MY YELPER - A HYBRID RECOMMENDATION ENGINE

10/4/2017

Data Science Project Final Report

Author: Sami Mustafa

Mentor: Alex Chao

Recommender systems enable users to access products or places that they would otherwise not be aware of due to the wealth of information to be found on the Internet. The two traditional recommendation techniques are content-based and collaborative filtering. While both methods have their advantages, they also have certain disadvantages, some of which can be solved by combining both techniques to improve the quality of the recommendation. The resulting system is known as a hybrid recommender system. This report describes the work and the results of building a hybrid recommendation engine for restaurants in the Yelp database.

This project has been done in fulfillment to the Second Capstone Project requirement of Springboard DS Bootcamp

My Yelper - A Hybrid Recommendation Engine

DATA SCIENCE PROJECT FINAL REPORT

Introduction

Nowadays, recommender systems are used to personalize our experience on the web, telling us what to buy, where to eat or even who we should be friends with. People's tastes vary, but generally follow patterns. People tend to like things that are similar to other things they like, and they tend to have similar taste as other people they are close with. Recommender systems try to capture these patterns to help predict what else we might like.

Two most ubiquitous types of recommender systems are Content-based Filtering and Collaborative Filtering. Collaborative filtering produces recommendations based on the knowledge of users' attitude to items, that is it uses the "wisdom of the crowd" to recommend items. In contrast, content-based recommender systems focus on the attributes of the items and gives recommendations based on the similarity between them.

A hybrid recommendation engine is built by combining various recommender systems to provide a more robust system. For example, by combining collaborative filtering methods, where the model fails when new items don't have ratings, with content-based systems, where feature information about the items is available, new items can be recommended more accurately and efficiently. Hybrid approaches can be implemented in several ways: by making content-based and collaborative-based predictions separately and then combining them; by adding content-based capabilities to a collaborative-based approach (and vice versa); or by unifying the approaches into one model.

In this project, we built a hybrid recommendation engine using content-based filtering and collaborative filtering for restaurants in Yelp. Additionally, and for more personalization, the social network of user's friends in Yelp will be used to add another recommender (friends' recommendation) to the hybrid engine.

Development Environment

- Apache Spark 2.2.0, Spark ML, Spark SQL
- PySpark on Jupyter Notebook, Python 3.5.2, Anaconda 4.2.0
- Apache Drill 1.11.0
- Ubuntu 16.10 (64-bit) Virtual Machine, quad-core i7 CPU, 4 GB RAM

Project repo on GitHub: https://github.com/samimust/my-yelper

Data Acquisition

This project uses the Yelp dataset available at https://www.yelp.com/dataset

The data set contains 4,700,000 reviews on 156,000 businesses in 12 metropolitan areas 1,000,000 tips by 1,100,000 users Over 1.2 million business attributes like hours, parking, availability, and ambience.

The data files are supplied in two flavors: json and SQL (MySQL, Postgres). This project utilizes the json version which has the following files:

- **business.json**: Contains business data including location data, attributes, and categories.
- **review.json**: Contains full review text data including the user_id that wrote the review and the business_id the review is written for.
- **user.json**: User data including the user's friend mapping and all the metadata associated with the user.
- **checkin.json**: Checkins on a business.
- **tip.json**: Tips written by a user on a business. Tips are shorter than reviews and tend to convey quick suggestions.
- **Photos.json**: (from the photos auxiliary file) this file is formatted as a JSON list of objects.

Each file is composed of a single object type, one JSON-object per-line. Description available on Yelp https://www.yelp.com/dataset/documentation/json

As the focus of this project is on building a recommendation engine, the core files will be used are **business.json**, **review.json**, and **user.json**. Additionally, taking into consideration the limited hardware resources used available, data for only one city will be considered for building the recommendation engine. The city of Toronto was chosen for its suitable data sizes (rank third in number of reviews).

Approach

Yelp users give ratings and write reviews about businesses and services on Yelp. These reviews and ratings help other Yelp users to evaluate a business or a service and make a choice. While ratings are useful to convey the overall experience, they do not convey the context which led a reviewer to that experience. Various Natural Language Processing (NLP) and Text Analytics techniques were applied on the review text to build features for the content-based filtering recommendations. The numeric ratings (1 to 5) will be used in the collaborative filtering recommendations. Friends' ratings aggregation will be used to derive friend's network recommendations. Figure 1 below shows the overall hybrid recommendation engine architecture and technology platforms utilized.

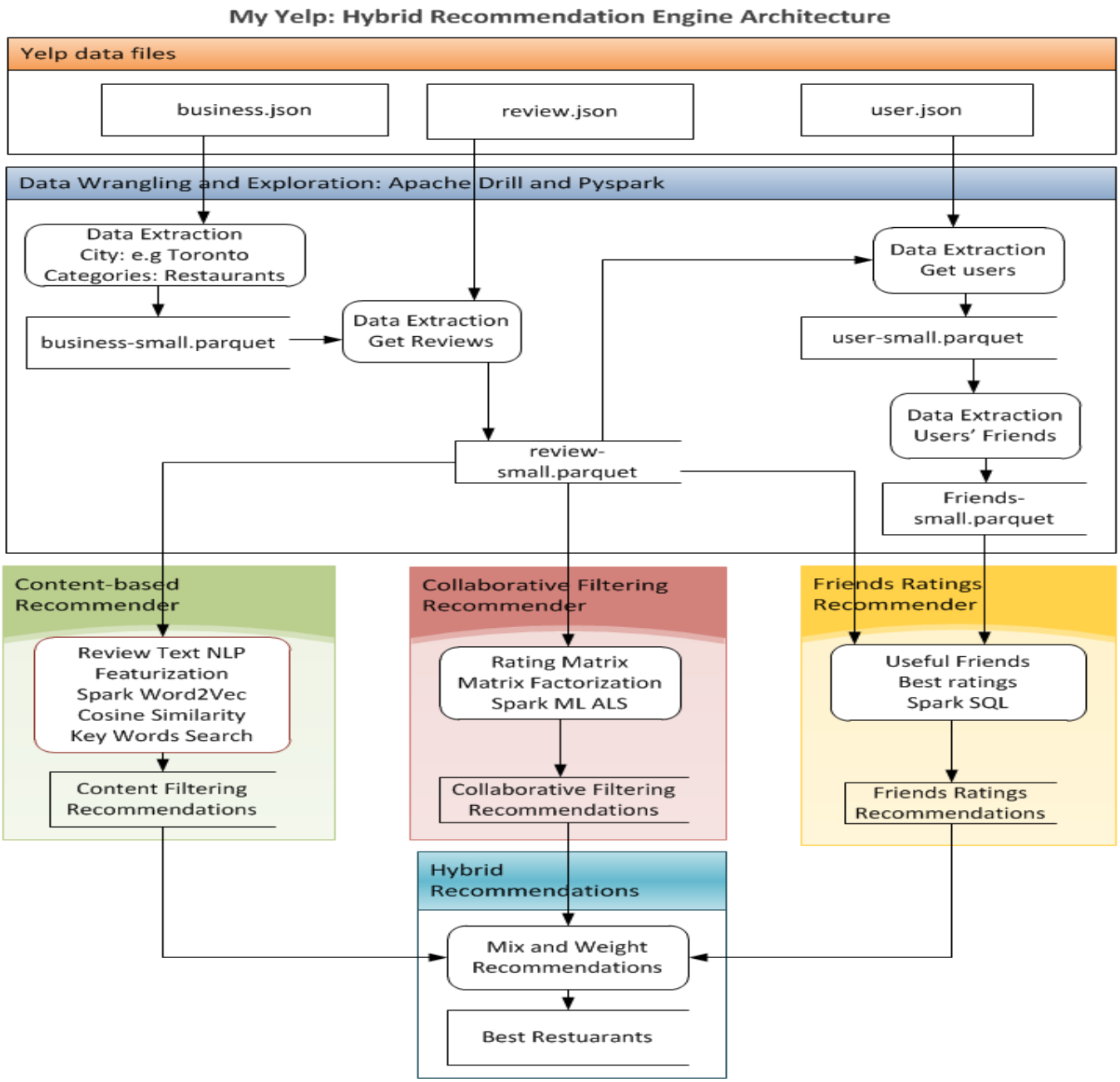


Figure 1: Hybrid recommendation engine architecture

Data Wrangling and Exploration

Data Extraction:

Apache Drill, an open-source software framework that supports data-intensive distributed applications for interactive analysis of large-scale datasets, was used to extract the slices of data needed for the

project. SQL queries were designed to extract Businesses, Reviews, Users, and Friends data for the city of Toronto from the large Yelp json datasets. Those extracted files were saved as Apache parquet format and then loaded into Spark dataframes and SQL views.

Data Loading Summary:

No. of Businesses: 6,750 (see sample below)

	business_id	business_name	neighborhood	address	city	state	postal_code	latitude	longitude	stars	review_count	categories
0	g6AFW-zY0wDvBl9U82g4zg	Baretto Caffe		1262 Don Mills Road	Toronto	ON	M3B 2W7	43.744768	-79.346527	5.0	146	[Italian, Restaurants, Cafes]
1	J9vAdD2dCpFuGsxPIn184w	New Orleans Seafood & Steakhouse		267 Scarlett Road	Toronto	ON	M6N 4L1	43.677592	-79.506363	5.0	62	[Cajun/Creole, Seafood, Steakhouses, Restaurants]

No. Users: 66,424 (see sample below)

	user_id	user_name	review_count	yelping_since	useful	funny	cool	fans	average_stars
0	om5ZiponkpRqUNa3pVPiRg	Andrea	2559	2006-01-18	83681	10882	40110	835	3.94
1	Wc5L6iuvSNF5WGBlqIO8nw	Risa	1122	2011-07-30	26395	4880	19108	435	4.10
2	uxKSnOVAoEj4I6X9YhLBlg	Vivian	73	2013-03-02	34	5	2	8	3.54
3	s8bVHRqx6cI8F8HGf3A_og	Colleen	32	2014-12-18	19	3	7	2	4.15

No of Friends: 2,047,943 (see sample below)

	user_id	friend_id
0	om5ZiponkpRqUNa3pVPiRg	eoSSJzdprj3jxXyi94vDXg
1	om5ZiponkpRqUNa3pVPiRg	QF0urZa-0bxga17ZeY-9lw
2	om5ZiponkpRqUNa3pVPiRg	U_sn0B-HWdTSlHNXIl_4XA
3	om5ZiponkpRqUNa3pVPiRg	1_4Q1prE_QcejmEH5Dp0Bw
4	om5ZiponkpRqUNa3pVPiRg	DhBu8qqXHqVVZGx73NFzpQ

No. of Reviews: 276,887 (see sample below)

	review_id	user_id	business_id	stars	review_date	review_text	useful	funny	cool
0	Z5l99h18E3_g1GLcDSsWqA	djpMXOA1ic5wv3FPtubHNw	mr4FiPaXTWlJ3qGzp4-7Yg	3	2009-07-21	I left Table 17 feeling very ambivalent. Meh a...	3	0	0
1	Z3Fw292i0Eg8liW0DT1jsw	-pXs08gJq9ExIk275YLvPg	mr4FiPaXTWlJ3qGzp4-7Yg	3	2008-12-13	for the time being, for all its worth, i am go...	1	0	0
2	hsKINx1dIKeFTDe-ZlCvgA	PTj29rhujYETuFlAZaDi3w	mr4FiPaXTWlJ3qGzp4-7Yg	5	2013-10-12	Love this place. I went there with me boyfrien...	1	0	1
3	oviMS8F4ACflGysxsXKmew	3hLMY2dBEP1kYbd_ywTsCQ	mr4FiPaXTWlJ3qGzp4-7Yg	5	2013-02-17	Had a lovely evening last night at Table 17. ...	0	0	0

Review Ratings Analysis:

As shown in Figure 2 and 3 below the number review ratings in the range of 4 and 5 (good reviews) are bigger than rating of 1 and 2 (bad reviews), and the trend is increasing over time.

Figure 2: Review Rating Count

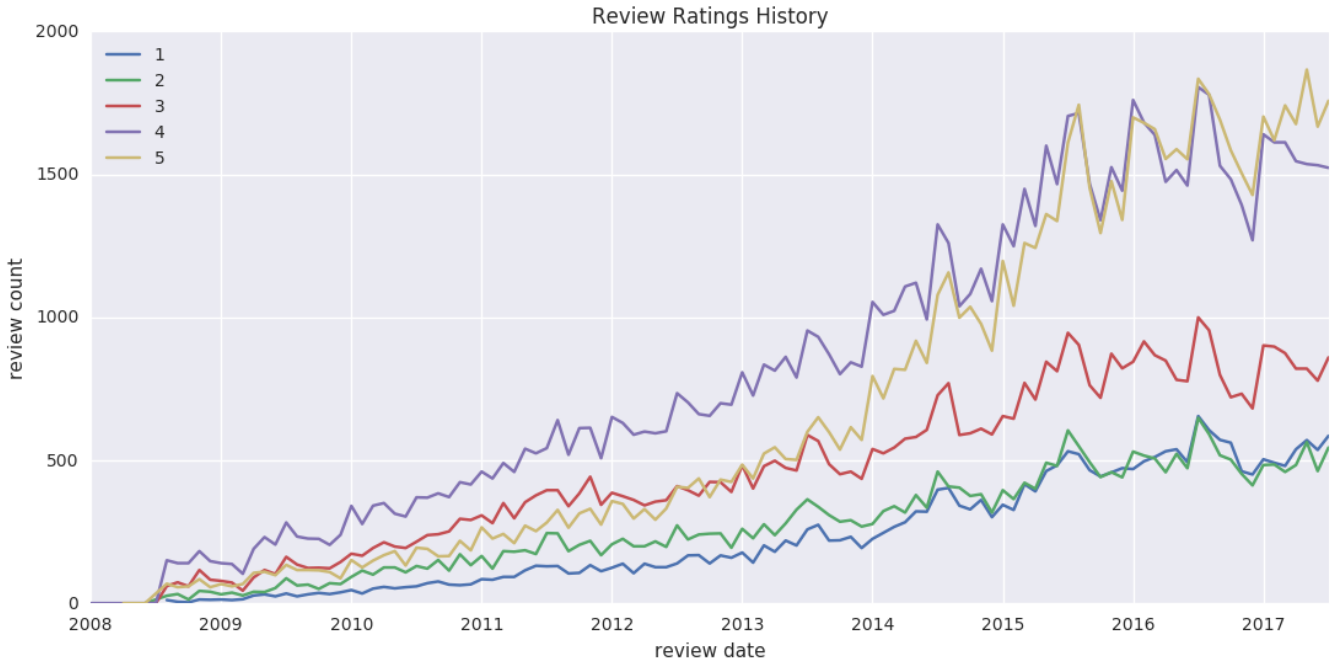


Figure 3: Review Rating Trend

Review Text Processing and Analysis:

Recommendation based on content-based filtering depends on items attributes and the similarity between them. To build item attributes in this project, we decided to use the aggregated review text for each

restaurant to create a rich set of features per restaurant, and then use those features to compute similarities between restaurants. The review text processing pipeline is shown in Figure 4 below:

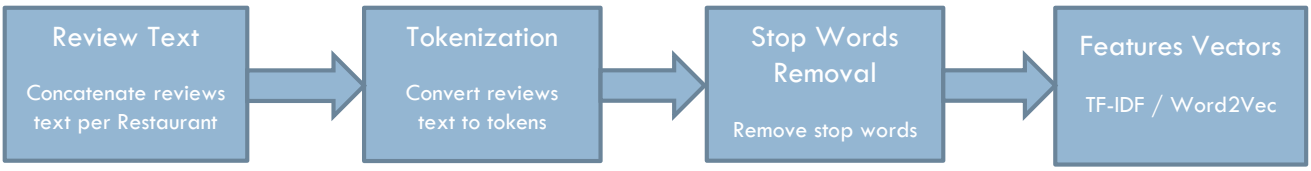


Figure 4: Review Text Processing Pipeline

In general, before converting words or sentences to vectors, some Natural language processing (NLP) techniques should be performed to transform the raw text into a form that should produce better outcome using the resultant features. These steps usually consist of: tokenizing the text into individual n-gram words (tokenization); remove stop words like "the" "and" "I am"; stemming i.e. convert all words to their root; and then finally use a vectorization method like TF-IDF, or Word2Vec to convert the final result into vectors, which is an easy way to perform various type of computations.

At time of building this recommender, Apache Spark has not yet included a word stemmer in its set ML transformation modules. Although, we considered including some external stemming module like Stanford CoreNLP, but for time-limit consideration we chose to go without stemming.

TF-IDF:

Term Frequency–Inverse Document Frequency (TF-IDF), is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval, text mining, and user modeling. The TF-IDF value increases proportionally to the number of times a word appears in the document, but is often offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general.

Word2Vec:

Word2vec is a group of related models that are used to produce word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space.

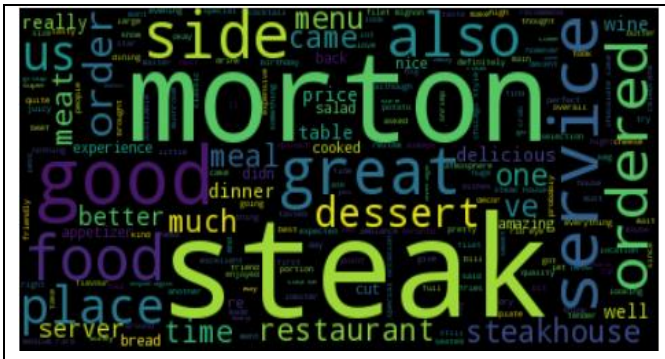
A sample of the text processing pipeline steps output is shown below:

	business_id	text	token	nostopwrd	idf_vec	word_vec
0	bfR-vJvrjdOJaWsXGJgzPA	Attention allergy sufferers: Claims to have a ...	[attention, allergy, sufferers, claims, to, ha...	[attention, allergy, sufferers, claims, nut, f...	(2.32949090224, 0.946347870074, 0.824754840176...	[-0.0949216104917, -0.0313844011, 0.0271677521...
1	Dl2vgi5W_nbe-A97D0zgfA	I don't understand previous review. I went the...	[i, don, t, understand, previous, review, i, w...	[understand, previous, review, went, three, ti...	(0.325045242174, 0.141952180511, 0.11782212002...	[-0.0657136221337, 0.0038411388924, 0.02664112...
2	65ZGMedBm7TBpWv6fzH2_Q	Food here is always fresh and healthy definite...	[food, here, is, always, fresh, and, healthy, ...	[food, always, fresh, healthy, definitely, gre...	(0.0541742070289, 0.0, 0.117822120025, 0.44249...	[-0.00367327128618, -0.0228931885972, -0.00643...

The resultant vectors (features) will be used to compute similarity score between restaurants in preparation for content-based filtering.

We computed the cosine similarity for some sample restaurants using both TF-IDF and Word2Vec vectors individually or combined and elected to use Word2Vec only for its superior performance both in time and similarity results.

Word clouds were also utilized to provide visual clues on the reviews text content. Below are some similarity test results:




```
+----------------------+--------------------------+
|business name         |categories                |
+----------------------+--------------------------+
|Morton's The Steakhouse|[Steakhouses, Restaurants]|
|Jacobs & Co. Steakhouse|[Restaurants, Steakhouses]|
+----------------------+--------------------------+

cosine similarity based on IDF vectors     : 0.334639335888
cosine similarity based on Word2Vec vectors: 0.980275255128
```


```
+---------------------+-----------------------------------------------------------------------------------+
|business_name        |categories                                                                         |
+---------------------+-----------------------------------------------------------------------------------+
|KINKA IZAKAYA ORIGINAL|[Pubs, Japanese, Restaurants, Bars, Nightlife, Tapas Bars, Tapas/Small Plates]|
|KINKA IZAKAYA BLOOR   |[Nightlife, Restaurants, Pubs, Japanese, Tapas/Small Plates, Tapas Bars, Bars]|
+---------------------+-----------------------------------------------------------------------------------+

cosine similarity based on IDF vectors     : 0.899718077666
cosine similarity based on Word2Vec vectors: 0.993977676194
```

Topic Modeling using LDA:

We also considered using Topic Modeling techniques to generate most important topics and features from the corpus of review text using Latent Dirichlet Allocation (LDA), which is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. LDA is an unsupervised learning method that maximizes the probability of word assignments to one of K fixed topics. The topic meaning is extracted by interpreting the top N probability words for a given topic, i.e. LDA will not output the meaning of topics, but rather it will organize words by topic to be interpreted by the user.

The ten topics and topic words identified by LDA are shown below. Clearly, some of them are about specific cuisines like Italian, Indian, Mexican or Japanese.

```
---------
topic: 0:  pizza pasta italian originally pizzas gelato wine tiramisu gnocchi ravioli
---------
topic: 1:  thai tacos pad burrito mexican taco curry balance jerk shawarma
---------
topic: 2:  burger ramen burgers patty kinton noodles games rings fries carrots
---------
topic: 3:  guu izakaya persian harlem kabob estimate sake brewhouse der appointment
---------
topic: 4:  pho vietnamese distance banh mi matcha tartar greater fixtures former
---------
topic: 5:  asshole perk simmered kare poutine knick surpass limes caeser replaced
---------
topic: 6:  classical nuff instant describes fryers multiple unassuming pho substandard capri
---------
topic: 7:  indian naan paneer buffet tikka butter addition masala coworkers sizzling
---------
topic: 8:  sushi brunch fries pizza pork coffee bar beer steak burger
---------
topic: 9:  christie pancetta drip frito avocados schnitzel nary fridges pizzaiolo las
```

Data Wrangling and Exploration Notebook on GitHub

Content-based Filtering

In a content-based recommender system, keywords (vectors) are used to describe the items and a user profile is built to indicate the type of item this user likes. In other words, these algorithms try to recommend items that are similar to those that a user liked in the past. In particular, various candidate items are compared with items previously rated by the user and the best-matching items are recommended.

A content-based profile of a user is usually built based on a weighted vector of items features. The weights denote the importance of each feature to the user and can be computed from individually rated content vectors using a variety of techniques. Simple approaches use the average values of the rated item vector while other sophisticated methods use machine learning techniques such as Bayesian Classifiers, cluster analysis, decision trees, and artificial neural networks in order to estimate the probability that the user is going to like the item.

In this project, we chose to build user's profile based on randomly selected restaurants that were top rated (stars > 3) by the user previously. This dynamic set representing the user profile or preference is

then compared to other existing restaurants to get best similar restaurants not seen by the user. This approach could be later improved with weighted user profile by computing weighted sum of the vectors for all items, with weights being based on the user's rating.

Cosine Similarity:

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0° is 1, and it is less than 1 for any other angle. It is thus a judgment of orientation and not magnitude: two vectors with the same orientation have a cosine similarity of 1, two vectors at 90° have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude. Cosine similarity is particularly used in positive space, where the outcome is neatly bounded in [0,1].

Given two vectors of attributes, A and B, the cosine similarity, cos(θ), is represented using a dot product and magnitude as:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|_2 \|\mathbf{B}\|_2} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$, where A_i and B_i are components of vector A and B respectively.

Content-based Filtering Results:

Based on features extraction, done by the review text processing pipeline and the user's profile created from restaurants previously top rated by the user, recommendations were generated based on the computed cosine similarity score. Top 10 restaurants recommendations for a user are shown below:

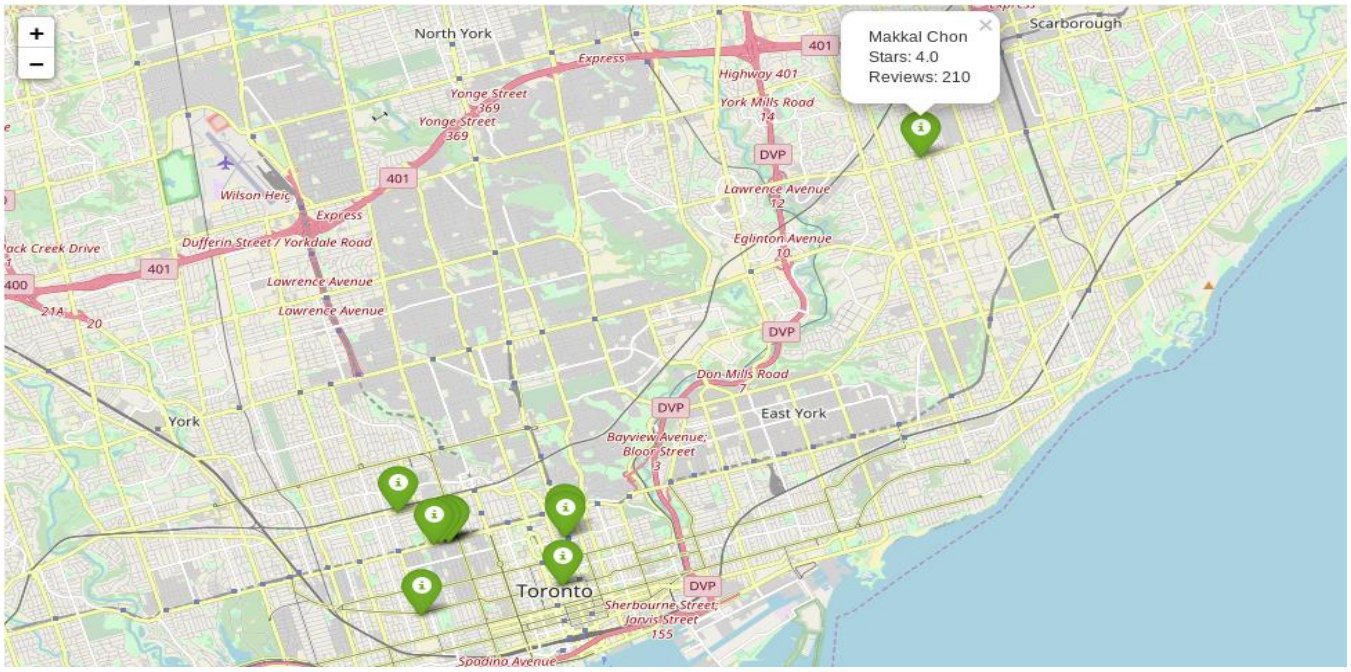
User Profile:

```
Businesses previously reviewed by the user: 'Wc5L6iuvSNF5WGBlqIO8nw'
+----------------------+--------------------------+-----------------------------------------------------------+
|business_id           |business_name             |categories                                                 |
+----------------------+--------------------------+-----------------------------------------------------------+
|F_oPMHJrH42R67xp5eKtQA|Yummy Korean Food Restaurant|[Korean, Restaurants]                                      |
|_HqZL3gK98-Q4ObAoyM1aw|Rose and Sons Swan          |[Breakfast & Brunch, American (Traditional), Restaurants]|
|aAYl1-tsvP9Kzs7YESi8Rg|Electric Mud BBQ            |[Restaurants, Sandwiches, Food, Barbeque, Desserts]        |
|acvmtF4ls5Qa1ZNadUV0Dw|Average Joe's Café          |[Breakfast & Brunch, Restaurants, Food, Bagels]            |
|M8S7poDhCIzqQx1--GB-ww|La Cubana                   |[Cuban, Restaurants, Mexican]                              |
+----------------------+--------------------------+-----------------------------------------------------------+
```

Recommendations:

	business_id	score	business_name	categories	stars	review_count	latitude	longitude
0	rO3lZpVSoRMhhd0AEJBjlg	0.986259	Sunrise House	[Restaurants, Korean]	4.0	135	43.664068	-79.415668
1	rhyjGfqYlCJoi8Zeulg6QA	0.983902	Kimchi Korea House	[Korean, Restaurants]	3.5	155	43.655256	-79.385475
2	j-Z_HAev26ZftdErMhIBuA	0.980066	Thumbs Up Korean Restaurant	[Restaurants, Korean]	4.0	56	43.664451	-79.413786
3	_MA98TVmvVIy-XdI0poc7w	0.979998	Mom's Korean Food	[Korean, Restaurants]	3.5	62	43.664686	-79.413785
4	SNkkuchbVtUzCwyENcai_g	0.979673	Danji	[Restaurants, Chinese, Japanese, Korean]	3.5	57	43.665300	-79.384899
5	X6ZZksefmR_piQj2Gbnduw	0.975525	Paldo Gangsan	[Restaurants, Korean]	4.0	47	43.663799	-79.417393
6	oQylTvXwGIkKFdCjmafKVg	0.973422	Fire on the East Side	[Southern, Restaurants, Breakfast & Brunch, Am...	3.5	119	43.666765	-79.384836
7	WnUttoJffplgWaQGR2J2Xw	0.973141	The Saint Tavern	[Restaurants, Bars, Nightlife, Gastropubs]	3.5	121	43.649062	-79.420478
8	ShUh_MMkaVp_KXCtNjPvXA	0.973117	Universal Grill	[American (Traditional), Canadian (New), Break...	3.5	45	43.670521	-79.426440
9	uChTCA6MsLAciDRklpO-Fw	0.972612	Makkal Chon	[Greek, Restaurants, Korean]	4.0	210	43.744944	-79.296636

Show in map:



Key Words Search:

To solve the cold-start problem (users with no history or profile in the system), we added recommendation by key words search. The word vectorization process described above is applied on input search words, and then similarity is computed. For example, below is the key word search: "chicken cheese burger" results:

	business_id	score	business_name	categories	stars	review_count	latitude	longitude
0	37joQpD9m5AIcrW1c8OBnQ	0.718460	Urban Smoke Fusion BBQ Food Truck	[Desserts, Barbeque, Food, Restaurants, Food T...	4.0	8	43.718711	-79.470037
1	3Cu-af4en3uWCrAkkqfiHQ	0.697351	Epic Burgers and Waffles	[Burgers, Food, Restaurants]	2.5	5	43.632351	-79.421280
2	nP87zXxeS-8got7IBvoAuA	0.662935	McCoy Burger Company	[Local Flavor, Sandwiches, Restaurants, Poutin...	4.0	33	43.731511	-79.404081
3	DiCMYxT69I22-1nfsvYAJQ	0.662169	Gourmet Burger Co	[Burgers, Restaurants]	3.5	37	43.664683	-79.368279
4	ky9RbwLtChekSrqcYR39kw	0.652767	Big Smoke Burger	[Burgers, Poutineries, Restaurants]	3.0	6	43.611289	-79.556867
5	ZzF5098L4xg-0COjng2LVA	0.648873	Burgatory	[Pubs, Burgers, Food Trucks, Nightlife, Bars, ...	3.0	9	43.655055	-79.418563
6	UN0UwUh7jaeX6Jg3lZImCg	0.644995	Holy Chuck	[Food, Restaurants, Desserts, Poutineries, Bur...	3.0	43	43.665211	-79.384925
7	ycAW6Q5quaCSDX5zwQ3tPg	0.640920	New York Fries	[Canadian (New), Specialty Food, Food, Restaur...	3.5	8	43.776875	-79.256655
8	PkeaeQS8aJTeS8PS_Hl_-g	0.635419	Steak and Cheese Factory	[Sandwiches, Cheesesteaks, Restaurants]	3.0	3	43.708213	-79.392367
9	67Pa_CtXthgJzXfY8JzLDQ	0.635081	Holy Chuck	[Burgers, Restaurants]	3.5	263	43.687527	-79.394060

Content-based Filtering Notebook on GitHub

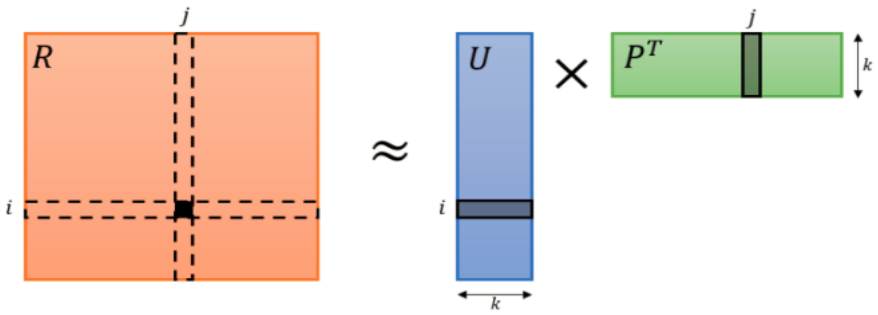
Collaborative Filtering

Collaborative Filtering is a subset of algorithms that exploit other users and items along with their ratings (selection, purchase information could be also used) and target user history to recommend an item that target user does not have ratings for. Fundamental assumption behind this approach is that other users preference over the items could be used recommending an item to the user who did not see the item or purchase before. This differs from content-based methods in the sense that the item itself does not play a role in recommendation.

Alternating Least Square (ALS):

The main input to collaborative filtering is the Rating Matrix R, which consists of the user_id, item_id, and rating. This large user/item matrix is decomposed into lower dimensional user factors and item factors using Matrix Factorization (MF). MF is a form of optimization process that aims to approximate the original matrix R with the two matrices U and P, such that it minimizes the following cost function:

$$J = ||R - U \times P^T||_2 + \lambda\left(||U||_2 + ||P||_2\right)$$



The first term in this cost function is the Mean Square Error (MSE) distance measure between the original rating matrix R and its approximation **U x P^T** . The second term is the "regularization term" and is added to govern a generalized solution.

The fact that both U's and V's values are unknown variables makes this cost function non-convex. If we fix P and optimize for U alone, the problem is simply reduced to the problem of linear regression or ordinary Least Square (OLS).

Alternating Least Squares (ALS) does just that. It is a two-step iterative optimization process. In every iteration it first fixes P and solves for U, and following that it fixes U and solves for P. Since OLS solution is unique and guarantees a minimal MSE, in each step the cost function can either decrease or stay unchanged. Alternating between the two steps guarantees reduction of the cost function, until convergence.

Following the fact that each known value in the rating matrix R was decomposed into the dot product of its matching user/item factor vectors, it is pretty straightforward to reconstruct a "full" ratings matrix R* by multiplying each user factor vector with every item factor vector. R* can be interpreted as the set of

the expected ratings given by any user to any item, given the collaborative patterns learned from the known values in R. Thus, it is straightforward to use all expected item-ratings in R* that were previously unknown in R, for a certain user, to produce a potentially well-ordered recommendation list of items never seen before by that user.

Apache Spark implements collaborative filtering using ALS. We trained this model in the constructed rating matrix and were able to get Root-Mean-Square Error of 1.258 using cross validation for hyper-parameters tuning. Better results could be obtained using high end computing hardware.

Collaborative Filtering Results:

```
Rating matrix no. of rows: 276,887
```

	userId	businessId	rating
0	23561	872	5.0
1	6268	872	4.0
2	8646	872	4.0
3	531	4253	3.0
4	2217	4253	5.0

```
Collaborative Filtering Recommendations for the user: 'ZWD8UH1T7QXQr0Eq-mcWYg'
```

	business_id	rating	business_name	categories	stars	review_count	latitude	longitude
0	LcIgUlWaJJwtOfPoPWCmBg	4.570364	Souppe Shoppe	[Restaurants, Street Vendors, Food, Soup, Food...	5.0	4	43.651425	-79.404123
1	mpDcuUs6dB5uBsYVKDWCNQ	4.527902	Druxy's Famous Deli	[Restaurants, Sandwiches, Delis, Breakfast & B...	4.0	4	43.648235	-79.379525
2	1VAsBosvx02jpvIUxiKvmg	4.490123	The Dumpling Shop	[Restaurants, Specialty Food, Chinese, Dim Sum...	4.5	11	43.767971	-79.401363
3	9GLN1xfck07CKfNfejKCwg	4.438345	T-Sushi	[Food, Restaurants, Sushi Bars, Food Delivery ...	5.0	13	43.644745	-79.390892
4	vAz5pelrjwkpMDo_OHCDAg	4.414823	Kuya Willie's Kainan	[Breakfast & Brunch, Filipino, Restaurants]	3.5	3	43.759288	-79.310866
5	y9yeMK6N0UINVECI3Ijz3Q	4.401293	Hot Dog Stand	[Hot Dogs, Restaurants]	4.0	3	43.681236	-79.377222
6	XKa5R1lJSvNrbo8InhNliQ	4.399106	Toronto Star Food Building	[Food, Fast Food, Restaurants]	4.5	3	43.632265	-79.420313
7	LIjlU7K-0SPXPtYFQiXamQ	4.392767	Magic Oven	[Food Stands, Sandwiches, Restaurants, Indian]	5.0	3	43.652294	-79.405521
8	fxRcHzovnRyWh_WMdQoNOQ	4.377005	Taj Restaurant	[Restaurants, Russian, Mediterranean]	5.0	4	43.696764	-79.446227
9	2H5EaBEreDzzP7sPmD_oDQ	4.362335	Vila Verde	[Restaurants, Event Planning & Services, Portu...	4.0	4	43.651243	-79.410631

Collaborative Filtering Notebook on GitHub

Friends' Network Recommendation

In this type of the recommendation, we chose to include restaurants as being top ranked by user's friends (4 or 5 stars). We have the assumption that a user trusts the agreement of his friends and their opinions. Only the first layer of the user's friends' network was considered.

Apache Spark SQL was the platform chosen for generating friends' recommendations.

Friends Recommendations results:

	business_id	4_5_stars_count	business_name	categories	stars	review_count	latitude	longitude
0	SGP1jf6k7spXkgwBlhiUVw	5	Kekou Gelato House	[Food, Restaurants, Ice Cream & Frozen Yogurt,...	4.5	332	43.655983	-79.392686
1	kOFDVcnj-8fd3doIpCQ06A	5	Mildred's Temple Kitchen	[Comfort Food, Event Planning & Services, Vege...	4.0	472	43.639911	-79.420424
2	0a2O150ytxrDjDzXNfRWkA	4	Miku Toronto	[Sushi Bars, Restaurants, Seafood, Japanese]	4.0	384	43.641235	-79.377370
3	G6EkDTXZ6zMUovg7JTG4YQ	3	Vietnam Noodle Star	[Restaurants, Vietnamese, Noodles]	3.5	148	43.804603	-79.287842
4	RwRNR4z3kY-4OsFqigY5sw	3	Uncle Tetsu's Japanese Cheesecake	[Desserts, Japanese, Restaurants, Bakeries, Food]	3.5	806	43.655969	-79.384013
5	Yv4P4qUwd7F-qQ4Y4eD1JQ	3	Han Ba Tang	[Nightlife, Pubs, Lounges, Korean, Asian Fusio...	3.5	213	43.762928	-79.411511
6	dTuT_G3Zp79RZmnF3oxfiA	3	The Bier Markt	[Belgian, Nightlife, Bars, Gastropubs, Canadia...	3.0	197	43.647095	-79.373915
7	MhiBpIBNTCAm1Xd3WzRzjQ	3	Messini Authentic Gyros	[Mediterranean, Sandwiches, Greek, Restaurants...	3.5	372	43.677691	-79.350536
8	9_CGhHMz8698M9-PkVf0CQ	2	Little Coxwell Vietnamese & Thai Cuisine	[Vietnamese, Thai, Restaurants]	4.0	109	43.696175	-79.329092
9	ofw8aDSEg1HoQdmCgvLtaQ	2	The Pie Commission	[Canadian (New), Fast Food, Food, Do-It-Yourse...	4.5	183	43.623881	-79.512074

Friends Recommendations Notebook on GitHub

Hybrid Recommendation Engine

Burke (2002) introduced Strategies for the hybrid recommendation systems. He classified them into seven categories, weighted, switching, mixed, feature combination, feature augmentation, cascade, and meta-level:

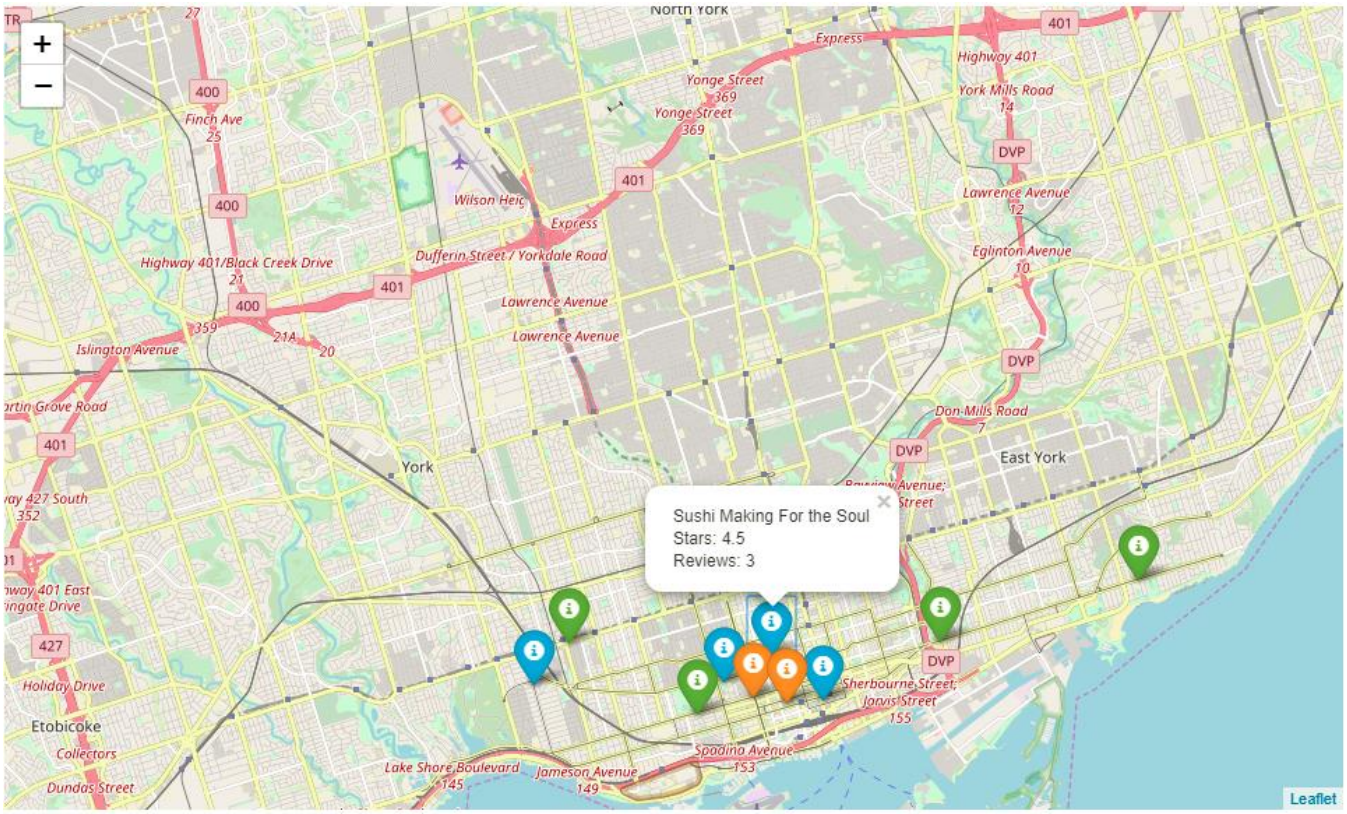
- Weighted: The score of different recommendation components are combined numerically.
- Switching: The system chooses among recommendation components and applies the selected one.
- Mixed: Recommendations from different recommenders are presented together.
- Feature Combination: Features derived from different knowledge sources are combined together and given to a single recommendation algorithm.
- Feature Augmentation: One recommendation technique is used to compute a feature or set of features, which is then part of the input to the next technique.
- Cascade: Recommenders are given strict priority, with the lower priority ones breaking ties in the scoring of the higher ones.

- Meta-level: One recommendation technique is applied and produces some sort of model, which is then the input used by the next technique.

In this project we chose to implement the mixed hybrid strategy. All recommendation types produced earlier have been included in a single recommendation engine that can generate:

1. Content-based Filtering Recommendations
2. Key Words Search Recommendations
3. Collaborative Filtering Recommendations
4. Friends Network Recommendations
5. Hybrid Recommendations (mix of 1, 3, and 4)

This engine has additional methods for supporting functions, like models training and loading; data loading and text processing/transformation.



[Hybrid Recommendation Engine Notebook on GitHub]

Summary and Next Steps

Recommendation engines now power most of the popular social and commerce websites. They provide tremendous value to the site's owners and to its users but also have some downsides. This project explored building a hybrid recommendation engine that overcomes the limitations of individual recommendation systems.

In content-based filtering, we built items attributes by vectorizing users review text, which were then compared to user's profile to get top-N restaurants similar to the user profile. We devised a dynamic user profile to derive these similarities but a proper user profile building method using weighted average of items attributes most liked by the user could enhance this type of recommendations.

The collaborative filtering recommendations were built using ALS algorithm of Apache Spark ML. The best score for RMSE obtained while training the ALS model was not optimal. A better score (less RMSE) could be reached if the same model is trained on a high end computing hardware that could deal with higher values for ALS *rank* and *maxIter* parameters during hyper-parameter tuning.

Moreover, recommendations systems performance evaluation methods and metrics should be investigated and introduced to assess individual recommender performance as well as evaluate and compare different hybrid recommendation strategies.