UNIV DS-1 PROJECT

BASED ON YELP DATASET

Recommendation system and Data analysis

PROJECT TEAM-

- **EKANKI AGARWAL**
- **SRISH KULKARNI**
- **BHASKAR BHARAT**
- **SARATH MANOJ**

CONTENT OUTLINE

WE TOOK 4 APPROACHES

Collaborative filtering

- Making Recommendations to Existing Users. Tried 4 different methods. Cosine similarity, Restaurant based, Customised to user.
- Recommendations Based on User Similarities
- Trained a Collaborative Filtering Model on the Reviews Dataset using 50 Latent Features for User and Restaurants

Social Network based Exploration

- Analysed the social interactions of user which include the friends, fans, reviews type (cool, funny etc)
- We realised that yelp has failed as social website. Very rarely did users have friends or fans, and when they did have those, it was single digit numbers.

Content based recommendation system

- Content based recommender works with data that the user provides and also the features or contents of items (restaurants in our case) to be recommended.
- We have used restaurant features as well as user profile data which is then used to make suggestions to the user about the restaurant.

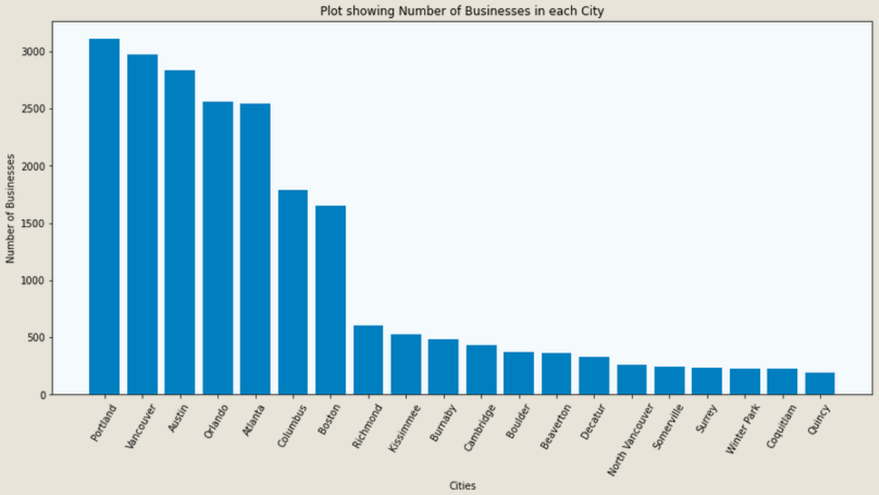
Location based Recommendation

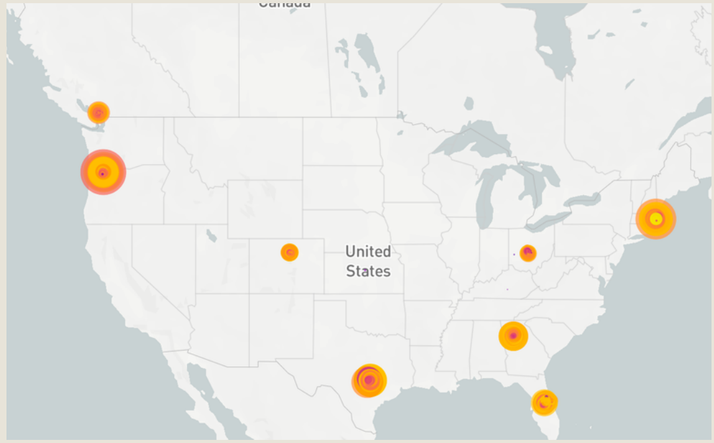
- We grouped restaurants together based on geographical location by using the K-Means Clustering Algorithm.
- The K-Means algorithm predicts the cluster where the user is located in and pulls out this cluster's top 5 restaurants and recommends them to the user.
- 1. Location Based Recommendation For New Users (To Resolve Cold-Start Problem)
- - Based on the City Entered by the User
- - Based on the Coordinates of the User

DATA ANALYSIS

From the data we understand:

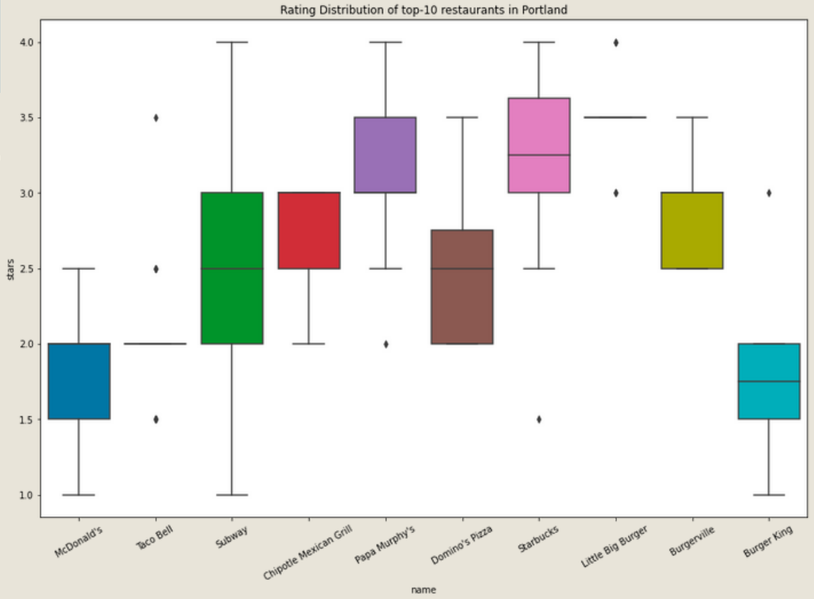
- Portland has the highest number of business





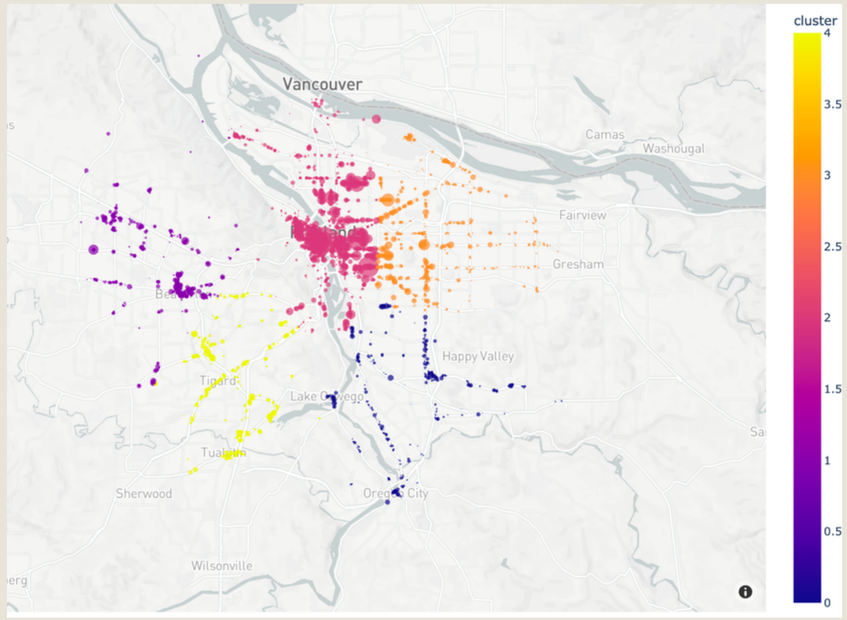
The distribution of ratings show us subway has the highest number and variance of rating in Portland. This could be because of the large number of Subways in this city.

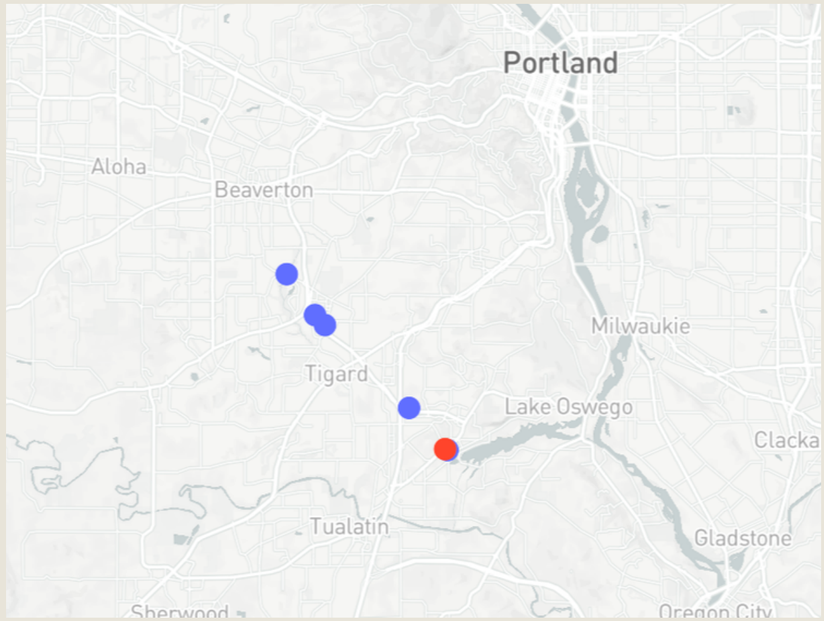
Portland is also the largest city in USA

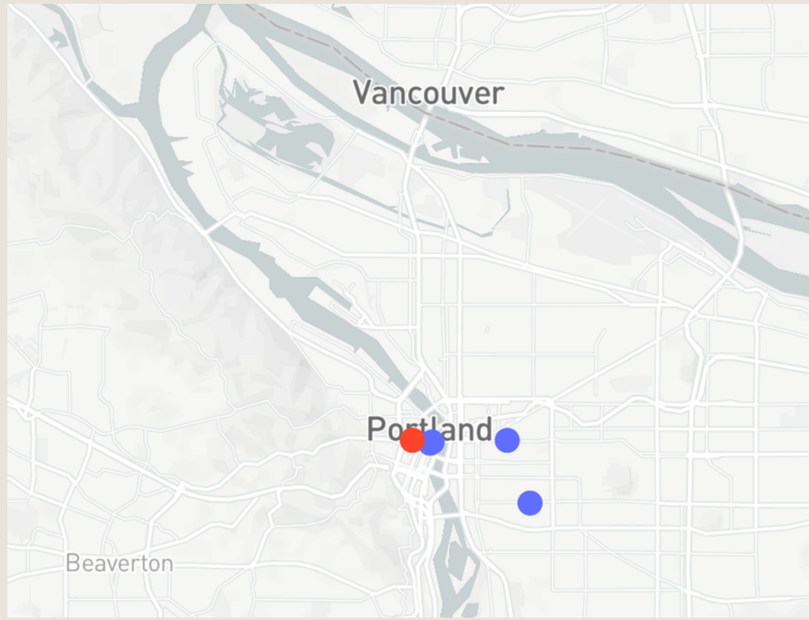


CLUSTERING ANALYSIS

CLUSTERING OF BUSINESS BY LOCATION AND RECOMMENDATION BASED ON LOCATION







EXPLORATARY ANALYSIS

With the clustering based on locations we can know that Vancouver and Portland had the highest business.

HOW IT FUNCTIONS

- Based on our model, it recommends top 5 restaurants based on user's latitude and longitude
- This model works on K-Means algorithm to find he nearest restaurant.

WHY PORTLAND

The data was too large and we just took Portland as an example since it had the highest number of restaurants.

CLUSTRING, PCA AND KMEANS

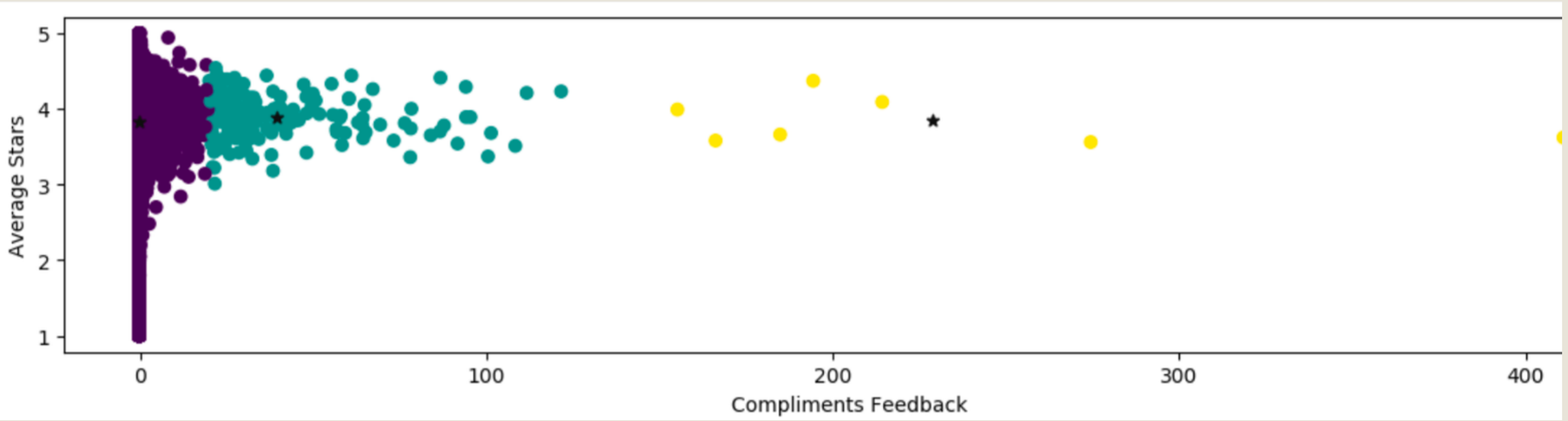
ANALYSIS OF USER DATABASE AND REVIEWS

	review_count	useful	funny	cool	fans	average_stars	compliment_hot	compliment_more	compliment_profile	compliment_cute	compliment_list	compliment_note	compliment_plain	compliment_cool	compliment_funny	compliment_writer	compliment_photos	friend_count	elite_count	yelp_since_YRMO	yelp_since_year	tagged_compliment	raters
review_count	1	0.66	0.57	0.59	0.49	-0.01	0.34	0.31	0.18	0.22	0.19	0.45	0.4	0.4	0.4	0.45	0.28	0.47	0.67	-0.17	-0.16	0.21	-0.099
useful	0.66	1	0.95	0.99	0.59	-9e-05	0.59	0.48	0.31	0.34	0.28	0.68	0.69	0.69	0.69	0.72	0.65	0.54	0.36	-0.075	-0.075	0.084	-0.035
funny	0.57	0.95	1	0.96	0.55	-0.0013	0.61	0.49	0.32	0.35	0.28	0.66	0.66	0.7	0.7	0.72	0.64	0.47	0.27	-0.063	-0.063	0.059	-0.028
cool	0.59	0.99	0.96	1	0.55	0.007	0.59	0.47	0.31	0.33	0.26	0.66	0.68	0.69	0.69	0.71	0.67	0.5	0.29	-0.054	-0.054	0.06	-0.018
fans	0.49	0.59	0.55	0.55	1	0.0092	0.4	0.32	0.19	0.32	0.22	0.52	0.57	0.46	0.46	0.46	0.36	0.57	0.33	-0.069	-0.069	0.079	-0.02
average_stars	-0.01	-9e-05	-0.0013	0.007	0.0092	1	0.0036	-0.0013	-0.00042	0.00064	-0.0021	0.00092	0.0069	0.0047	0.0047	0.0036	0.0062	0.047	0.02	0.039	0.039	-0.021	0.72
compliment_hot	0.34	0.59	0.61	0.59	0.4	0.0036	1	0.76	0.68	0.8	0.6	0.81	0.75	0.94	0.94	0.87	0.77	0.37	0.21	-0.064	-0.064	0.043	-0.017
compliment_more	0.31	0.48	0.49	0.47	0.32	-0.0013	0.76	1	0.93	0.81	0.89	0.76	0.6	0.73	0.73	0.83	0.62	0.3	0.2	-0.066	-0.066	0.052	-0.024
compliment_profile	0.18	0.31	0.32	0.31	0.19	-0.00042	0.68	0.93	1	0.83	0.92	0.62	0.44	0.62	0.62	0.71	0.55	0.17	0.1	-0.033	-0.033	0.023	-0.011
compliment_cute	0.22	0.34	0.35	0.33	0.32	0.00064	0.8	0.81	0.83	1	0.83	0.65	0.53	0.68	0.68	0.68	0.49	0.27	0.15	-0.069	-0.069	0.034	-0.017
compliment_list	0.19	0.28	0.28	0.26	0.22	-0.0021	0.6	0.89	0.92	0.83	1	0.58	0.43	0.54	0.54	0.64	0.43	0.19	0.12	-0.049	-0.049	0.025	-0.016
compliment_note	0.45	0.68	0.66	0.66	0.52	0.00092	0.81	0.76	0.62	0.65	0.58	1	0.87	0.88	0.88	0.83	0.69	0.49	0.29	-0.075	-0.075	0.065	-0.028
compliment_plain	0.4	0.69	0.66	0.68	0.57	0.0069	0.75	0.6	0.44	0.53	0.43	0.87	1	0.86	0.86	0.79	0.66	0.51	0.25	-0.057	-0.057	0.049	-0.014
compliment_cool	0.4	0.69	0.7	0.69	0.46	0.0047	0.94	0.73	0.62	0.68	0.54	0.88	0.86	1	1	0.91	0.84	0.43	0.25	-0.065	-0.065	0.052	-0.019
compliment_funny	0.4	0.69	0.7	0.69	0.46	0.0047	0.94	0.73	0.62	0.68	0.54	0.88	0.86	1	1	0.91	0.84	0.43	0.25	-0.065	-0.065	0.052	-0.019
compliment_writer	0.45	0.72	0.72	0.71	0.46	0.0036	0.87	0.83	0.71	0.68	0.64	0.83	0.79	0.91	0.91	1	0.81	0.43	0.29	-0.065	-0.065	0.059	-0.023
compliment_photos	0.28	0.65	0.64	0.67	0.36	0.0062	0.77	0.62	0.55	0.49	0.43	0.69	0.66	0.84	0.84	0.81	1	0.31	0.15	-0.02	-0.02	0.03	-0.0051
friend_count	0.47	0.54	0.47	0.5	0.57	0.047	0.37	0.3	0.17	0.27	0.19	0.49	0.51	0.43	0.43	0.43	0.31	1	0.37	0.052	0.052	0.061	0.024
elite_count	0.67	0.36	0.27	0.29	0.33	0.02	0.21	0.2	0.1	0.15	0.12	0.29	0.25	0.25	0.25	0.29	0.15	0.37	1	-0.14	-0.14	0.22	-0.073
yelp_since_YRMO	-0.17	-0.075	-0.063	-0.054	-0.069	0.039	-0.064	-0.066	-0.033	-0.069	-0.049	-0.075	-0.057	-0.065	-0.065	-0.065	-0.02	0.052	-0.14	1	1	-0.36	0.13
yelp_since_year	-0.16	-0.075	-0.063	-0.054	-0.069	0.039	-0.064	-0.066	-0.033	-0.069	-0.049	-0.075	-0.057	-0.065	-0.065	-0.065	-0.02	0.052	-0.14	1	1	-0.36	0.13
tagged_compliment	0.21	0.084	0.059	0.06	0.079	-0.021	0.043	0.052	0.023	0.034	0.025	0.065	0.049	0.052	0.052	0.059	0.03	0.061	0.22	-0.36	-0.36	1	-0.13
raters	-0.099	-0.035	-0.028	-0.018	-0.02	0.72	-0.017	-0.024	-0.011	-0.017	-0.016	-0.028	-0.014	-0.019	-0.019	-0.023	-0.0051	0.024	-0.073	0.13	0.13	-0.13	1

- Correlation matrix of unscaled original data which includes, elite count.

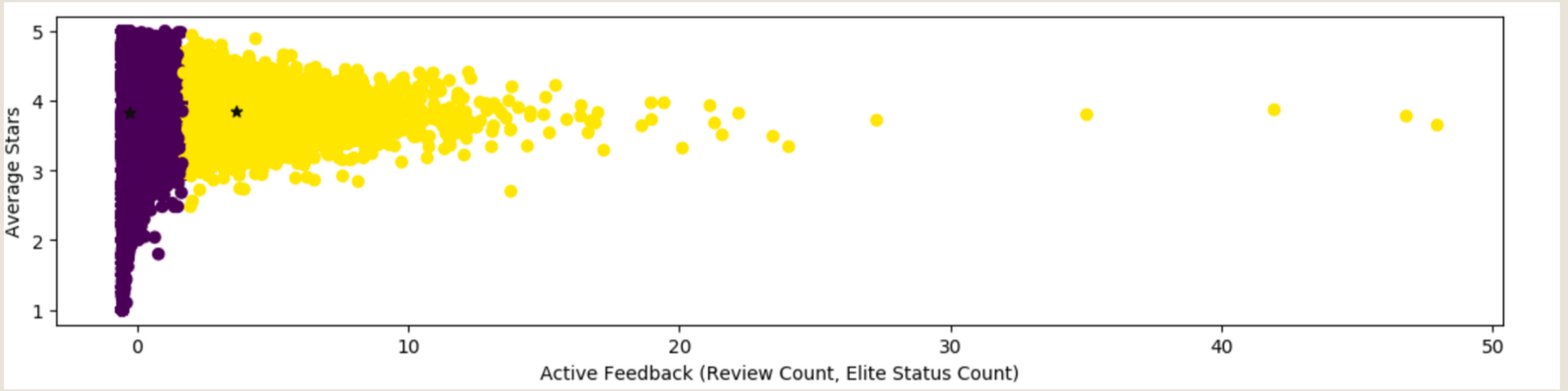
- Idea is to see which features are correlated and can be combined (PCA) .

- Review count is correlated with useful–funny–cool and elite_count

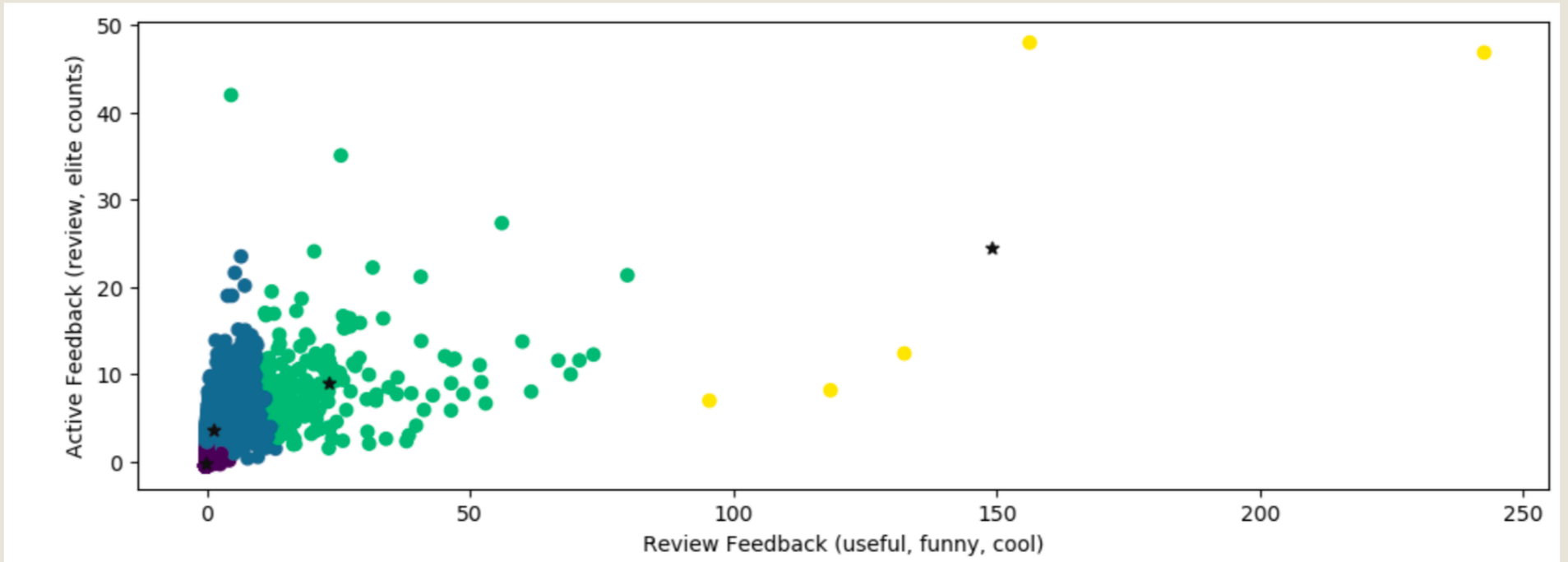


1. low_complimented users with low count of compliments_feedback rate across the spectrum of average ratings, but mostly staying at the center (3.5–4 avg).
2. Users with moderate compliments, rate more highly on average.
3. highly_complimented users, are low in numbers, as opposed to others, and rate 4 stars in most cases, but show wider variance as opposed to moderately complimented users

DIFFERENT FILTERING OF KMEANS CLUSTERING



We tried different filtering attributes to see how the rating is with different types of users. We checked the ratings of active users, Popular users etc against each other



It is interesting to see that Review Feedback is very poorly correlated with User Activity or active feedback. If we ignore, poor_reviewers clusters, which essentially just shows not active and not useful review writers, we can focus on other 3 segments.

CONTENT BASED RECOMMENDATION SYSTEM

CONTENT BASED RECOMMENDER WORKS WITH DATA THAT THE USER PROVIDES AND ALSO THE FEATURES OR CONTENTS OF ITEMS (RESTAURANTS IN OUR CASE) TO BE RECOMMENDED.

APPROACH 1: ATTRIBUTES AND CATEGORIES OF RESTAURANTS ARE ENCODED USING ONE-HOT ENCODINGS. RESTAURANT RATINGS AND REVIEW COUNTS ARE ALSO USED. THESE FORM FEATURE MATRIX AVAILABLE FOR ALL RESTAURANTS. COSINE SIMILARITY IS USED TO FIND RESTAURANTS SIMILAR TO A PARTICULAR RESTAURANT.

RESULT :

```
Query Restaurant: Oskar Blues Taproom
**********Recommended top 5 Similar Restaurants**********
['Carts On Foster' 'Boston Hot Dog Company' "McDonald's" "Daniel's Bakery"
 'Panda Express']
```

APPROACH 2: RECOMMENDING RESTAURANTS WHICH ARE SIMILAR TO USER'S PREFERENCES. USER PROFILE IS GENERATED REFLECTING THEIR PREFERENCES BASED ON THE DOT PRODUCT OF THEIR RATINGS AND THE EMBEDDED FEATURES OF THE RESTAURANTS. FINALLY, COSINE SIMILARITY IS USED TO RECOMMEND RESTAURANTS TO THE USER.

CASE1: REVIEWS GIVEN BY THE USER ARE ENCODED USING PRE-TRAINED EMBEDDINGS WEIGHTED BY TF-IDF (TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY) VALUES TO CREATE EMBEDDED FEATURES OF THE RESTAURANTS.

CASE2: BUSINESS ATTRIBUTES AND CATEGORIES ARE ENCODED TO CREATE EMBEDDED FEATURES OF THE RESTAURANTS. UNDER THIS, FOR ONE CASE, USER RATINGS ARE NORMALIZED USING MIN-MAX NORMALIZATION. IN ANOTHER CASE, INSTEAD OF USER RATINGS, RATING SENTIMENTS (USER RATING <3 IMPLIES NEGATIVE SENTIMENT (-1) WHILE USER RATING>=3 IMPLIES POSITIVE SENTIMENT (+1)) USING NORMALIZED RATINGS OR RATING SENTIMENTS LEAD TO SIMILAR RECOMMENDATIONS.

RESULT:

```
Query User: --0YW17u1XvJ75JTWzhzjw
**********Recommended top 5 Similar Restaurants**********
['LongHorn Steakhouse' 'Meeka Japanese Restaurant' 'LongHorn Steakhouse'
 "McGrath's Fish House" 'China Hot Pot']
```

COLLABORATIVE FILTERING

A) CONSIDERED ONLY THE TOP-2 CITIES WITH THE MAXIMUM NUMBER OF BUSINESSES FOR THE ANALYSIS. THE CITIES CONSIDERED ARE PORTLAND AND VANCOUVER.

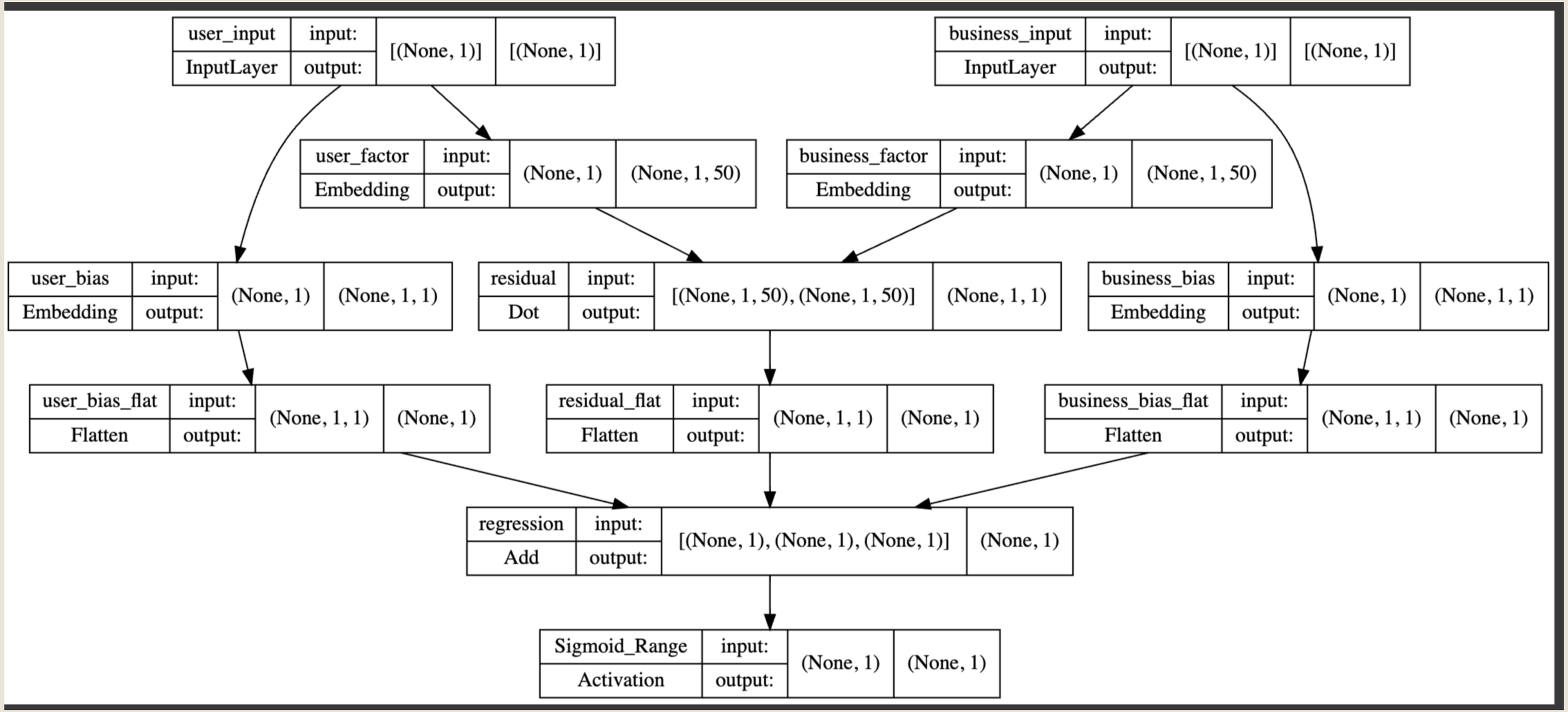
B) TRAINED A COLLABORATIVE FILTERING MODEL ON THE REVIEWS DATASET USING 50 LATENT FEATURES FOR USER AND RESTAURANTS

C) FOLLOWING RECOMMENDATIONS WERE PROVIDED USING THE CF-MODEL AND THE USER AND RESTAURANT EMBEDDINGS:

1. LOCATION BASED RECOMMENDATION FOR NEW USERS (TO RESOLVE COLD-START PROBLEM)
- BASED ON THE CITY ENTERED BY THE USER
- BASED ON THE COORDINATES OF THE USER

2. MAKING RECOMMENDATIONS TO EXISTING USERS
- COSINE SIMILARITY SCORE CALCULATION
- RECOMMENDATIONS WHEN A USER SEARCHES FOR A PARTICULAR RESTAURANT
- RECOMMENDING SIMILAR RESTAURANTS TO A PARTICULAR RESTAURANT
- CUSTOMIZING TO THE USER: RECOMMENDING SIMILAR RESTAURANTS TO USER (AN EXISTING USER) WHEN HE/SHE SEARCHES FOR A PARTICULAR RESTAURANT CUSTOMIZED TO USER PREFERENCES

3. RECOMMENDATIONS BASED ON USER SIMILARITIES


user_input
InputLayer
input:
output:
[(None, 1)]
[(None, 1)]
business_input
InputLayer
input:
output:
[(None, 1)]
[(None, 1)]
user_factor
Embedding
input:
output:
(None, 1)
(None, 1, 50)
business_factor
Embedding
input:
output:
(None, 1)
(None, 1, 50)
user_bias
Embedding
input:
output:
(None, 1)
(None, 1, 1)
residual
Dot
input:
output:
[(None, 1, 50), (None, 1, 50)]
(None, 1, 1)
business_bias
Embedding
input:
output:
(None, 1)
(None, 1, 1)
user_bias_flat
Flatten
input:
output:
(None, 1, 1)
(None, 1)
residual_flat
Flatten
input:
output:
(None, 1, 1)
(None, 1)
business_bias_flat
Flatten
input:
output:
(None, 1, 1)
(None, 1)
regression
Add
input:
output:
[(None, 1), (None, 1), (None, 1)]
(None, 1)
Sigmoid_Range
Activation
input:
output:
(None, 1)
(None, 1)

HOW TO IMPROVE THIS PROJECT

- We can build a hybrid model by using the output of both the collaborative and content based recommendations system. This can be done by assigning a score to all the top restaurents of the system. If any restaurant is on both , it will have a higher score and recommended first. This way we ensure that a user is recomended similar products and new options to try out.

- Build a better pipeline.

- Sentiment analysis of reviews.