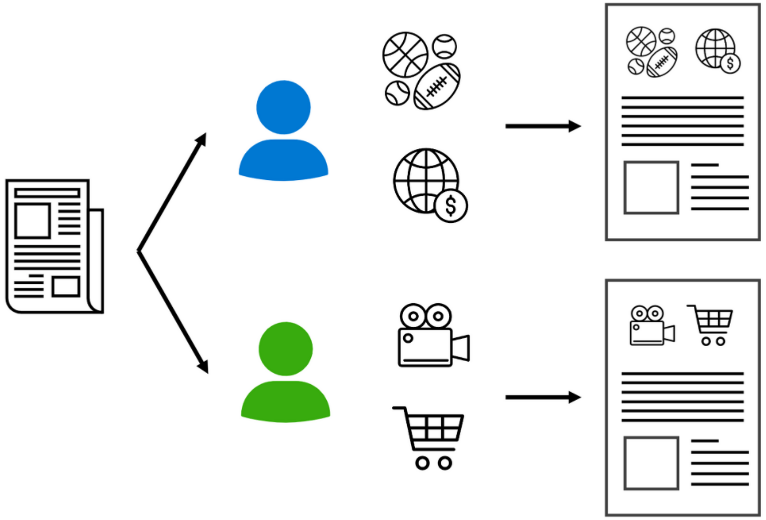
https://www.kaggle.com/fff9512/recomm-system-for-deli-app-cf-mf-tf-idf-d2v

Recommendation system for Delivery Application

(with CF, MF, TF-IDF, Doc2Vec)

Contents.

Recommendation System.



In the point of view by customers, there are too many goods and services. So paradoxically, customers have to make an effort more when choosing what they want. If customers feel fatigue, they will leave.

Recommendation system receive attention as solution for the problem.

Kinds of Recommendation System

- Segmentation 〉 Personalization
- Collaborative Filtering
- Contents Based Filtering
- Matrix Factorization
- Deep Learning
- Hybrid Filtering

Examples of Recomm. System

amazon

Amazon has recommended books
for each customers

NETFLIX

Netflix recommend video
which fit for each customers.

YouTube

Youtube recommend video
which fit for each customers.

01 Preface

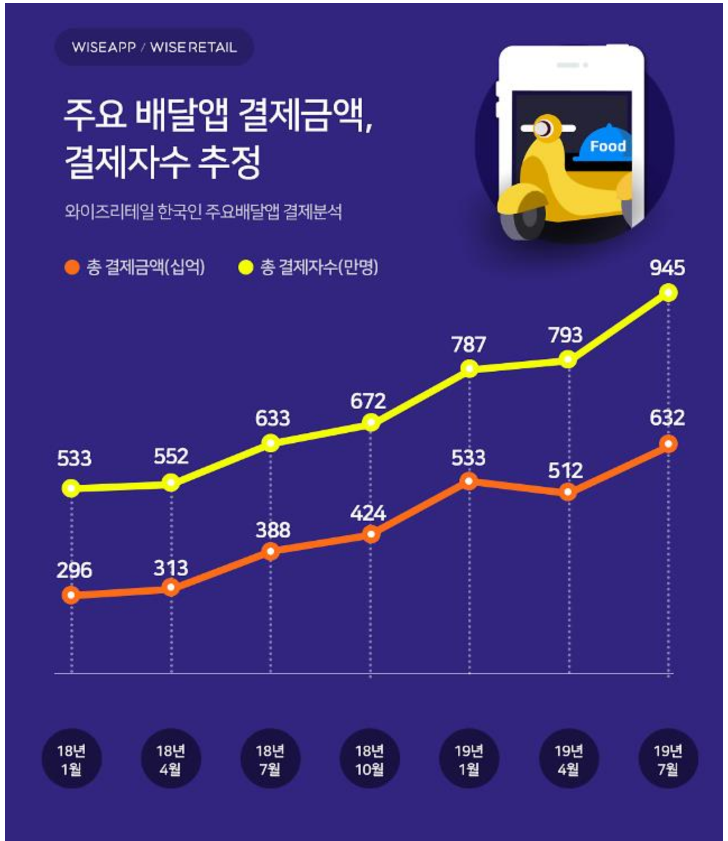
Backgrounds

Market of Delivery Service.

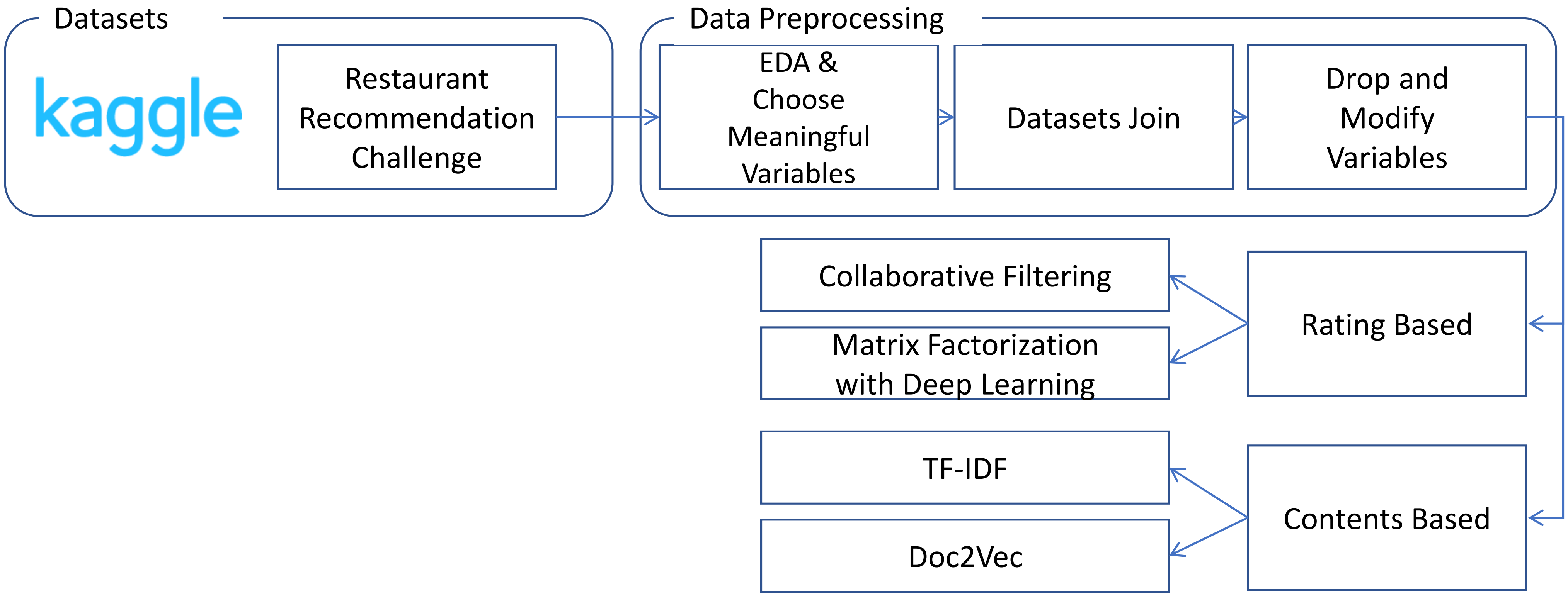
Because of increasing **single-person household and seeking convenience** by people,
delivery application expands rapidly in many countries.

Especially state of **COVID-19**, with necessity for avoiding contact with other people,
it becames more important.

(출처 : 아주경제)



Framework.

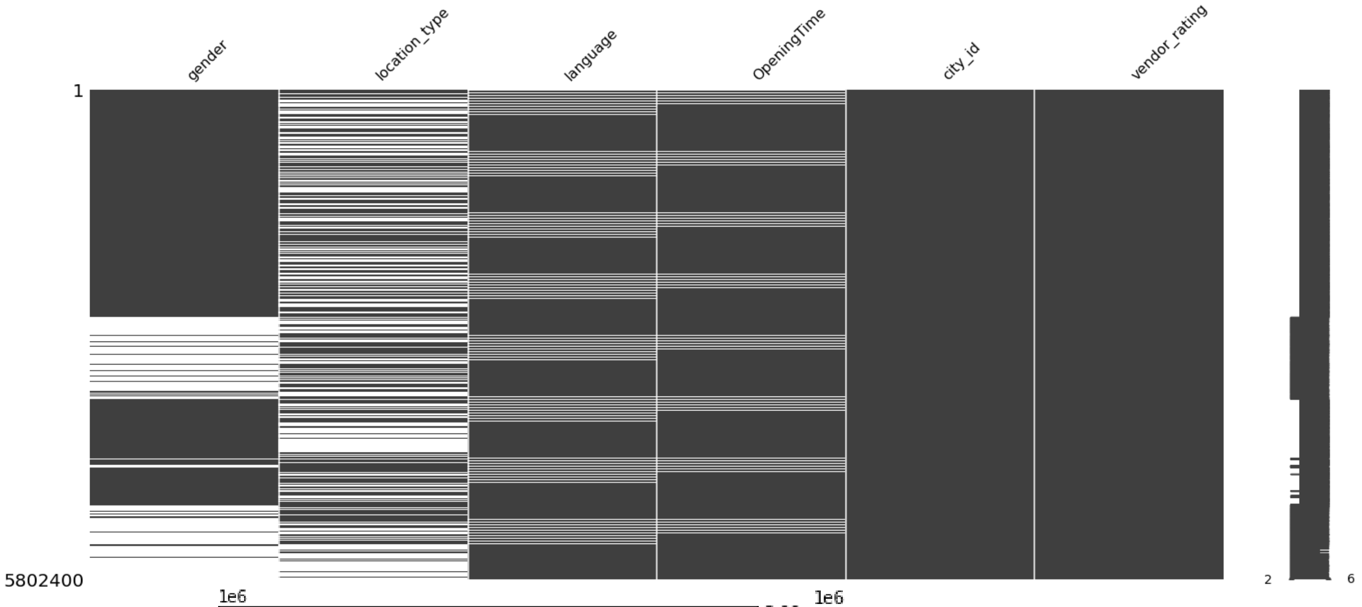

Datasets
kaggle
Restaurant Recommendation Challenge
Data Preprocessing
EDA & Choose Meaningful Variables
Datasets Join
Drop and Modify Variables
Collaborative Filtering
Matrix Factorization with Deep Learning
Rating Based
TF-IDF
Doc2Vec
Contents Based

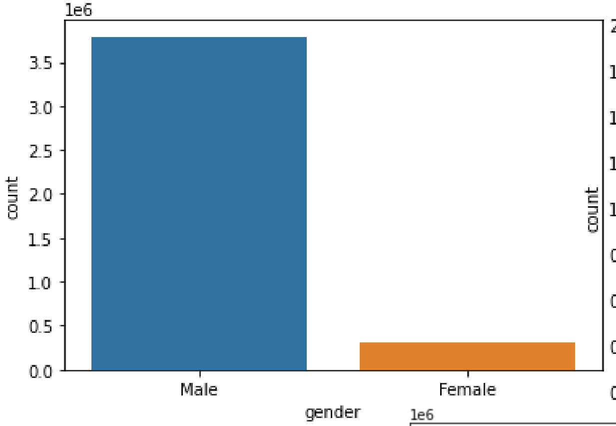
EDA.
customer datasets

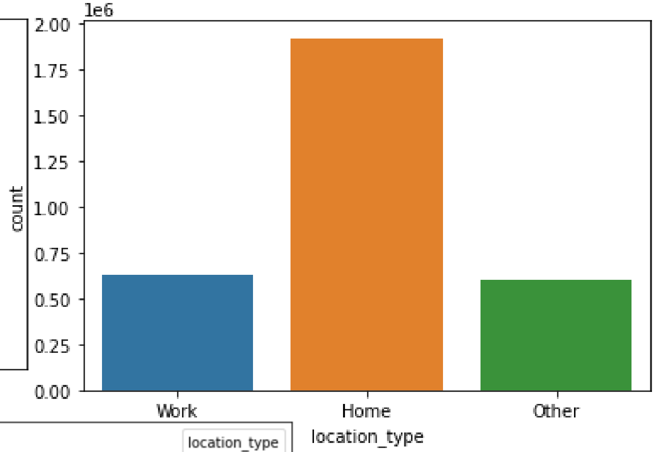
The dataset has 73 columns and 5802400 rows :

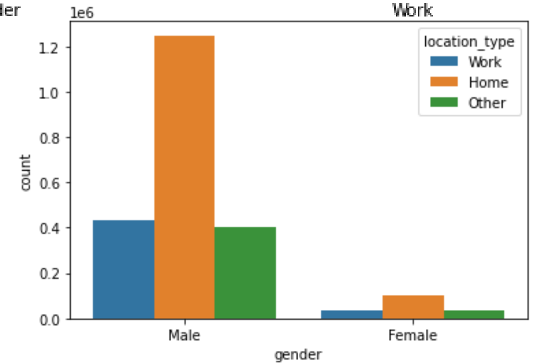
So we just look over about 8-10 columns

- gender : Customer's sex
- location type : Customer orders from one or more locations
- language : Chose language
- Opening Time : Vendr's operating time
- city_id : City's id
- vendor_rating : The vendor's average rating score









02 Datasets

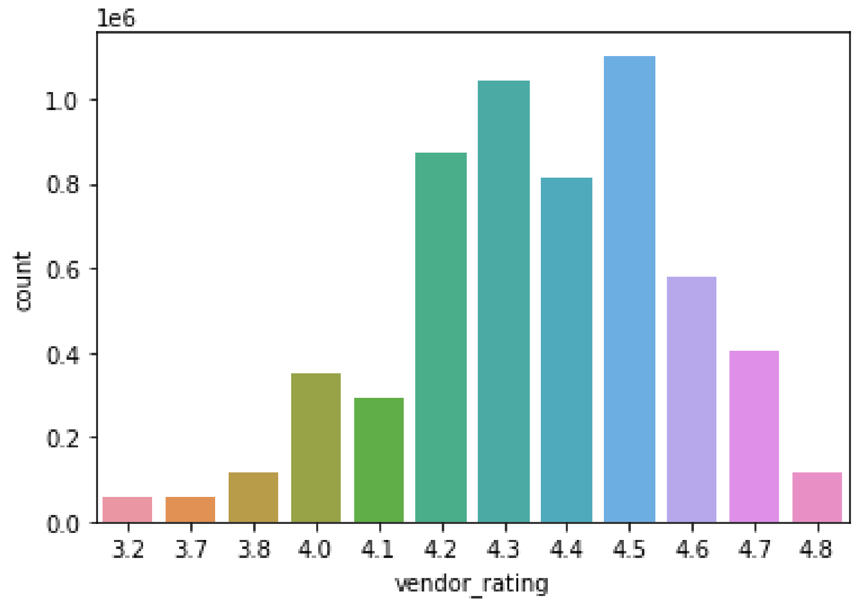
EDA

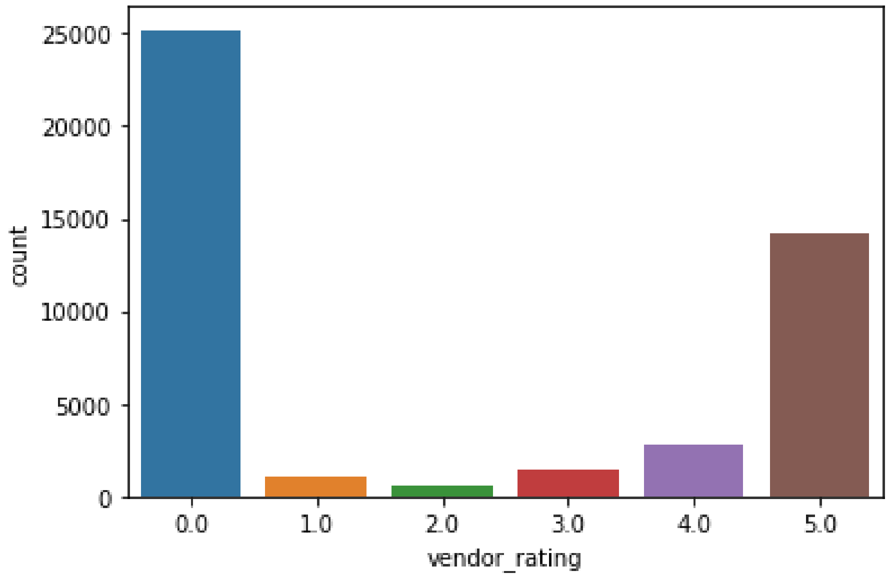
EDA.
vendor datasets

The dataset has 26 columns and 135303 rows :

Just simply explain some columns

- akeed_order_id : Unique customer ID, used in train_locations and train_orders
- vendor_rating : The ratings are rated by customers who use the vendor
- vendor_id : vendor's unique id
- customer_id : customer's unique id

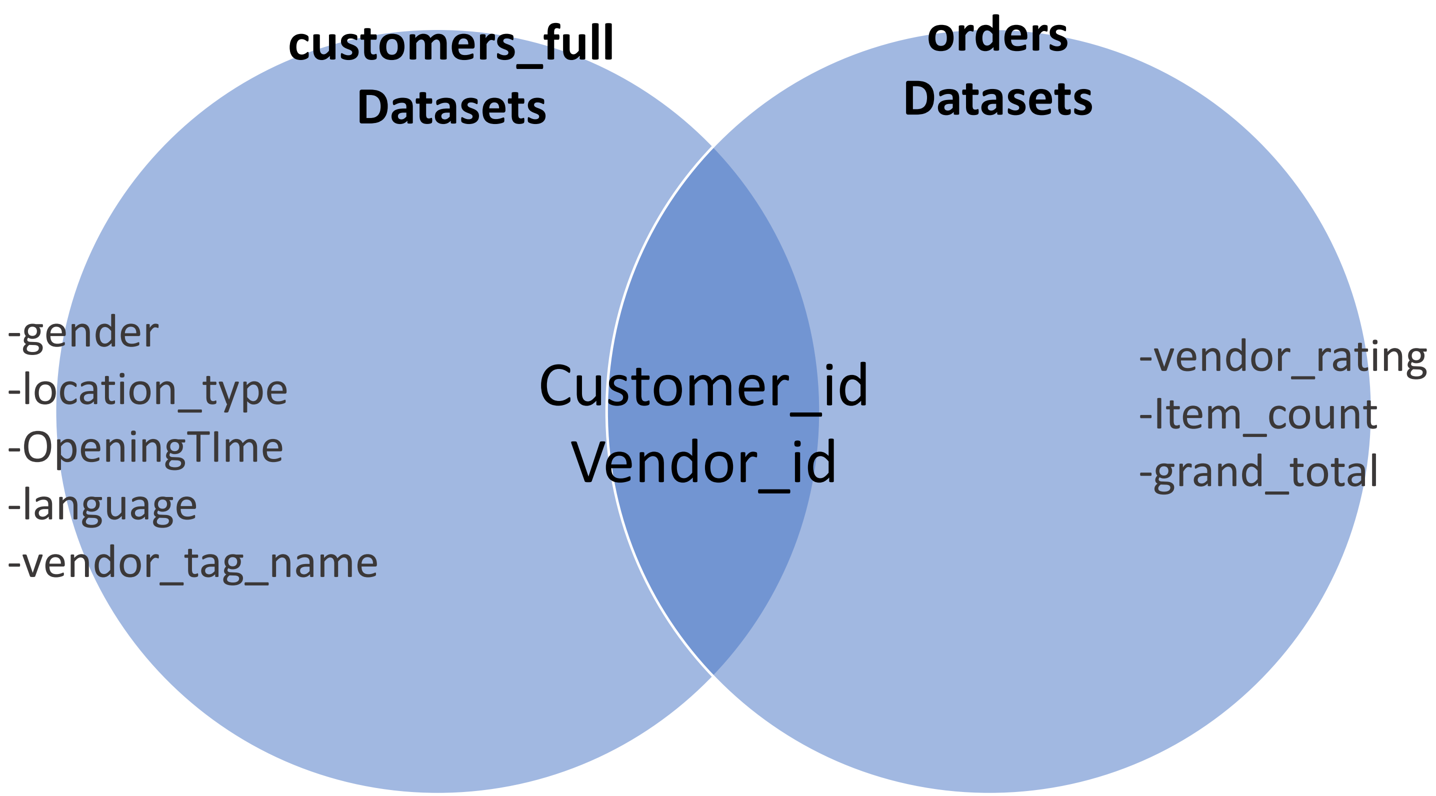




02 Datasets

Join Datasets

JOIN.



Preprocessing.

- Drop Null value
- Remove un-available variables
- Change type of variables

ex. int -> str / str -> list

- Make one-hot vectors
- Add morning/ afternoon/evening variables based OpeningTime
- Calculate similarity of words -> Change
- Make matrix

	customer_id	vendor_id	OpeningTime	vendor_tag_name
0	TCHWPBT	113	10:59AM-10:59PM	Arabic,Desserts,Free Delivery,Indian
1	TCHWPBT	237	08:30PM-11:59PM	American,Burgers,Desserts,Donuts,Fries,Pasta,S...
2	ZGFSYCZ	4	11:00AM-11:30PM	Arabic,Breakfast,Burgers,Desserts,Free Deliver...
3	ZGFSYCZ	28	11:00AM-11:45PM	Burgers
4	ZGFSYCZ	28	11:00AM-11:45PM	Burgers

OpeningTime	vendor_tag_name	Open	Close	afternoon	evening	vendor_tag	morning
10:59AM-10:59PM	arabic,desserts,free delivery,indian	10:59AM	10:59PM	1.0	1.0	[arabic, desserts, free delivery, indian]	1.0
08:30PM-11:59PM	american,burgers,desserts,donuts,fries,pasta,s...	08:30PM	11:59PM	0.0	1.0	[american, burgers, desserts, donuts, fries, p...	0.0
11:00AM-11:30PM	arabic,breakfast,burgers,desserts,free deliver...	11:00AM	11:30PM	1.0	1.0	[arabic, breakfast, burgers, desserts, free de...	0.0

Collaborative Filtering.

- Calculate mean rating by only valid ratings (except missing & rated zero)
- Substitute missing and zero rating to mean by valid ratings

- **Making Full Matrix** (Sparse Matrix)

- **Build CF Model**
- CF model (restrict number of neighbor size)
- Calculate accuracy (Root Mean Squared Error)

#RMSE
0.386589260498215 7

Example of recommendation list
(customer_id='ZZV76GY')

	vendor_id	mean_rating	vendor_tag_name
0	288	4.6	Asian,Desserts,Rice,Salads,Soups,Thai
1	577	4.5	Burgers,Desserts,Family Meal,Salads
2	92	4.6	Asian,Fresh Juices,Kids meal
3	92	4.6	Asian,Fresh Juices,Kids meal
4	676	4.4	Biryani,Desserts,Indian,Kebabs,Rice

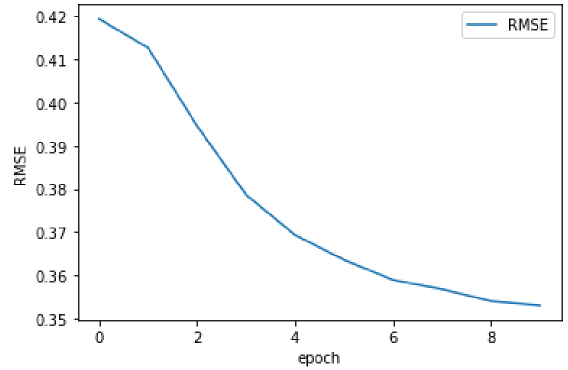
Matrix Factorization with Deep Learning.

- **Making Full Matrix with index**

For MF with DL, it is needed that Full Matrix composed with continuous numeric column name and row name

- **Build MF(with DL) Model**
- Embedding for Keras model
- Model compile
- Model fitting

#RMSE



Example of recommendation list
(customer_id='ZZV76GY')

	vendor_id	predicted_rating	mean_rating	vendor_tag_name
0	310	4.014	4.8	Bagels,Desserts,Salads
1	679	3.994	4.5	Biryani,Desserts,Indian,Keba
2	386	3.992	4.5	Churros
3	216	3.989	4.7	Coffee,Organic
4	298	3.984	4.7	Free Delivery,Fresh Juices,F

03 Model

Contents Based

TF-IDF.

- **TF-IDF**

Term Frequency – Inverse Document Frequency

TF means a value that how often certain words appear in a document. TF-IDF is multiply of TF and IDF.

- **Vectorizing word**

usingTfidfVectorizer

- **Calcurate word**

Using Cosine Similarity

- **Adding time tag**

(morning /afternoon/evening)

Example of Recommendation list (vendor_id='113')

```
get_recommendations('113')
```

```
[(36, 0.7180609886155626),
 (61, 0.5781243366476634),
 (30, 0.5015331801967973),
 (42, 0.4813164228848414),
 (8, 0.4220814531962946 4),
```

Example of Recommendation list **with time tag** (vendor_id='113')

```
item_recommendations('113')
```

```
[(36, 0.7646218746937348),
 (61, 0.60208312512282),
 (30, 0.5282753498785006),
 (42, 0.4795944045350 8845),
 (2, 0.4270039253402 4345),
 (8, 0.4154209006850713),
```

03 Model

Contents Based

Doc2Vec.

- **Doc2Vec**

Document Embedding with Paragraph Vectors

is an extended algorithm in Word2Vec which predicts word by sequential paragraph analysis.

- **Word2Vec**

Calculate similarity in Vendor_tag

- **Doc2Vec**

Calculate similarity among Document

- **Adding time tag**

(morning /afternoon/evening)

word_vectors_similar

```
word_vectors.most_similar('breakfast')
```

```
[('coffee', 0.8484305143356323),
 ('organic', 0.8430677652359009),
 ('sandwiches', 0.8271241188049316),
 ('vegetarian', 0.7660354971885681),
 ('seafood', 0.6889695525169373),
 ('shawarma', 0.6674776673316956),
 ('kebabs', 0.656210720539093),
 ('hotdogs', 0.6306989789009094),
 ('desserts', 0.5386044383049011),
 ('donuts', 0.536520779132843)]
```

Example of Recommendation (vendor_id='113')

```
get_recommendations_w2v('113')
```

```
[(52, 0.9846329),
 (78, 0.9652579),
 (40, 0.9541999),
 (81, 0.94047564),
 (7, 0.9133894),
```

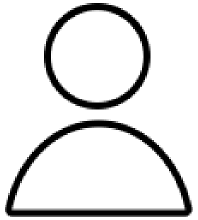
with time tag

```
get_recommendations_w2v_time('113')
```

```
[(52, 0.9869789),
 (29, 0.9603498),
 (40, 0.9457837),
 (81, 0.9347794),
 (19, 0.92800534),
```

Expected Effects.

Delivery app.



customers

restaurants

More transactions means more revenue.
Also, decrease bounce rate.

When customers access the list,
it is easy to choose
to eat and to order that.

it can be an accelerator to secure regular customers

Expandability.

Combination of two directions

Run the app.

Rating based models
recommend restaurants
by using information of each customer

	vendor_id	mean_rating
0	288	4.6
1	577	4.5
2	92	4.6
3	92	4.6
4	676	4.4

Select item.

If customers select certain items
(or add to cart),
show lists of similar items

```
get_recommendations_
```

```
[(52, 0.9869789),
 (29, 0.9603498),
 (40, 0.9457837),
 (81, 0.9347794),
 (19, 0.92800534),
```

Limitations of analysis

- For more accurate analysis, the model can have more various variables as a feature
- Post-Processing Algorithm is needed (considerate location, gender, distance)