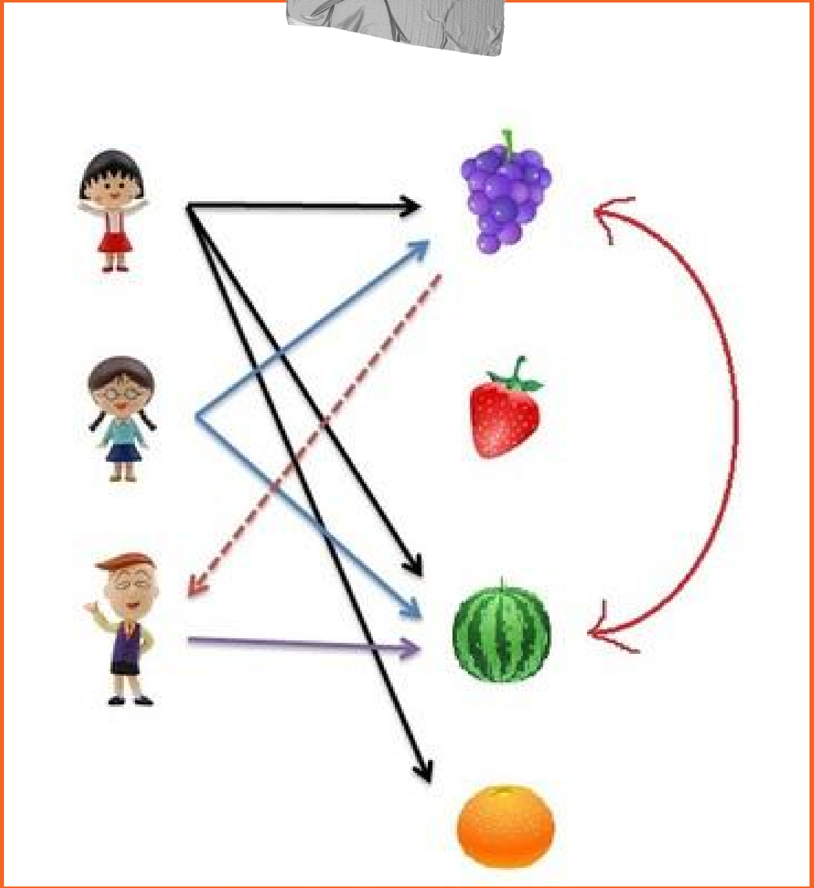


Focus on TOP 1% active users

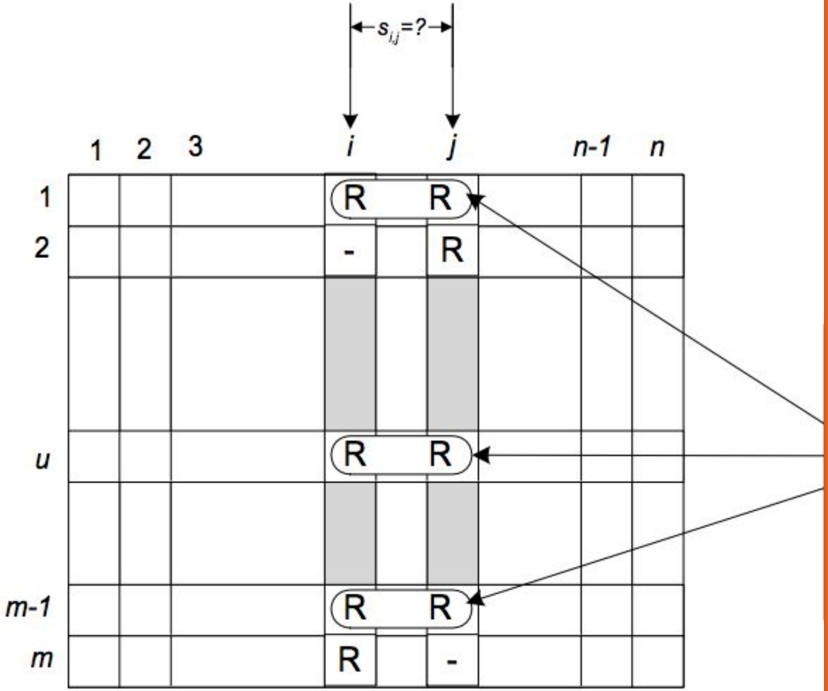
Users who reviewed 20+ restaurants
Covered 92% restaurants

Item-Based Collaborative Filtering

Calculate correlation between each pair of restaurants by observing all the users who have rated both restaurants



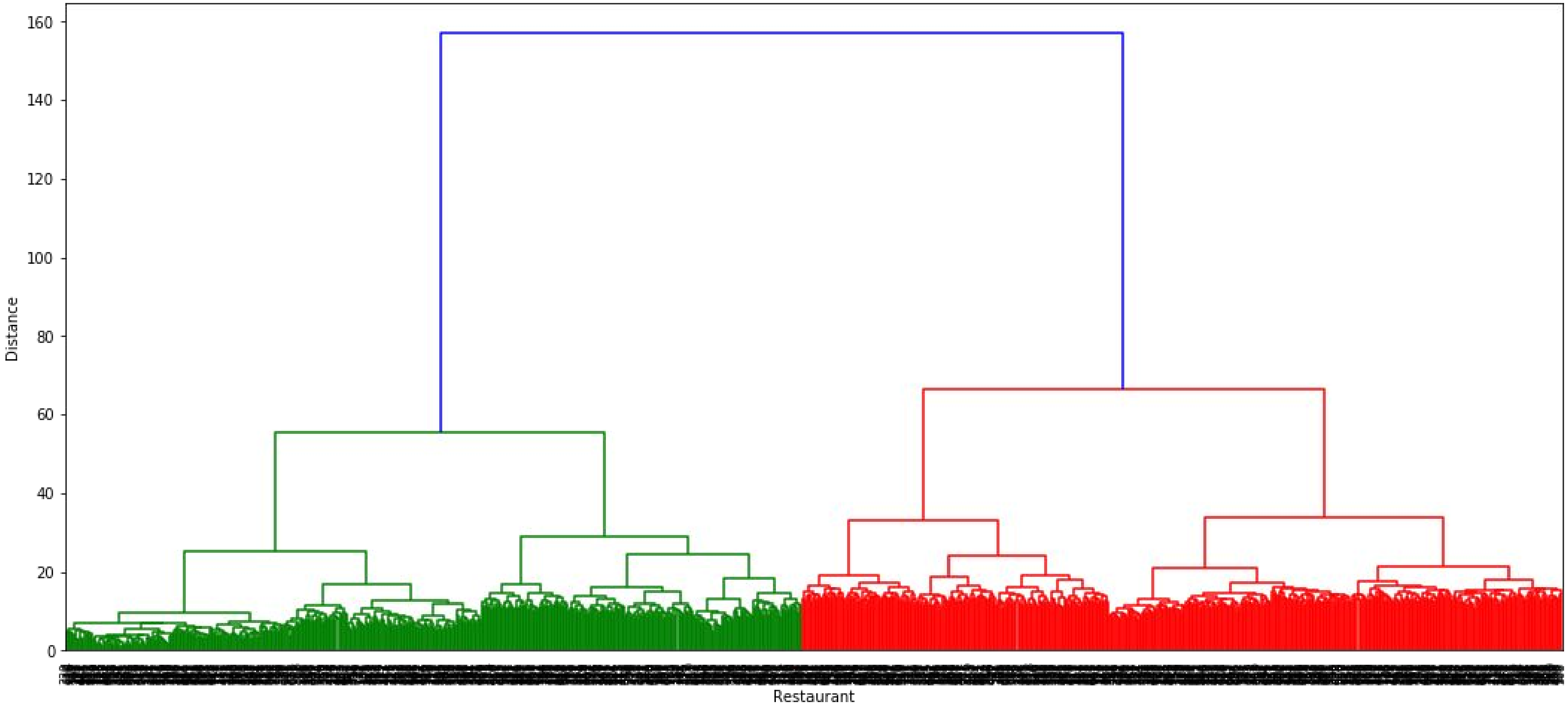
Co-rated items



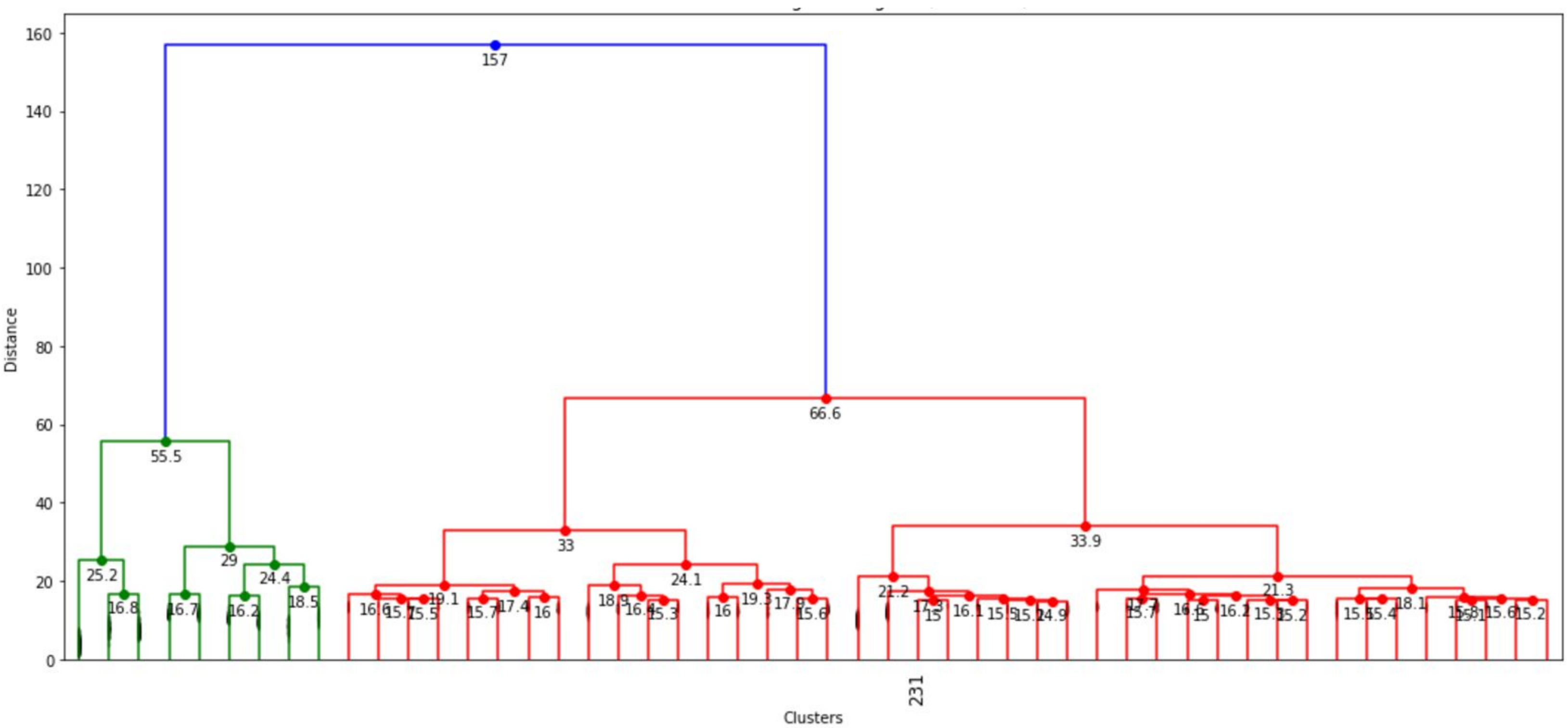
Correlation-based Similarity

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R}_i)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R}_j)^2}}.$$

Clustering!



Clustering!



Distance
Clusters
157
66.6
55.5
33
33.9
29
25.2
24.4
24.1
21.2
21.3
18.5
19.1
231

Content Based Recommendation

Review Text:

TF IDF

Category Data:

Euclidean

Distance

Attribute Data:

Binary Information about Features.

Text Data

Remove numbers, punctuation and stop-words using NLTK.

Stem and Lemmatize to recognize semantically identical words.

Having to Hav.

Am to be.

Extract top 1,000 words.

SKLearn to vectorize and analyze similarity.

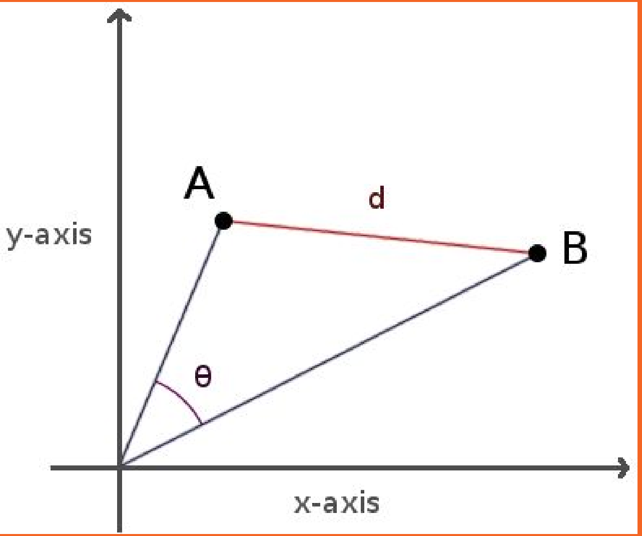
$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Category Data

Hundreds of categories.

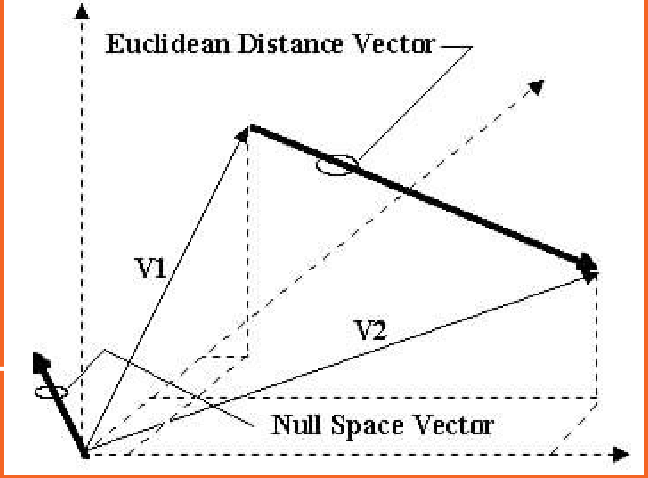
Binarized and analyzed as pre-processed text data using cosine similarity



Attribute Data

List of attributes about each restaurant.

Common length and structure made Euclidean distance an effective measurement.



Results were... Good?

At this point in time, we thought we were going to proceed to build machine learning models to decide which restaurants to recommend.

That approach was fundamentally wrong.

Pinpointing clusters of restaurants in a similar vector space does not meet our established objectives.

Provided uninteresting homogenous results.

Rethinking Our Methods

Techniques we used pinpointed close items in a vector space.

A list of similar restaurants is useless.

New goal:

- ➔ **Create a list that offers Novelty.**
- ➔ **Helps user Explore vector space.**
- ➔ **Identify unexplored areas of the vector space that may be interesting given user history.**

Product

Novel hybrid recommendation system that incorporates collaborative filtering, expert user identification, and arbitrary multidimensional similarity to help users explore new types of restaurants at the periphery of their established interests.

- ➔ **Picks top recommendations based on multiple levels of similarity.**
- ➔ **Offers clusters to dig deeper into individual categories.**
- ➔ **Weights popular, highly rated, and highly rated by serial reviewers.**

Demo