[캡스톤디자인 결과보고서]

■ 연구과제

과 제 명 (작품명)	데이트 마이닝: 빅데이터 기반 서울 데이트코스 추천 시스템	참여학기	2020년 1학기

■ 강좌정보

과 목 명	데이터분석 캡스톤 디자인	학수번호	SWCON32100
과제기간	2020년 3월 20일 ~ 2020년 6월 27일	학 점	3

■ 팀구성

팀 명	Khupid		팀구성 총인원	3명
구 분	성명	학번	소속학과	학년
대표학생	유정수	2017100907	산업경영공학과	4
참여학생	김동혁	2015100905	산업경영공학과	4
	류연주	2015105665	관광학과	4

■ 지도교수 확인

지도교수	성 명	이대호	직 급	전임교수
	소속학과	소프트웨어융합학과	지도교수 확인	성명 : (인)

■ 붙임

[첨부1] 과제 요약보고서

[결과물] 최종결과물 (최종작품 사진/도면/발표자료 등)

본 팀은 과제를 성실히 수행하고 제반 의무를 이해하여 이에 따른 결과보고서를 제출합니다.

일자 : 2020년 6월 28일

신청자(또는 팀 대표) _____ (인)

[캡스톤디자인 과제 요약보고서]

과 제 명	데이트 마이닝: 빅데이터 기반 서울 데이트코스 추천 시스템

1. 과제 개요

가. 과제 선정 배경 및 필요성

빅데이터 기반의 추천 시스템은 오래 전부터 연구되어왔고, 유튜브, 넷플릭스, 왓챠 등 미디어 플랫폼과 인터넷 쇼핑몰에서 좋은 성능을 발휘했으나, 조사 결과 맛집이나 데이트 명소를 추천해주는 애플리케이션은 수요가 높음에도 불구하고 빅데이터 기반의 추천 알고리즘을 사용하지 않고 있었다.

요즘 많은 커플들이 데이트 코스를 짜는 것에 부담이나 어려움을 느끼는 경우가 많다. 여기에 두 사람의 취향이나 위치, 경비 등을 모두 고려하여 매번 다른 데이트 코스를 짜는 것은 더더욱 어렵다. 기존에도 데이트 코스를 추천하는 서비스는 있지만, 사용자의 취향을 반영하는 것이 아닌 에디터들이 직접 구성한 콘텐츠 밖에 없었다.

따라서 본 프로젝트를 통해 위치 기반 추천 시스템을 개발해 사용자의 취향에 맞는 장소를 높은 성능을 발휘하는 최근 트렌드 알고리즘을 적용해 추천해줄 예정이다. 이 알고리즘은 데이트 코스를 결정하는 데에 어려움을 겪었던 커플들에게 도움을 줄 것이며, 랜덤 데이트 코스라는 재미 요소도 줄 것이라고 예상한다.

나. 과제 주요내용

개인의 취향이나 성향에 적합한 데이트 코스를 추천해주는 것이 목표이다. 맛집 뿐만 아니라 카페, 영화 및 다양한 액티비티 장소가 모두 추천 대상에 포함된다. 단, 장소의 카테고리 별 추천 방향이 모두 다르기 때문에 각각에 대해 다른 알고리즘을 적용시키는 것이 중요할 것이다. 구체적인 추천 방향으로는 취향 기반/위치 기반/유사한 사용자 기반 등이 있다.

우리 프로젝트는 해당 장소에 대한 사용자의 리뷰 분석을 기반으로 각 장소와 사용자를 대표할 수 있는 형용사 클러스터를 만들어 이를 추천 알고리즘에 반영했다. 기존에 높은 성능을 발휘했던 알고리즘 중 하나인 NCF(Neural Collaborative Filtering) 기법과 함께 형용사를 이용한 Contents-based Filtering 기법을 적용해 추천 성능을 강화할 목적이다.

다. 최종결과물의 목표

1) 정성적 목표

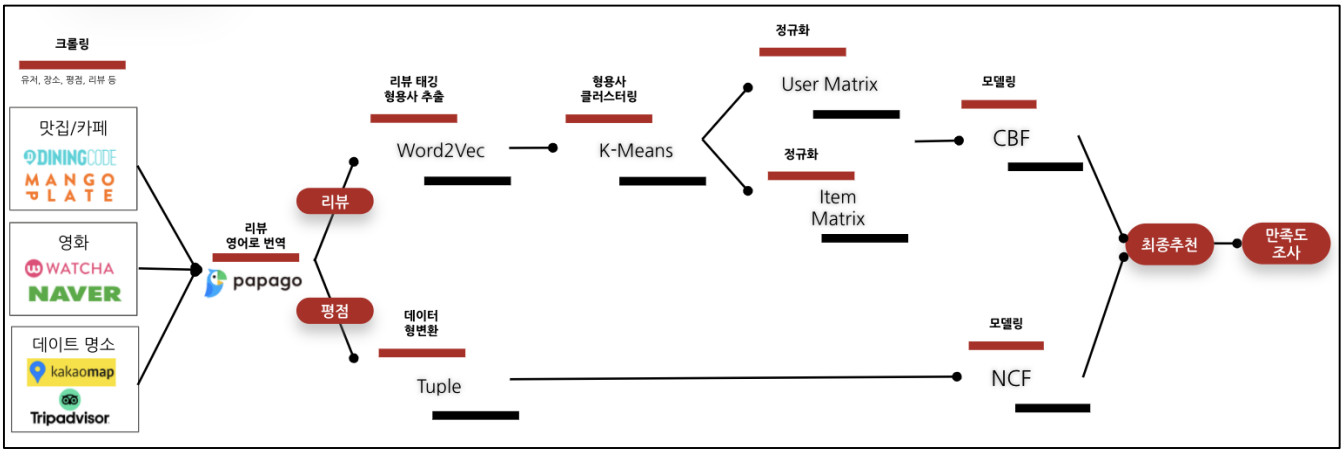
이 프로젝트의 최종 결과물은 데이트 취향과 위치를 고려한 데이트 코스 추천 알고리즘이며 데이트 코스에는 사용자의 취향에 맞춘 맛집/카페, 데이트 명소(유명 방문지, 보드게임 카페, 볼링장, 방탈출 카페 등등), 영화가 포함되어 있다. 또한 데이트를 하고자 하는 지역구를 선택한다면, 이에 대한 코스 필터링 기능을 제공하는 것 또한 최종 결과물에 포함되도록 한다. 기존의 목표는 웹앱 제작까지 포함하였으나, 시간상의 관계로 구글 폼으로 대체하여 유저로 하여금 웹앱에서 데이트 코스 추천을 처음 받을 때의 경험을 똑같이 할 수 있도록 문항들을 구성하도록 한다.

2)정량적 목표

정형화된 데이터 셋이 아닌 점과 선행 연구에 대한 오픈 소스 프로젝트가 없는 관계로, 선행 연구와 비교한 정량적 목표 설정은 무리가 있다는 점을 인지하였다. 따라서 표본을 모집 후 만족도를 조사하여 MOS 기법을 사용할 것이다. 각 구글 설문지를 통해 요구되는 몇 가지 문항들에 답변한 이후, 메일로 알고리즘이 추천한 데이트 코스 결과를 받고 이후 다시 메일에 첨부된 만족도 조사에 답하게 된다. 만족도 조사는 알고리즘이 추천한 코스에 대한 만족도를 묻는 총 1 문항으로, 1 점 매우 불만족한다 부터 시작하여 5 점은 매우 만족한다까지로 척도를 제시해 30 명 내외의 표본을 대상으로 실시하게 된다. 이를 통해 최종적으로 평점 3.5 점 이상(5 점 만점)을 받는 것을 목표로 한다.

2. 과제 수행방법

가. 과제수행 개요



[그림 1] 과제 수행 플로우 차트

나. 데이터 수집

1. 장소명이나 위치, 영업시간 등 기초정보를 같은 형식으로 통일되도록 기본적인 데이터 필드를 정의한다.
2. 팀 내에서 선정한 포털사이트나 플랫폼을 통해 갈 곳, 볼 것, 먹을 곳 등에 대한 정보를 파이썬 패키지인 selenium, BeautifulSoup 을 통해 크롤링한다.
3. 갈 곳(데이트 명소)의 경우 트립어드바이저와 카카오 맵 api 을 통해 관광지와 각종 콘텐츠를 즐길수 있는 장소들의 기본정보와 리뷰와 평점을 크롤링한다.
4. 영화의 경우 왓챠, 네이버 영화 등에서 영화에 대한 정보와 기본적인 정보, 리뷰, 평점 데이터를 크롤링한다
5. 먹을 곳(맛집/카페)의 경우 다이닝코드, 망고플레이트를 통해 맛집에 대한 정보와 그 곳의 사용자 리뷰, 평점들을 크롤링한다.
6. 각각 다른 플랫폼에서 크롤링을 하기때문에 데이터의 type 이나 형식이 다르므로, 데이터들의 형식을 통일하는 전처리 과정이 필요하다.
7. 최종적으로 프로젝트에 사용된 데이터 수는 다음과 같다.

	Users	Items	Ratings
EAT	22173	5397	101595
GO	20246	1260	41315
WATCH	14795	800	40232

[표 1] 데이터 수

다. 자연어 처리를 통한 형용사 클러스터링

1. 리뷰 데이터에서 추출한 형용사 태그들을 clustering 하기 위해선 word embedding 이 필요하다. 이 분야에서 가장 많이 이용되는 word2vec 을 포함해 Fast-Text, GloVe 등의 기법을 연구해보았고, BERT 와 ELMO 등 최신 NLP 알고리즘도 시도해보았다.
2. 각 알고리즘을 이용해 정성적으로 평가해보았을 때, word2vec 를 통해 word embedding 을 진행하는 것이 가장 높은 성능을 보였다.

	comfortable	only	favorite	little	weak	first	want	female	overall	other	delicious
0	-0.000428	0.001228	-0.000180	0.000972	-0.000185	0.000038	0.000038	0.000261	0.000047	-0.000214	-0.000271
1	0.000183	0.000173	0.000307	0.001191	0.000233	0.000596	0.000923	0.000445	-0.000409	0.000150	-0.000466
2	0.000478	-0.000791	0.000173	-0.000499	0.001446	0.001369	0.000924	-0.000424	-0.000552	-0.000090	-0.000167
3	-0.000368	-0.001272	0.000377	0.000133	0.001009	0.000273	-0.000876	0.000082	-0.000340	0.000455	-0.000392
4	-0.000772	-0.000109	0.000543	-0.000014	0.000653	-0.000085	-0.000363	0.000610	-0.000167	-0.001110	-0.000231
...	...	...	...	...	...	...	...	...	...	...	...
295	0.000125	0.000061	-0.000115	0.000619	-0.001195	-0.001046	0.000390	0.000690	0.000301	0.000077	0.000119
296	-0.000071	0.000744	-0.000208	-0.000682	0.000181	0.000426	-0.000330	0.000058	-0.000014	0.000821	-0.000109
297	-0.000125	-0.000324	-0.000019	-0.000676	0.000222	0.000235	-0.000293	0.000132	-0.000472	-0.000355	0.000125
298	-0.000319	0.000006	-0.000451	0.000197	-0.000604	-0.000112	-0.000614	0.000107	-0.000610	0.000303	-0.000287
299	0.000050	-0.000063	-0.000577	0.000161	-0.000171	0.000441	0.000525	0.000271	0.000296	0.000548	0.000539

[그림 2] word embedding

3. word embedding 을 거쳐 다음과 같은 알고리즘 등을 통해 clustering 해보았다.

① K-means clustering
② K-medoids clustering
③ DBSCAN(Density-Based Spatial Clustering of Applications with Noise)
④ GMM(Gaussian Mixture Models)
⑤ Affinity Propagation
⑥ OPTICS(Ordering points to identify the clustering structure
⑦ BIRCH(Balanced Iterative Reducing and Clustering using Hierarchies)
⑧ Spectral

4. cluster 별 단어 분포 및 각 cluster 에서 단어 유사도를 파악해본 결과 k-means clustering 이 가장 높은 성능을 보였다.

0	1	2	3	4	5	6	7	8	9
cave	available	korea	kindness	hateful	sour	traditional	hidden	reasonable	atmosphere
ancient	uninvited	chinese	respectful	inflammatory	bitter	orthodox	invincible	sincere	busy
medieval	designated	indian	respect	NaN	acrimonious	conventional	silent	satisfactory	noisy
prehistoric	anonymous	asian	appreciate	NaN	NaN	unconventional	invisible	proper	calm
olden	unmanaged	african	gratitude	NaN	NaN	customary	forgotten	transparent	loud
paleolithic	mandatory	asia	praise	NaN	NaN	nonmainstream	unseen	doable	messy
primitive	specific	japan	NaN	NaN	NaN	mainstream	exist	fair	hectic
neolithic	optional	india	NaN	NaN	NaN	nontraditional	isolated	sufficient	chaotic
dinosaurs	preferred	mexican	NaN	NaN	NaN	NaN	unrecognized	appropriate	orderly
dinosaur	minimum	america	NaN	NaN	NaN	NaN	unaltered	rational	tangled
herbivorous	forbidden	africa	NaN	NaN	NaN	NaN	extinct	realistic	congested
NaN	onsite	australia	NaN	NaN	NaN	NaN	bankrupt	honest	disorderly

[그림 3] word clustering

라. 장소/유저별 태그 점수 부여, 매트릭스 생성

1. 위에서 제작한 태그사전을 기반으로 user/place 에 해당하는 태그가 각 cluster 마다 얼마나 속해있는지를 count 해 점수를 부여하고, matrix 를 만든다. 그 결과 매트릭스는 다음과 같이 만들어진다.

	group_5	group_0	group_1	group_2	group_3	group_4	group_6	group_7	group_8	group_9	group_10	group_11
0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	2	0	2
3	2	4	15	1	0	0	5	13	0	21	0	3
4	0	0	1	0	0	0	0	0	3	1	0	0
5	1	0	3	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0	1	1	1	0	0	1	3	0	1	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	1	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0

[그림 4] user 별 리뷰 word count 결과

2. 그렇게 만들어진 matrix 는 각 user 가 얼마나 많은 리뷰를 남겼는지, 각 장소마다 리뷰가 얼마나 있는지에 따라 편향된 점수가 부여되므로, 이를 정규화해야한다. 정규화는 (해당 셀 값/해당 cluster 에 속한 word 수)를 각 row 에 대한 백분율로 환산하여 진행했다.

u_id	0	1	2	3	4	5	6	7	8	9	...	110	111	112	113	114
5351	0.025431	0.0	0.0	0.000000	0.027046	0.00000	0.036643	0.0	0.031998	0.020653	...	0.000000	0.058252	0.000000	0.000000	0.0
3024	0.017392	0.0	0.0	0.000000	0.004624	0.00266	0.021927	0.0	0.005471	0.007062	...	0.006697	0.004980	0.026483	0.000000	0.0
5941	0.015396	0.0	0.0	0.003558	0.010164	0.00000	0.048199	0.0	0.008017	0.005175	...	0.004907	0.014595	0.000000	0.001655	0.0
13580	0.019210	0.0	0.0	0.000000	0.000000	0.00000	0.083038	0.0	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.059865	0.0
6953	0.000000	0.0	0.0	0.000000	0.000000	0.00000	0.000000	0.0	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
18726	0.000000	0.0	0.0	0.000000	0.000000	0.00000	0.114951	0.0	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.0
10858	0.078430	0.0	0.0	0.000000	0.000000	0.00000	0.000000	0.0	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.0
10956	0.055347	0.0	0.0	0.000000	0.000000	0.00000	0.000000	0.0	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.0
21694	0.022082	0.0	0.0	0.000000	0.000000	0.00000	0.000000	0.0	0.000000	0.000000	...	0.000000	0.151741	0.000000	0.000000	0.0
18178	0.000000	0.0	0.0	0.000000	0.000000	0.00000	0.000000	0.0	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.0

p_id	0	1	2	3	4	5	6	7	8	9	...	110	111	112	113	114
721	0.000000	0.000000	0.000000	0.00000	0.0	0.000000	0.000000	0.0	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.0
1832	0.053308	0.000000	0.000000	0.00000	0.0	0.000000	0.046086	0.0	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.0
741	0.034111	0.000000	0.000000	0.00000	0.0	0.000000	0.000000	0.0	0.000000	0.055405	...	0.000000	0.078135	0.000000	0.000000	0.0
218	0.042261	0.000000	0.000000	0.00000	0.0	0.077576	0.182678	0.0	0.079761	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.0
3276	0.000000	0.000000	0.000000	0.00000	0.0	0.000000	0.000000	0.0	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3793	0.003882	0.000000	0.000000	0.00000	0.0	0.000000	0.000000	0.0	0.000000	0.018916	...	0.179379	0.026677	0.000000	0.000000	0.0
2809	0.016743	0.000000	0.000000	0.00000	0.0	0.006147	0.043423	0.0	0.018959	0.024475	...	0.000000	0.080536	0.000000	0.010435	0.0
3297	0.016136	0.002938	0.001383	0.00235	0.0	0.006439	0.021229	0.0	0.007945	0.010256	...	0.003242	0.043390	0.002137	0.006559	0.0
445	0.000000	0.000000	0.000000	0.00000	0.0	0.000000	0.039573	0.0	0.000000	0.000000	...	0.000000	0.000000	0.055763	0.000000	0.0
3592	0.000000	0.000000	0.000000	0.00000	0.0	0.000000	0.060476	0.0	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.0

[그림 5] 정규화한 user matrix 와 item matrix

마. Contents Based Filtering 모델 적용

1. 위에서 생성한 matrix 를 통해 각 user row 와 각 item row 를 vector 로 보고 두 vector 의 유사도를 구하는 방식으로 CBF score 를 도출했다. 여기서 vector size 는 각 분야별 word cluster 의 개수 이다.
2. 유사도를 구하는 방법으로는 가장 많이 쓰이는 cosine similarity 를 이용했다.

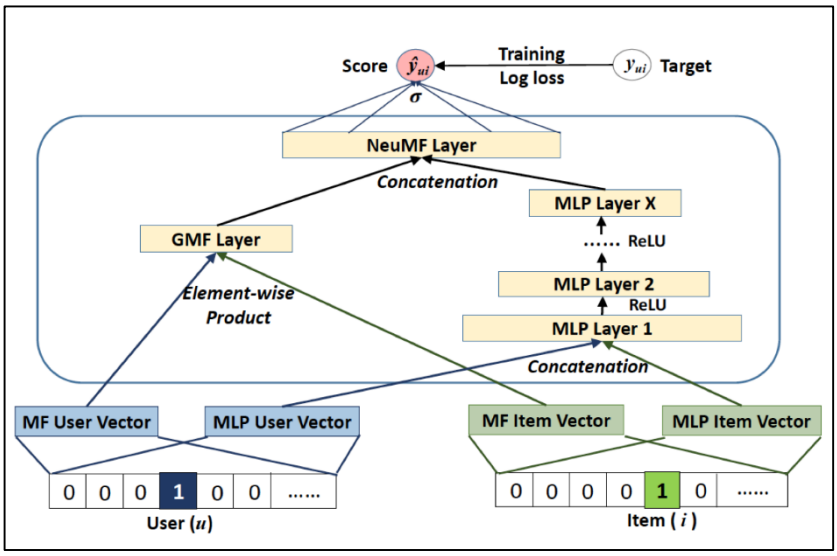
u1s2&쥬	0.000000	0.000000	0.003287	0.000000	0.034548	0.000000	0.000000	0.000000	0.0	0.000000

p_id	0	1	2	3	4	5	6	7	8	9	...	110	111	112	113	114
721	0.000000	0.000000	0.000000	0.00000	0.0	0.000000	0.000000	0.0	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.0
1832	0.053308	0.000000	0.000000	0.00000	0.0	0.000000	0.046086	0.0	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.0
741	0.034111	0.000000	0.000000	0.00000	0.0	0.000000	0.000000	0.0	0.000000	0.055405	...	0.000000	0.078135	0.000000	0.000000	0.0
218	0.042261	0.000000	0.000000	0.00000	0.0	0.077576	0.182678	0.0	0.079761	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.0
3276	0.000000	0.000000	0.000000	0.00000	0.0	0.000000	0.000000	0.0	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3793	0.003882	0.000000	0.000000	0.00000	0.0	0.000000	0.000000	0.0	0.000000	0.018916	...	0.179379	0.026677	0.000000	0.000000	0.0
2809	0.016743	0.000000	0.000000	0.00000	0.0	0.006147	0.043423	0.0	0.018959	0.024475	...	0.000000	0.080536	0.000000	0.010435	0.0
3297	0.016136	0.002938	0.001383	0.00235	0.0	0.006439	0.021229	0.0	0.007945	0.010256	...	0.003242	0.043390	0.002137	0.006559	0.0
445	0.000000	0.000000	0.000000	0.00000	0.0	0.000000	0.039573	0.0	0.000000	0.000000	...	0.000000	0.000000	0.055763	0.000000	0.0
3592	0.000000	0.000000	0.000000	0.00000	0.0	0.000000	0.060478	0.0	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.0

[그림 6] user vector 와 item vector

4. 그 결과 해당 user 의 item 별 score 를 도로 저장한 뒤 추후 모델 결과값 결합에 사용한다.

바. Neural Collaborative Filtering 모델 적용



[그림 7] Neural Collaborative Filtering model

(출처: Xiangana He 외 5 인.「Neural Collaborative Filtering」. 2017)

1. 크롤링했던 유저들의 평점 데이터를 이용해 Neural Collaborative Filtering(NCF) 모델로 결과값을 예측한다. NCF 는 기존의 collaborative filtering 기법과는 달리 DNN 적용을 통해 implicit feedback 을 사용하기 위함으로, 크게 'GMF(Generalized Matrix Factorization)'과 'MLP(Multi-Layer Perceptron)'파트로 나뉘어져 있다.
2. NCF 모델은 PreferredAI 에서 개발한 corncac 패키지를 이용해 구현한다. 패키지를 사용할 시 모델에 넣는 input 은 [(userID, ItemID, rating), ...]의 리스트이다.

```
[('0', 3375, 3.0),
 ('0', 2778, 5.0),
 ('0', 612, 5.0),
 ('0', 3949, 5.0),
 ('0', 1679, 1.0),
 ('0', 4805, 5.0),
 ('0', 287, 3.0),
 ('0', 3601, 5.0),
 ('0', 3526, 5.0),
 ('0', 3601, 5.0),
 ('0', 3357, 5.0),
```

[그림 8]input 예시

3. input 을 넣으면 모델은 input size 에 따라 해당 user 의 index 를 나타내는 User vector 와 해당 유저가 실제로 interact 한 item 을 나타낸 Item vector 로 변환한다.
4. 3)에서 생성한 두 벡터를 GMF, MLP 모델에 적용하기 위해 각각의 size 로 embedding 한다. 각 embedding size 는 모델 parameter 인 num_factors 와 layers 를 통해 결정된다. 우리 프로젝트에서는 layer=[64, 32, 16, 8]이므로 MLP 의 input layer 는 user vector 와 item vector 를 각각 size 32(layer[0]/2)로 embedding 한 뒤 concatenate 한 layer 이다.
5. MLP 의 활성함수는 'tanh'이며 output layer 에서는 'sigmoid'를 이용한다.
6. GMF 와 MLP 의 output 을 concatenate 한 최종 layer 를 통해 최종 output 을 예측하며, 'log loss'를 이용해 오차를 계산하며, 'adam' optimizer 를 통해 학습한다.
7. 위 모델에서 나온 결과값은 해당 user 의 index 를 통해 도출할 수 있으며 다음과 같다.

	u_id	p_id	ncf_score
0	dhyeok1996	53.0	2.205372e-06
1	dhyeok1996	1200.0	1.928508e-04
2	dhyeok1996	130.0	5.960464e-08
3	dhyeok1996	22.0	1.336336e-04
4	dhyeok1996	1341.0	5.662441e-07
5	dhyeok1996	906.0	2.782643e-04
6	dhyeok1996	2519.0	1.353025e-05
7	dhyeok1996	1105.0	4.139245e-04
8	dhyeok1996	485.0	0.000000e+00
9	dhyeok1996	525.0	3.571719e-03
10	dhyeok1996	1037.0	2.086163e-07
11	dhyeok1996	675.0	2.920628e-06
12	dhyeok1996	531.0	7.679164e-04

[그림 9]output 예시

8. NCF 모델의 hyper parameter 는 Random Search 를 이용해 구했고, 분야별 최적의 파라미터는 다음과 같았다.

	batch size	learning rate	number of epochs	number of factors
EAT	512	0.0057864483	50	8
GO	512	0.0022595564	50	4
WATCH	128	0.0075019904	100	8

사. 모델 결과값 결합

1. CBF 모델과 NCF 모델의 결과값을 결합하는 방식으로 최종 prediction score 를 도출한다. 이때 최종 score 는 두 모델의 score 를 단순 곱하는 식으로 사용했다.
2. 이때 결합 방식을 산술평균, 조화평균으로도 생각해보았지만 이는 다른 한 모델이 낮게 예측한 item 을 추천해주는 상황을 야기했다. 따라서 단순 곱을 통해 두 모델이 모두 높은 점수를 낸 item 을 추천하도록 했다.

CBF 모델 결과

p_id	cbf_score
643	0.936923
962	0.935923
2481	0.928743
1210	0.926539
3554	0.926400
...	...

X

NCF 모델 결과

p_id	ncf_scores
2210	0.013282
1944	0.000021
420	0.015204
2900	0.029513
4571	0.161402
...	...

=

p_id	name	address	result
4142	중앙해장	서울특별시 강남구 영동대로86길 17 육인빌딩	0.532276
1068	동경산책	서울특별시 성북구 보문로34길 45	0.505191
1712	용탄	서울특별시 용산구 백범로99길 50	0.503543
3468	온달 왕 돈까스	서울시 성북구 동선동1가 2-9	0.491465
1658	모리돈부리	서울특별시 관악구 관악로12길 6	0.458265
2704	쉐이크쉑 두타점	서울특별시 중구 을지로6가 18-12	0.454874
4412	카레시오	서울특별시 관악구 관악로14길 28	0.453978
1587	메이크샐러드	서울특별시 동대문구 휘경로2길 16 1F	0.448346
747	뉴질랜드스토리	서울특별시 송파구 송파동 32-1	0.434375
4772	토끼정	서울특별시 강서구 하늘길 38 롯데몰 김포공항점 MF	0.433607

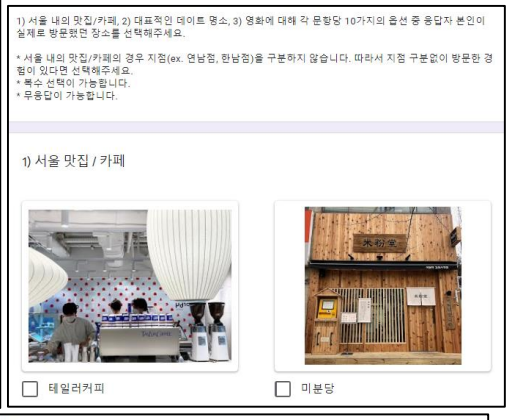
[그림 10] 모델 결과 결합

아. 지역구 필터링 적용

1. 위에서 도출한 결과 중 'EAT'과 'GO'에서 사용자가 희망하는 지역구에 속한 item 들을 조회한 뒤, 두 분야의 score 곱을 가장 크게 만드는 지역구로 최종 추천을 해준다.
2. 'WATCH'는 지역 영화관 현황과 해당 영화관의 상영작, 시간대에 따라 경우의 수가 많으므로 Naver 영화 정보를 기준으로 현재 상영 중인 영화 중 한가지로 추천을 해준다.
3. 사용자의 추천 희망 지역구가 단일 지역이면 해당 지역에서만 추천을 해주며. 2 개 이상의 복수지역이라면 score 합이 높은 상위 2 개의 지역구를 대상으로 추천을 해준다.

자. 만족도 평가

1. 본 프로젝트의 원래 계획으로는 앱 서비스 개발을 완료해 표본들에게 사용하게한 뒤 만족도 조사를 시행하려고 했으나, 시간관계상 앱 서비스를 직접 개발하지는 못했고, 구글 폼을 이용해 사용자의 데이터를 수집한 뒤, 우리 모델에 적용해 나온 결과값을 보내주고, 그에 대한 추천 만족도를 조사하기로 했다.
2. 설문의 내용은 크게 '선호하는 형용사 단어'와 '방문/관람해본 장소/영화'에 대한 내용이다. 보기에 주어진 형용사는 총 34 개이며, 각 장소/영화는 20 개이다.



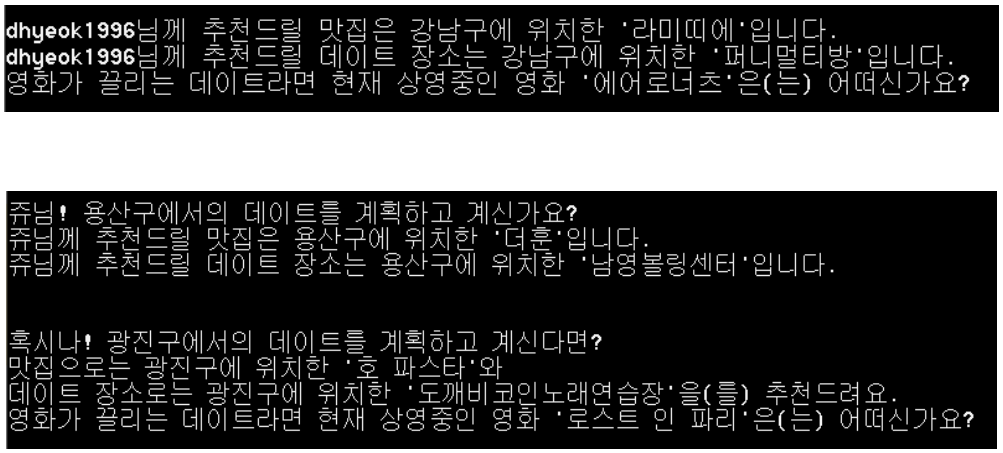
[그림 11] 설문 조사 내용

3. 각 형용사에 대한 선호도는 해당 유저의 word cluster 별 선호 점수 데이터를 입력하기 위해 수집했으며, 각 장소/영화에 대한 방문/관람 여부는 NCF 모델의 input 으로 들어갈 데이터를 입력하기 위해 수집했다.

3. 수행결과

가. 과제수행 결과

1. 정성적 결과



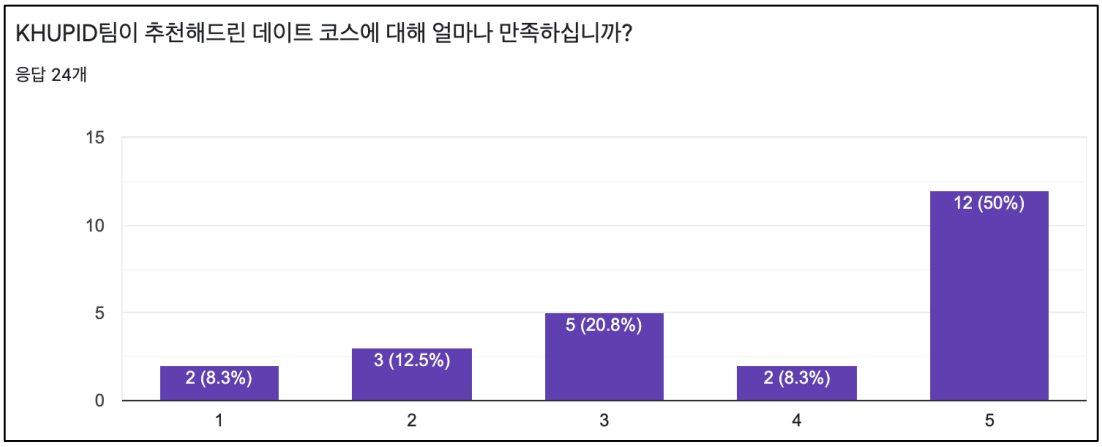
[그림 12] 알고리즘 실행 결과

로컬에서 돌린 위와 같은 결과를 앞서 설문을 통해 받은 테스터들의 이메일로 각각 발송하였으며, 이메일 내용에는 단순히 추천 결과 텍스트뿐만 아니라 데이트 코스에 따른 구글 맵 경로를 첨부하였다. 또한 최종적으로 만족도 조사가 가능하도록 구글 설문 링크를 첨부하여, 정량적 결과인 MOS 평가가 가능하도록 데이터를 수집하였다.

[그림 13] 결과 전송 이메일 내용

2. 정량적 결과

a) MOS 평가방법 결과



[그림 14] 최종 만족도 결과 그래프

본 프로젝트에서는 앞서 제시하였듯이 MOS 평가 방법을 사용해 5점 만점에 3.5점 이상의 만족도를 목표로 제시하였다. 알고리즘을 개발하고 30명의 테스트 대상자를 모집하여 데이트 모집 결과를 추천해준 결과, 24명의 응답 회신자들로부터 만족도 최종 평균 약 3.8점의 결과를 얻음으로써 프로젝트 처음에 제시한 목표를 충족시킬 수 있었다. 점수별 빈도는 림그림 *에서와 같이 1점 2명, 2점 3명, 3점 5명, 4점 2명, 5점 12명으로 추천한 코스에 대해 보통 이상으로 만족한 테스터가 약 79%이상이라는 점에서 의미있는 결과가 도출되었다.

b) 모델 최종 성능 결과

프로젝트에서 Random Search를 사용해 나온 최적 parameter를 사용해 나온 NCF 모델의 성능은 다음과 같다.

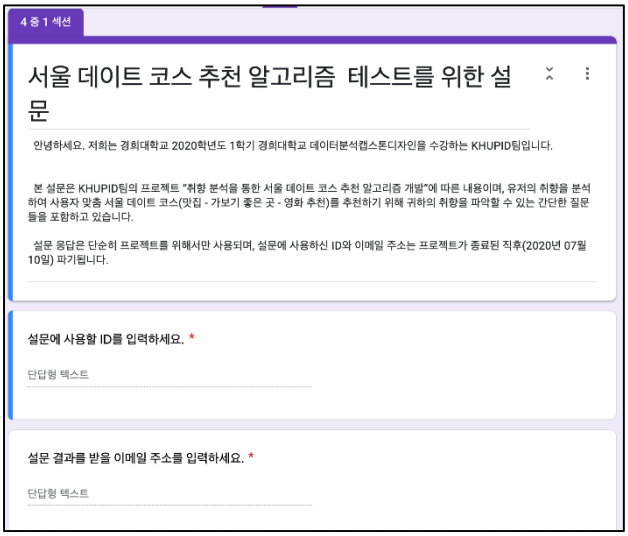
	F1@10	NDCG@10	Precision@10	Recall@10	Train(s)	Test(s)
EAT	0.0149	0.0313	0.0097	0.0540	14024.7585	11.2788
GO	0.0970	0.2482	0.0556	0.4532	3795.3764	2.3491
WATCH	0.0150	0.0283	0.0098	0.0535	4831.9023	1.0951

[표 2] 분야별 NCF 모델 성능 지표

참고한 논문에서는 NDCG@10가 평균적으로 약 0.43 정도였으며 마이크로소프트가 NCF 논문을 재구성한 라이브러리 예시로 제시한 결과 값에서는 NDCG@10가 0.198938였다는 점을 고려했을때 'GO'의 경우 논문보다는 낮고 라이브러리 예시보다는 높은 점수를, EAT과 WATCH는 전체적으로 모델 자체의 성능이 낮았다는 아쉬움이 남는다. 위와 같은 결과에 대한 원인으로 EAT과 WATCH에서는 리뷰의 갯수가 적은 유저가 많기 때문인 것으로 예상하였다. 이와 같은 한계점은 직접 플랫폼을 운영하는 서비스가 구축되어 데이터 문제가 해결된다면, 어느정도 극복할 수 있을 것이라 예상된다.

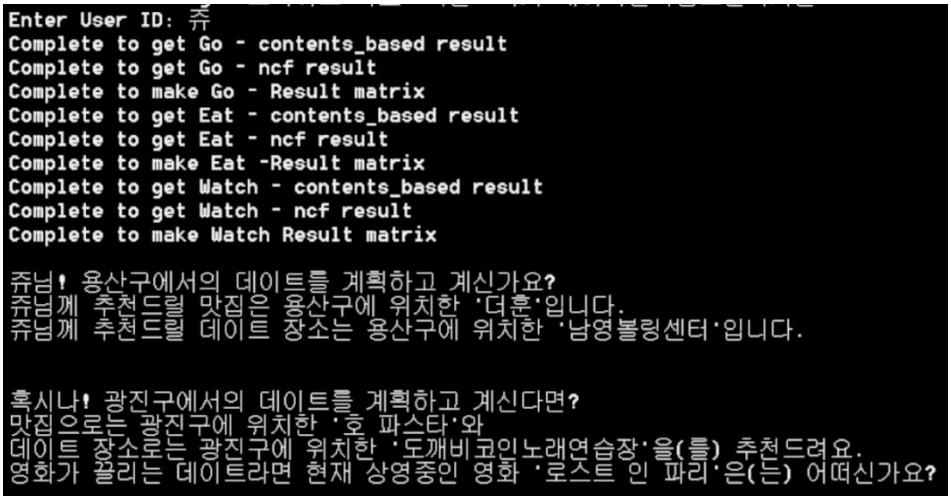
나. 최종결과물 주요특징 및 설명

최종 결과물은 크게 웹앱을 대체할 구글 설문지와 알고리즘 코드, 그리고 알고리즘에 사용되는 모델들로 구성된다. 가장 먼저, 테스터들이 서비스를 처음 이용하는 유저라고 가정하고 구글 설문지 문항을 구성하였다.



[그림 15] 만족도 조사 설문

구글 설문지에서는 가장 먼저 유저의 기본정보인 아이디와 이메일을 묻게 된다. 이는 실제로 추후 알고리즘 아웃풋 결과를 제시할 때 테스트로 하여금 실제 서비스를 이용하는 듯한 경험을 주기 위해 구성한 질문이며, 이메일은 테스트 결과를 수신하고자 하는 목적으로 추가되었다. 이후 질문 묻는 문항들은 2-자-2) 에 기재한 문항들이며, 이 설문지를 통해 유저의 기본적인 취향 정보 데이터를 얻을 수 있다. 이는 이후 user matrix와 ncf model training 을 위해 사용된다



[그림] 알고리즘 input 과 output 예시

두번째로 알고리즘 코드에서는 유저의 ID를 입력 받고 서울 내 유저가 원하는 지역구의 데이트 코스를 출력한다.이때 유저가 다양한 지역구를 선택했다면 최종 result가 가장 커지는 두 행정구를 선택해 각 행정구마다 하나씩 총 두개의 데이트 코스를 출력하게 된다.

마지막으로 알고리즘을 구성하는 모델은 앞선 순서에서 제시한 바와 같이 Contents based filtering과 Neural Collaborative filtering인데, 본 프로젝트에서는 두 필터링 모델을 모두 사용한 하이브리드 모델을 도입하여 사용했다는 것을 시사점으로 꼽을 수 있다.

4. 기대효과 및 활용방안

가. 기대효과

기존의 카페나 블로그, SNS 등 포털사이트에서 제공하는 데이트 정보에는 대체적으로 부정확한 정보를 포함하고 있거나, 광고/홍보성이 짙은 게시물이 많다. 또한 최근 트렌드는 소수의 파워블로거나 에디터들의 영향이 크게 미치는 경우가 많다.

우리가 제작한 데이터 기반의 데이트 코스 추천 알고리즘은 어느 한 사용자의 의견에 편향될 가능성이 적으면서도, 사용자 취향에 맞는 아이템을 추천해줄 수 있다. 이는 사용자 개개인에게 적합한 결과를 내줄 뿐 아니라, 비슷한 사용자를 기반으로 기존에 경험했던 아이템이 아닌 새로운 아이템 또한 추천해줄 수 있다. 따라서 데이트 코스를 얻고자 하는 사용자들에게 흥미로운 추천 결과를 제시해줄 수 있을 것이다.

나. 활용방안

고객의 범주를 확대하는 측면에서, 데이트를 계획하고 있는 커플들뿐만 아니라 가족이나 친구들과의 여행, 외국인 관광객들에게도 주요 여행 코스를 추천해줄 수 있을 것이다. 추천 대상의 범주를 확대하는 측면에서는 병원이나 약국 등의 추천을 통해 범용적인 장소 추천 알고리즘으로 활용할 수 있다.

또한 알고리즘 측면에서 보면, 우리는 아이템을 추천하는 데에 아이템의 단편적인 특성(ex. 한식, 로맨스)을 사용한 것이 아닌 사용자들의 리뷰를 사용하였다. 실질적인 리뷰에 중점을 둔 추천 알고리즘을 기획할 때 이와 같은 방식을 사용하면 그 목적을 잘 달성할 수 있을 것으로 생각된다.

5. 결론 및 제언

이번 프로젝트에서 우리는 맛집, 카페, 데이트 명소, 영화를 모두 포함한 데이트 코스 추천 알고리즘을 기획하고 개발하였다. 단일 장소에 대한 추천을 넘어 각 장소들을 조합하여(필터링 등) 데이트 코스로 추천해주었다. 이를 위해 어떤 추천 알고리즘을 사용해야 이 주제에서 가장 적합한 추천을 할 수 있을지 끊임없이 고민하였고, 다양한 추천 알고리즘의 특성 및 활용 방법을 익힐 수 있었다. 또한 최종 추천을 위해 사용되는 자연어 처리/형용사 추출, 형용사 embedding, vector clustering/clustering distance, data normalization, vector 유사도 측정에 대해서도 각각 어떤 방법론들이 있고 어떤 상황에서 사용해야 하는지 등을 찾아보면서 진행했다. 분석에 적합한 방법들에 대해 성능 평가를 동시에 진행하며 최종적인 알고리즘을 완성할 수 있었다.

결과적으로 이번 프로젝트를 통해 추천 시스템의 A-Z를 직접 구현해봄으로써 추천 알고리즘이 어떻게 동작하는지 잘 이해할 수 있었다. 다만, 적용한 추천 알고리즘의 논문이나 내부 로직을 100% 이해하고 적용시킨다면 더 좋은 결과를 도출할 수도 있을 것이라고 생각한다.

6. 한계점

p_id	cbf	ncf	result
b37	0.480479	9.620190e-05	4.622297e-05
1303	0.469894	9.536743e-07	4.481258e-07
1398	0.456894	2.056360e-06	9.395392e-07
e45	0.451930	3.382564e-05	1.528680e-05
f190	0.451930	7.590652e-05	3.430439e-05
...	...	...	...

p_id	cbf	ncf	result
g239	0.000000	9.937845e-01	0.000000e+00
g84	0.171600	2.636588e-01	4.524373e-02
f231	0.244371	7.537377e-02	1.841918e-02
3	0.222064	6.744722e-02	1.497761e-02
e32	0.089166	3.127259e-02	2.788444e-03
...	...	...	...

p_id	cbf	ncf	result
g84	0.171600	0.263659	0.045244
f231	0.244371	0.075374	0.018419
3	0.222064	0.067447	0.014978
c113	0.159505	0.028977	0.004622
a59	0.186089	0.023632	0.004398
...	...	...	...

[그림 16] 왼쪽부터 CBF, NCF, Result 기준 내림차순 정렬

이번 프로젝트는 모델들의 결함을 서로 보충하여 hybrid model을 직접 구축하는 단계로 진행하였다. 실제로 Contents based filtering과 Neural Collaborating filtering의 결과를 조합하는 과정에서 한쪽 모델만 높게 예측한 아이템이 아닌 두 모델 모두 예측값이 좋은 아이템을 찾는 데에 비중을 두게 되었다. 그 과정에서 적절한 조합 방법을 찾지 못했고 단순히 두 값을 곱하는 방법을 사용했으며 결과적으로 최종 result의 prediction값이 낮아졌다는 점이 이 프로젝트의 한계점으로 남는다.

7. 참고문헌

1) Florian Strub 외 2인.「Hybrid Recommender System based on Autoencoders」. 2016.

2) Xiangana He 외 5인.「Neural Collaborative Filtering」. 2017

3) Maxim Naumov 외 23인.「Deep Learning Recommendation Model for Personalization and Recommendation Systems」. 2019

8. 역할 분담 및 활동 내역

주차	이름	추진내용
2 3/27	김동혁	- 1차 프로젝트 제안서(3.기대효과 및 활용방안, 4. 수행방법 일부) 작성 - 맛집/카페 : 수집 가능한 플랫폼 탐색
	류연주	- 1차 프로젝트 제안서(2. 과제 목표, 4. 수행방법 일부) 작성, 최종 취합 - 데이트 명소: 수집 가능한 데이터 탐색
	유정수	- 1차 프로젝트 제안서(1. 과제 개요, 4. 수행방법 일부) 작성 - 전시/영화 : 수집 가능한 데이터 탐색
3 4/3	김동혁	- 맛집/카페 : 망고플레이트를 통해 기본 데이터 및 리뷰 수집
	류연주	- 데이트 명소 : 대한민국 구석구석(장소 데이터), TripAdvisor(리뷰 데이터) 크롤링 작업 시작
	유정수	- 전시/영화 : 왓챠를 통한 영화 데이터 및 리뷰 크롤링
4 4/10	김동혁	- 맛집/카페 : 망고플레이트, 다이닝코드에서 데이터 크롤링 - clustering distance에 대한 research
	류연주	- 데이트 명소 : 대한민국 구석구석(장소 데이터), TripAdvisor(리뷰 데이터) 크롤링 작업 완료
	유정수	- 전시/영화 : 왓챠, 네이버영화에서 데이터 크롤링
5 4/17	김동혁	- 맛집/카페 : data 부족으로 인한 추가 크롤링 - 맛집/카페 : okt 패키지를 이용한 한글 형태소 분석
	류연주	- 데이트 명소 : Kakao Map 장소/리뷰 데이터 크롤링

	유정수	-전시/영화 : 오픈갤러리에서 전시 데이터 크롤링, 네이버 카페에서 전시 리뷰 데이터 크롤링
6 4/24	김동혁	- 맛집/카페 : 한글 리뷰에서 형용사 추출 - 한글 기반 Fast-Text로 임베딩 연구
	류연주	- 리뷰 자연어 처리를 위한 불용어 사전 구축 (한글) - Papago로 데이트 명소 리뷰 데이터 번역 후 tokenizing 테스트
	유정수	- 전시/영화 : 한글 리뷰에서 형용사 추출 - 단순 형용사 추출 방법과 리뷰를 중요 문장으로 요약 후 형용사 추출 방법에 대해 연구
7 5/1	김동혁	- word embedding에서의 CBOW와 SKIPGRAM에 대한 research - 맛집/카페: 데이터 통합 형식으로 변형
	류연주	- 데이트 명소: Kakao Map 데이터 크롤링 완료, 리뷰 및 장소 데이터를 통합 형식으로 변형 - 리뷰 자연어 처리를 Fast-Text 기법 + Papago로 변형한 리뷰데이터로 진행
	유정수	- 전시/영화 : 크롤링 완료 후 데이터 통합 형식으로 변형 - 번역하지 않은 데이터를 Fast-Text 기법을 사용해 embedding
8 5/8	김동혁	- 맛집/카페: 리뷰 영어로 번역 - GMM, Affinity Propagation, Spectral 등 clustering 적용 시도
	류연주	- 리뷰 자연어 처리 Word2Vec, BERT, ELMO, GLoVe research - 데이트 명소: Word2Vec 적용 및 Fast-text와 비교 - 데이트 명소: Kmeans, DBSCAN, GMM 클러스터링 기법 적용
	유정수	- 전시/영화: 전시 리뷰 데이터 부족으로 추가 탐색 - 전시/영화: 한글 데이터에 대해 Word2Vec, Fast-text 적용 및 비교 -embedding한 vector 클러스터링 진행(K-means, K-medoids, DBSCAN)
9 5/15	김동혁	- Collaborative Filtering에서 Matrix Factorization에 대한 research
	류연주	- 데이트 명소: 카카오/트립어드바이저 모든 리뷰 Papago api 사용하여 영어로 번역 - 데이트 명소: User matrix, Item matrix 생성
	유정수	- 전시/영화: Papago api를 사용하여 모든 리뷰 데이터 번역 - 전시/영화: User matrix, Item matrix 생성
10 5/22	김동혁	- Neural Collaborative Filtering 논문에 제시된 코드 연구
	류연주	- 데이트 명소: 카카오/트립어드바이저 모든 리뷰 api 사용하여 영어로 번역 - 데이트 명소: User matrix, Item matrix 생성
	유정수	- NCF 논문 리뷰 및 적용 방안 제시

		- 영화: User matrix, Item matrix 생성
11 5/29	김동혁	- 맛집/카페 : NCF 모델 적용 - 추천시스템 성능 평가방법 research
	류연주	- NCF모델 적용 방향 수정, Hybrid method research - 데이트 명소: NCF 모델 적용 (Microsoft 라이브러리)
	유정수	- Hybrid method 적용 방법 연구 및 라이브러리 탐색 - 영화: Cornac 라이브러리 내부 로직 연구 및 NCF 모델 적용
12 6/5	김동혁	- NCF를 이용한 라이브러리 검색 및 각 장단점 파악 - User Matrix * Item Matrix cosine similarity 방식 테스트
	류연주	- User Matrix * Item Matrix 단순 행렬곱 적용 방식 테스트 - Data Normalization 방식 research
	유정수	- 벡터의 유사도 측정 방법 research - User Matrix * Item Matrix 단순 행렬곱 적용 방식 테스트
13 6/12	김동혁	- 프로젝트 Flow chart 생성 - 맛집/카페 : Matrix Data Normalization
	류연주	- 데이트 명소: 형용사 word-cluster 라벨링, 조사 문항에 넣을 20개 대표 형용사 선정 - 데이트 명소 : Matrix Data Normalization
	유정수	- 영화: cosine similarity 방식으로 변경 후 Matrix Data Normalization
14 6/19	김동혁	- 테스터들에게 보낼 설문지 중, 1) 형용사에 대해 느끼는 호감도 정도 문항 및 3) 지역구 선택 작성 - 맛집/카페 : 최종 모델 완성 - Random Search를 통한 NCF hyper parameter tuning
	류연주	- 테스터들에게 보낼 설문지 중, 2-2) 데이트 명소 문항 작성 - 맛집/카페, 데이트 명소, 영화 데이터를 조합한 최종 알고리즘 코딩 - 지역구 필터링 코딩
	유정수	- 영화: 현재 상영 영화 추천을 위한 데이터 추가 - 영화: 최종 모델 완성
15 6/26	김동혁	- 제작된 알고리즘을 python 파일 및 모듈로 제작 - 영상 촬영
	류연주	- 최종 만족도 조사 방식 제안 - 만족도 설문지 작성 - 최종 발표 데모 영상 시나리오 작성 및 제작
	유정수	- 최종 발표자료 준비

		- 유저별 알고리즘 실행 결과 및 만족도 조사 메일 발송, 취합

※ 본 양식은 요약보고서이며, 최종결과물을 필히 추가 제출하여야 함.

팀 학생대표 성명 : _____ (인)