이커머스 고객 데이터 분석

[목표]

경진대회 데이터를 바탕으로 대회에서 요구하는 목적에 맞는 분석 진행

[역할]

데이터 전처리, 스케일링, 클러스터링(K-means), 프리미엄 고객 칼럼 분석

[데이터]

기본 데이터는 다음과 같다.

Dataset Info.

- **Onlinesales_info.csv [파일]**
  - 온라인거래와 관련된 정보
  - 고객ID : 고객 고유 ID
  - ID : 거래 고유 ID
  - 거래날짜 : 거래가 이루어진 날짜
  - ID : 제품 고유 ID
  - 제품카테고리 : 제품이 포함된 카테고리
  - 수량 : 주문한 품목 수
  - 평균금액 : 수량 1개당 가격 (단위 : 달러)
  - 동일 상품이어도 세부 옵션에 따라 가격이 다를 수 있음
  - 배송료 : 배송비용 (단위 : 달러)
  - 쿠폰상태 : 할인쿠폰 적용 상태
- **Marketing_info.csv [파일]**
  - 마케팅비용과 관련된 정보
  - 날짜 : 마케팅이 이루어진 날짜
  - 오프라인비용 : 오프라인 마케팅으로 지출한 비용 (단위 : 달러)
  - 온라인비용 : 온라인 마케팅으로 지출한 비용 (단위 : 달러)
- **Customer_info.csv [파일]**
  - 고객과 관련된 정보
  - 고객ID : 고객 고유 ID
  - 성별 : 고객 성별
  - 고객지역 : 고객지역
  - 가입기간 : 가입기간 (단위 : 월)
- **Discount_info.csv [파일]**
  - 할인과 관련된 정보
  - 월 : 월(Month) 정보
  - 제품카테고리 : 제품이 포함된 카테고리
  - 쿠폰코드 : 쿠폰코드
  - 할인율 : 해당 쿠폰에 대한 할인율(%)
- **Tax_info.csv [파일]**
  - 세금과 관련된 정보
  - 제품 카테고리 : 제품이 포함된 카테고리
  - GST : Goods and Services Tax(%)

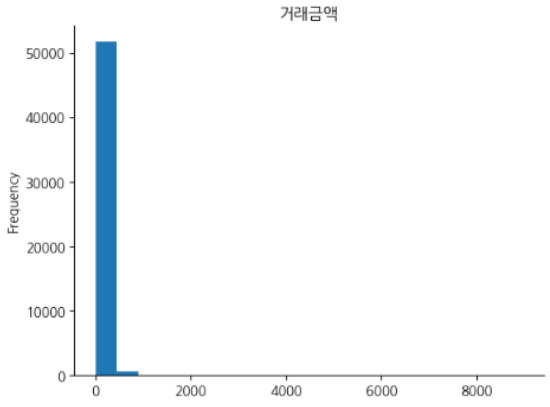
5개의 csv파일을 고유 칼럼인 '고객ID'를 기준으로 하나의 csv파일로 합쳤다.

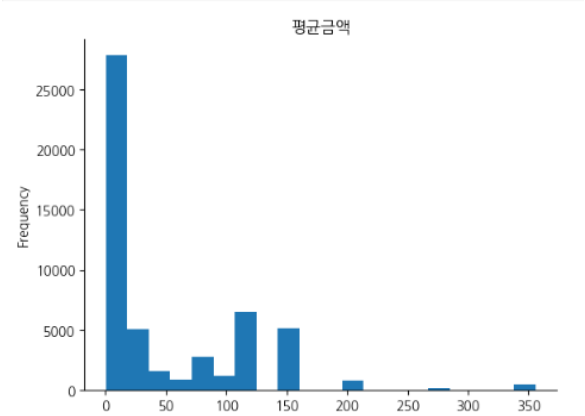
'**고객 세분화하고 그들의 행동 패턴과 구매 경향을 이해**' 하는 것이 목적이었으므로 칼럼 분석을 진행하되 가공이 필요하다고 생각했다.

1. 칼럼 추가
   A. 구매금액 (수량 * 평균금액)
   B. 초중말 (거래날짜 [1~10일 : 초 / 11~20일 : 중 / 20일~ : 말])
   C. 월 (거래날짜 ['Jan', 'Feb', 'Mar' ... 'Dec']
   D. 요일 (거래날짜 ['월요일', ... '일요일']
   E. 가입기간 (가입기간/12 [1년기준])

2. 칼럼 제거
   A. 거래ID, 제품ID, 쿠폰코드, GST (분석과 관련없는 칼럼)
   B. 거래날짜, 평균금액 (내용 추출 완료)

고객 세분화를 진행하기 앞서 스케일링이 필요하다고 생각했다. 그 이유는 데이터 자체가 long tail 성질을 띄고 있어 추후 모델을 돌릴 때 과적합이 될 수 있기 때문이다.





진행한 스케일링 기법은 4가지이다.

1. Log Scaling : 데이터가 양수고, 데이터 사이의 크기가 크므로 이상치의 영향력을 줄이기 위함
2. Robust Scaling : 중앙값과 IQR(사분범위)값을 이용하여 이상치의 영향력을 줄이기 위함.
3. Cox-Box Scaling : 데이터가 양수이므로 정규분포의 형태로 만들어 모델의 성능을 높이기 위함
4. RFM Score : 엄밀히 스케일링 기법은 아니지만 고객 세분화에 주로 사용하는 기법으로 고객을 Recency, Frequency, Monetary를 비교하는 측면에 있어서 스케일링 기법으로 분류하기로 함.

스케일링 방법의 평가는 스케일링된 데이터들을 클러스터링 모델에 돌려 성능을 평가하는 방법으로 진행했다. 각 스케일링된 데이터들을 먼저 Elbow method를 통해서 적절한 K값을 알아내고 K-means 모델에 돌려 나오는 1. Loss function 2. Silhouette Score을 비교하여 적절한 스케일링 방법을 선택할 것이다.

	Elbow	Loss	Silhouette
Log	4	1961	0.3710
Robust	4	1477	0.4621
Cox-Box	4	1182	0.4325
RFM	4	**779**	**0.5237**

Loss Function값은 모델 예측값과 실제값의 차이이므로 적을수록 모델의 성능이 좋고, Silhouette Score값은 해당 데이터가 클러스터에 적절히 배치됐는가를 -1~1 사이로 알려주므로 클수록 성능이 좋다. 결국 스케일링 기법은 RFM으로, 군집수는 4개로 진행하게 되었다.

클러스터링 기법도 마찬가지로 비교 분석을 진행했다.

1. K-means : 가장 일반적인 방법. 간단하고 직관적이라 결과해석에 문제가 없고 대규모 다양한 데이터셋에 적용가능하며 빠름
2. GMM : 정규분포를 바탕으로 하는 모델로 이상치 탐지에 적절하며 소프트 클러스터링을 통해 데이터가 여러 클러스터에 속할 가능성을 고려함
3. DBSCAN : 밀도 차이를 기반으로 한 알고리즘으로, 복잡하고 기하학적인 분포도를 가진 데이터에 이점이 있음. 이상치 탐지에 좋은 성능이 있음

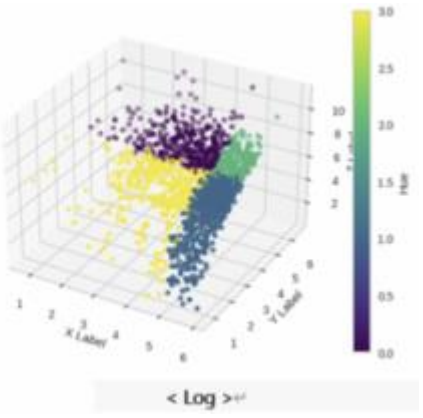
평가방법은 우선 각 기법의 최적의 클러스터 개수를 파악하고, 그 개수대로 진행했을 때 나온 silhouette score를 비교하기로 했다. GMM모델은 BIC라는 모델성능평가지표가 따로 있으므로 해당 지표와 비교했다. 결과는 GMM모델은 적정 클러스터 수가 10개로 세분화를 진행하기에는 너무 많은 개수라 기각했고, K-means가 가장 높은 silhouette score (0.5237)로 결정했다.

즉 RFM분석을 통해 데이터를 스케일링하고 K-means기법을 사용해 클러스터링을 진행했다.

< 클러스터 확인 >

	고객수	R	F	M
0	195	↑	↑	↑
1	390	↓	↓	↓
2	351	↓	↑	↑
3	532	↑	↓	↓

	해석	등급
0	높은 방문율과 구매율을 가졌지만 방문이 뜸해진 고객	재구매 유도 고객
1	자주 방문하지만 높은 실적을 넘기진 않는 금액	일반 고객
2	최근까지도 방문하며 매장에 이익을 가져다 주는 우수 고객	프리미엄 고객
3	방문도 뜸하고 구매율이 적은 고객	이탈 위험 고객



< Log >

4개의 군집을 1. 일반고객 2. 재구매 유도 고객 3. 프리미엄 고객 4. 이탈 위험 고객으로 나누고 칼럼별 분석을 진행했다.

담당한 고객 군집은 프리미엄 고객은 모든 측면에서 가게에 이익을 가져다주는 고객층이다. 따라서 해당 고객층을 유지하는 것에 주의를 기울여야 한다. 그러기 위해선 할인, 경품, 이벤트같은 서비스 차원에서 힘을 쏟는 전략을 세웠다.

1. 고객층 분석

우선 고객층에 대해 분석을 진행했다. 총 7개의 칼럼을 (성별, 고객지역, 시간대(초중말/월/요일), 제품카테고리, 쿠폰상태)기준으로 분석해봤다.

< 성별 >

성별	총 판매수량	총 거래횟수	총 매출	평균 매출	평균 배송료	평균 오프라인비용	평균 온라인비용	고유 고객수	총 판매수량_비율	총 매출_비율	총 거래횟수_비율
남	50824	11404	1097818.69	96.266108	9.963713	2903.367240	1949.829004	175	0.373461	0.381274	0.37549
여	85265	18967	1781522.01	93.927453	10.215217	2936.784942	1935.693429	261	0.626539	0.618726	0.62451

< 고객지역 >

고객지역	총 판매수량	총 거래횟수	총 매출	평균 매출	평균 배송료	평균 오프라인비용	평균 온라인비용	고유 고객수	총 판매수량_비율	총 매출_비율	총 거래횟수_비율
California	40642	8838	877981.04	99.341598	10.676041	3019.891378	1918.275575	126	0.298643	0.304924	0.291001
Chicago	55059	11465	1080225.34	94.219393	10.246071	2947.893589	1941.955157	153	0.404581	0.375164	0.377498
New Jersey	11583	2636	250909.92	95.185857	9.363930	2736.191199	1873.616855	39	0.085113	0.087141	0.086793
New York	23074	5872	526673.54	89.692360	9.497745	2836.665531	2008.520131	92	0.169551	0.182915	0.193342
Washington DC	5731	1560	143550.86	92.019782	9.678250	2855.833333	1922.453583	26	0.042112	0.049855	0.051365

<시간대 - 초중말>

초중말	총 판매수량	총 거래횟수	총 매출	평균 매출	평균 배송료	평균 오프라인비용	평균 온라인비용	고유 고객수	총 판매수량_비율	총 매출_비율	총 거래횟수_비율
초	41975	9031	866206.58	95.914802	9.764517	2920.595726	2014.286079	140	0.308438	0.300835	0.297356
중	48649	10404	982719.97	94.455976	10.404428	3195.934256	2002.963690	133	0.357479	0.341300	0.342564
말	45465	10936	1030414.15	94.222216	10.145134	2668.763716	1821.533963	163	0.334083	0.357865	0.360080

<시간대 - 월>

월	총 판매수량	총 거래횟수	총 매출	평균 매출	평균 배송료	평균 오프라인비용	평균 온라인비용	고유 고객수	총 판매수량_비율	총 매출_비율	총 거래횟수_비율
Jan	5533	1132	110784.73	97.866369	15.491237	3223.586572	2161.224647	50	0.040657	0.038476	0.037272
Feb	6100	1517	137081.28	90.363401	17.143164	3066.183256	2103.202426	1	0.044824	0.047609	0.049949
Mar	8603	1909	158505.09	83.030430	14.376029	2536.930330	1560.253735	25	0.063216	0.055049	0.062856
Apr	14561	1535	202430.06	131.876261	10.645414	3341.693811	2113.867270	2	0.106996	0.070304	0.050542
May	7378	1847	126075.84	68.259794	8.906145	2051.164050	1639.671971	2	0.054215	0.043786	0.060815
Jun	10699	1902	150411.58	79.080747	8.866945	2659.305994	1727.527292	33	0.078618	0.052238	0.062626
Jul	11904	2568	185607.85	72.277200	9.422574	2176.207165	1731.322418	1	0.087472	0.064462	0.084554
Aug	16641	3552	249466.09	70.232570	9.957185	2755.489865	1820.730935	88	0.122280	0.086640	0.116954
Sep	16051	3690	316531.83	85.780984	9.763870	2770.189702	1726.029363	57	0.117945	0.109932	0.121497
Oct	14966	3567	352156.82	98.726330	9.085091	3064.900477	1873.102871	49	0.109972	0.122305	0.117448
Nov	12664	3524	455510.78	129.259586	7.825948	3067.111237	2222.278734	53	0.093057	0.158200	0.116032
Dec	10989	3628	434778.75	119.839788	8.588170	3956.312018	2475.345196	75	0.080749	0.150999	0.119456

<시간대 - 요일>

요일	총 판매수량	총 거래횟수	총 매출	평균 매출	평균 배송료	평균 오프라인비용	평균 온라인비용	고유 고객수	총 판매수량_비율	총 매출_비율	총 거래횟수_비율
월요일	6604	2733	232936.39	85.231025	7.992012	2933.260154	2021.115229	39	0.048527	0.080899	0.089987
화요일	5344	2589	234614.41	90.619703	8.493167	2946.427192	1897.292696	27	0.039268	0.081482	0.085246
수요일	24104	5409	543356.98	100.454239	10.037040	2895.230172	1695.129131	85	0.177119	0.188709	0.178098
목요일	23987	4913	486761.76	99.076279	9.743403	2910.136373	1507.110767	83	0.176260	0.169053	0.161766
금요일	31290	5240	547161.71	104.420174	10.884523	2931.870229	1871.259101	78	0.229923	0.190030	0.172533
토요일	24361	5001	428096.65	85.602210	11.222905	2917.076585	2287.693677	64	0.179008	0.148679	0.164664
일요일	20399	4486	406412.80	90.595809	10.750531	2955.416852	2384.040588	60	0.149895	0.141148	0.147707

<제품카테고리>

제품카테고리	총 판매수량	총 거래횟수	총 매출	평균 매출	평균 배송료	평균 오프라인비용	평균 온라인비용	총 판매수량_비율	총 매출_비율	총 거래횟수_비율
Accessories	934	183	4724.14	25.814973	11.051148	3282.513661	2077.699781	0.006863	0.001641	0.006025
Android	19	19	289.35	15.228947	7.205263	2578.947368	1740.852632	0.000140	0.000100	0.000626
Apparel	19246	10072	337361.15	33.494951	9.344010	2850.744639	1889.989626	0.141422	0.117166	0.331632
Bags	8210	1070	85254.07	79.676701	13.497056	2828.598131	1839.851486	0.060328	0.029609	0.035231
Bottles	1171	145	3872.74	26.708552	12.779034	2793.103448	1922.621379	0.008605	0.001345	0.004774
Drinkware	18876	1919	125181.05	65.232439	14.697822	2896.508598	1911.745857	0.138703	0.043476	0.063185
Gift Cards	108	103	13651.54	132.539223	0.000000	3115.533981	1953.446311	0.000794	0.004741	0.003391
Headgear	2200	431	33068.16	76.724269	9.541601	2826.682135	1898.391183	0.016166	0.011485	0.014191
Housewares	1051	56	2118.50	37.830357	19.573750	2678.571429	1947.421429	0.007723	0.000736	0.001844
Lifestyle	13682	1744	40851.59	23.424077	13.977930	2799.082569	1904.045419	0.100537	0.014188	0.057423
Nest	2462	1904	450173.87	236.435856	7.365572	3319.852941	2153.074785	0.018091	0.156346	0.062691
Nest-Canada	230	165	35493.92	215.114667	9.232485	3033.939394	2047.436303	0.001690	0.012327	0.005433
Nest-USA	12756	8256	1511819.46	183.117667	7.212678	3007.279554	2004.130159	0.093733	0.525058	0.271838
Notebooks & Journals	6643	376	75286.86	200.231011	20.778059	2647.340426	1826.703936	0.048814	0.026147	0.012380
Office	47744	3608	156065.11	43.255297	14.231749	2856.125277	1899.465327	0.350829	0.054202	0.118798
Waze	757	320	4129.19	12.903719	8.637781	2933.750000	1913.128250	0.005563	0.001434	0.010536

<쿠폰상태>

쿠폰상태	총 판매수량	총 거래횟수	총 매출	평균 매출	평균 배송료	평균 오프라인비용	평균 온라인비용	고유 고객수	총 판매수량_비율	총 매출_비율	총 거래횟수_비율
Clicked	68585	15408	1450302.76	94.126607	10.265616	2923.435877	1941.106698	221	0.503972	0.503693	0.507326
Not Used	19631	4629	463003.35	100.022327	9.744264	2927.003672	1948.663556	72	0.144251	0.160802	0.152415
Used	47873	10334	966034.59	93.481187	10.073485	2924.191988	1937.411614	143	0.351777	0.335505	0.340259

종합해보자면 프리미엄 고객층은 **여성**이 남성에 비해 총 판매수량(+167%), 총 거래횟수(+166%), 총 매출(+162%)을 기록했고, 전체적으로 60%이상의 비중을 차지하고 있다.

지역에서는 **Chicago**(미국 중서부)와 **California**(미국 서부)에서 판매수량, 매출, 거래횟수의 70%를 담당하고 있다.

시간대별로 살펴보자면, 초(1일~10일) 중(11일~20일) 말(21일~) 시기별로는 큰 차이가 없고, 그나마 **중순**에 소비패턴이 늘어난 것을 볼 수 있었다.

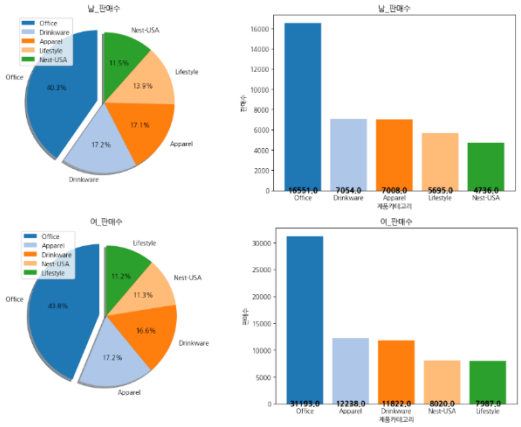
월단위로 보자면 연초에는 소비가 적다가 **4월**에 한번 소비가 늘어나고 **8월부터 연말**까지도 높은 소비율을 보였다. 4월에는 연말 신고 기간, 따뜻해진 날씨, 학기의 시작 8월 이후는 명절, 학기의 시작, 연말 휴가 등 여러 요인으로 인한 것으로 보인다. 따라서 이 시기를 적절하게 이용하면 될 것 같다.

요일단위에서는 예상한 그대로 **금요일**에 판매수량이 가장 많았지만 의외로 **수요일**에는 거래량이 적음에도 매출이 금요일과 비슷하게 나왔다.

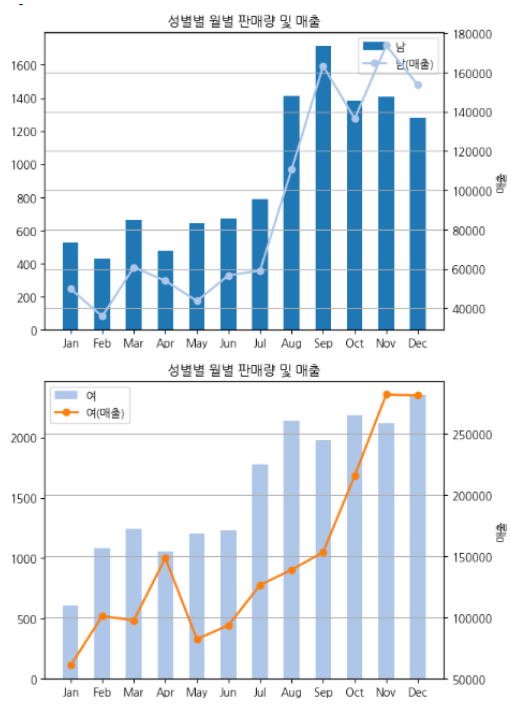
제품별로는 **Office**(사무용품)이 제일 많은 판매량을 가졌지만 그에 비례해 매출은 나오지 않았다. 그에 이어서 **Apparel**, Drinkware, Lifestyle, **Nest-USA**가 순위에 있지만 Apparel은 판매수량 대비 매출이 준수하게 나오고, Nest-USA가 판매되는 수량도, 금액도 매우 높기 때문에 해당 제품에 대한 관리가 필요하다. 참고로 다른 **Nest**+@ 제품들도 높은 수익성을 보장하고 있다.

쿠폰같은 경우 **Clicked**(클릭만하는 고객)가 전체의 50%를 차지하고 Used(쿠폰을 사용한 고객)은 35%밖에 안됐다. 따라서 클릭에서 사용으로 전환할 수 있도록 쿠폰이 넓은 범용성을 가지고 가야한다.

2. 판매량, 매출이 높은 제품에 대한 장려 (상위 5)



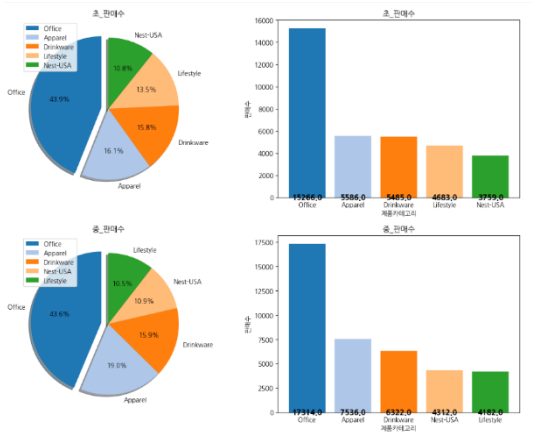
성별별로 판매량이 높은 제품들을 확인해 봤다. 비율을 봤을 때 남성/여성간의 제품 선호도은 큰차이가 보이지는 않았다. (각 성별 다 비슷한 제품을 구매를 한다) 그나마 여성은 Drinkware보단 Apparel을 선호하는 것으로 보인다. 하지만 판매량은 여성이 압도적으로 많으므로 여성에게 집중적으로 마케팅을 해야한다,



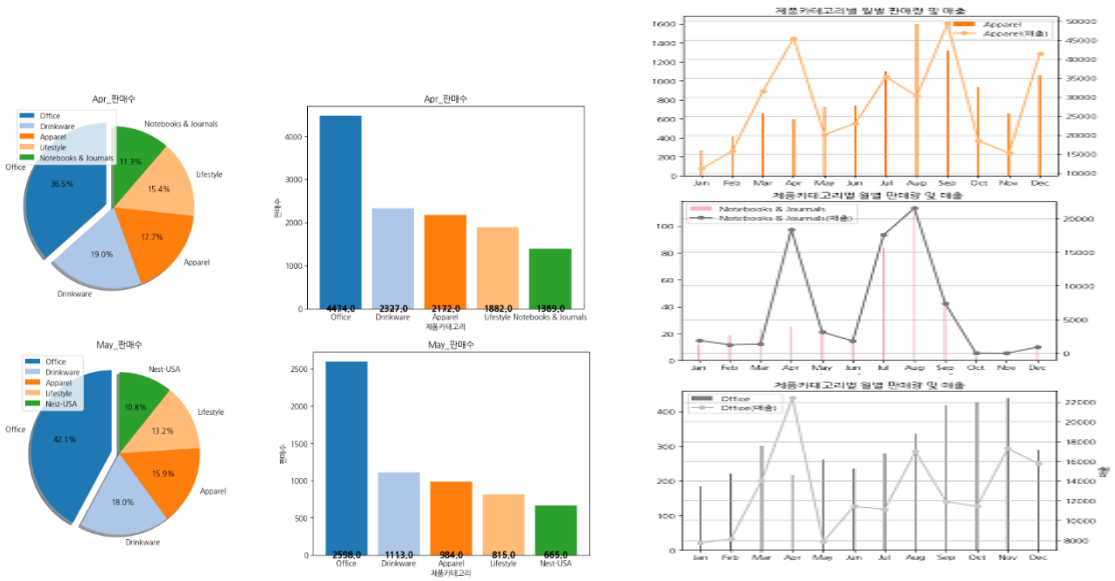
성별별 월별 판매량, 매출이다.

남성은 8,9월에 급격하게 소비량이 늘어나고, 여성은 4월에 한번 많은 소비를 하고 5월부터 천천히 소비량을 늘려나간다.

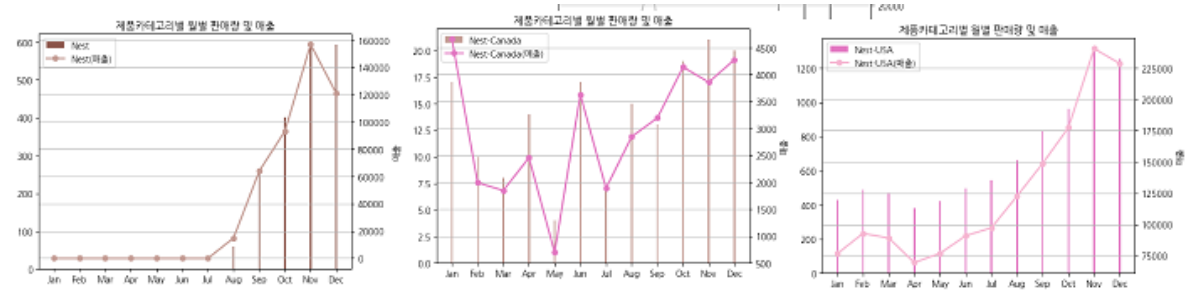
지역별 판매량이다. 높은 판매량과 매출을 가진 Chicago와 California지역을 보면 소비내역이 비슷하다. New Jersey는 Lifestyle / New_York 과 Washington DC지역은 Apparel의 비중이 높았다. 지역별로 달리 상품에 대한 마케팅을 진행해야한다.



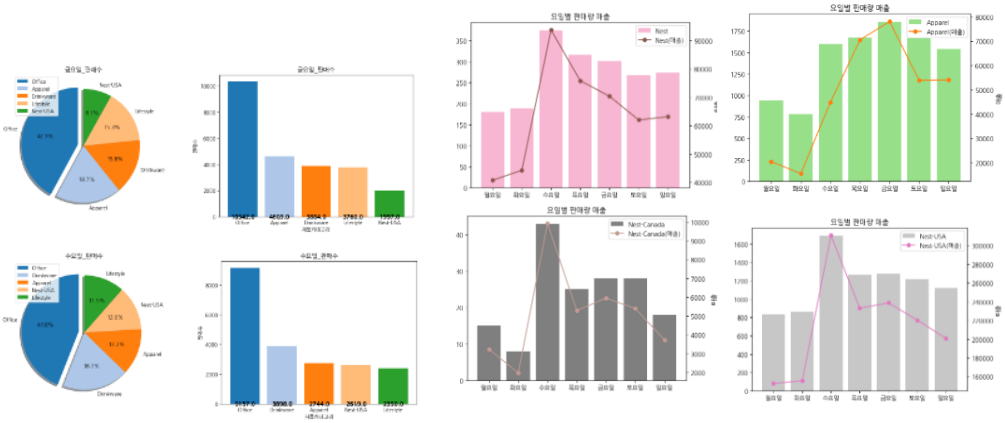
시간대(초중말)별 판매량이다. 중순에 소비가 늘어나는 것은 전체적으로도 판매량이 증가했지만, Apparel과 Office 제품에서 특히 늘었다. 즉, 중순에는 Apparel과 Office에 대한 소비가 많아지는 시기이므로 두 제품에 대한 마케팅을 진행해야한다.



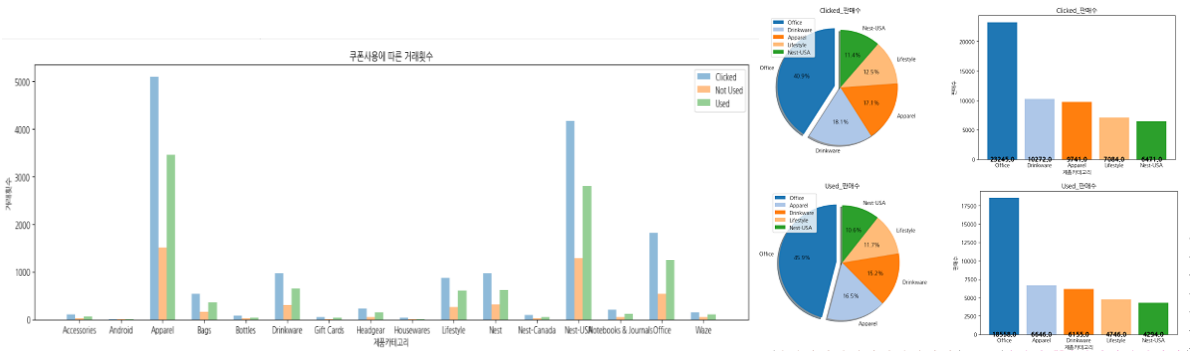
시간대(월)별 판매량이다. 4월, 8월달에는 전체적으로도 올랐지만 Apparel과 Notebooks & Journals의 판매량이 증가한걸 봐서 '학기의 시작'이 매출에 영향을 미친 것을 알 수 있었다. 또한 거꾸로, 5월달의 Office 판매량이 4월에 비해 판매량이 절반으로 줄어든 것을 봐서 4월달의 '연말 신고 기간'이 끝난 후, 새로운 예산을 절약하기 위함이라고 생각된다. 따라서 4월달의 소비가 늘어난 것에 대해 '연말 신고 기간'도 영향을 미친다고 추측할 수 있다.



특이하게도 8월부터 12월까지는 판매량은 줄지만 매출은 늘어난다. 이익을 많이 낼 수 있는 'NEST' 계열의 제품들의 판매량을 확인해보니 판매량이 증가한 것으로 보였다



시간대(요일)별 판매량이다. 수요일부터 살펴보면 판매수가 적지만 매출이 금요일 수준으로 높은 것은 역시 NEST-USA 제품의 영향이 크다. 추측하건데 NEST 제품들은 설치가 필요로 하는 제품들이기 때문에 설치기사가 집을 방문해야 한다. 따라서 물건이 주말에 도착해야 하므로 수요일날 주문을 하는 것으로 보인다. 금요일은 수익률이 높은 Apparel 제품이 많이 판매되기 때문에 매출도 높은 것으로 추정된다.



쿠폰상태별 판매량이다. Clicked에서 Used로 전환이 중요하므로, 고객이 실제로 쿠폰을 써서 구매한 물건과, 쿠폰을 쓰려고 했으나 어떤 이유로 인해 쿠폰을 쓰지 않고 구매한

물건을 살펴봐야한다. 살펴보니 제품목록은 동일했고, 해당 제품들에 대하여 쿠폰의 범용성을 늘려서 고객 충성도를 올려야한다. 예를 들면 요즘 쿠폰은 '특정 금액 이상 구매시', '특정 브랜드 구매시' 같은 조건들이 붙어있기 때문에 쿠폰이 있다고해도 못쓸 수 있으므로, 그런 제한 없이 유연하게 쓸 수 있는 쿠폰을 제공한다면 쿠폰 사용량이 늘 것으로 예상된다. 달리 쿠폰을 안쓰는 고객들에게는 마케팅을 더 하여 충성도를 늘리는 방법을 사용하면 좋을 것 같다.

카페추천시스템

[목표]

카페를 방문하는 목적이 다양해졌으므로 목적에 맞는 카페를 추천해주는 시스템이 필요

[역할]

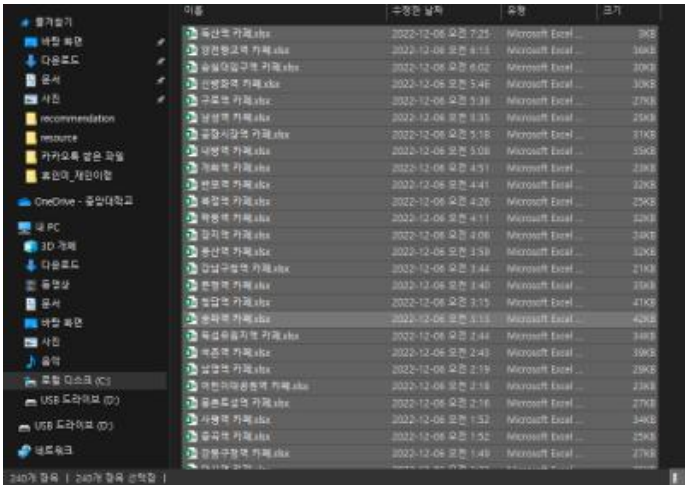
데이터 수집 자동화, 전처리, 평가, 시각화

[데이터]

원하는 데이터가 없어 카페 데이터를 직접 크롤링을 통해 수집하기로 했다. 여러 웹 지도들을 살펴본 결과 '네이버 지도'의 리뷰 탭에서 이미 labeling된 지표들이 있어 활용하기로 했고, 영수증 인증을 통해 실제로 구매한 사람들만 리뷰를 남길 수 있다는 점에서 신뢰성도 확보했다.

```
14  browser = webdriver.Chrome("./chromedriver.exe")
15  browser.get("https://map.naver.com/v5/")
16  browser.implicitly_wait(10)
17  browser.maximize_window()
18
19  search = browser.find_element_by_css_selector("input.input_search")
20  search.click()
21  time.sleep(1)
22  search.send_keys("강남역 카페")
23  time.sleep(1)
24  search.send_keys(Keys.ENTER)
25  time.sleep(2)
```

위 스크린샷은 크롤링 코드의 일부분으로 실제 크롤링은 서울시에 위치한 역사명을 받아, 자동화된 크롤링을 수행하게 된다

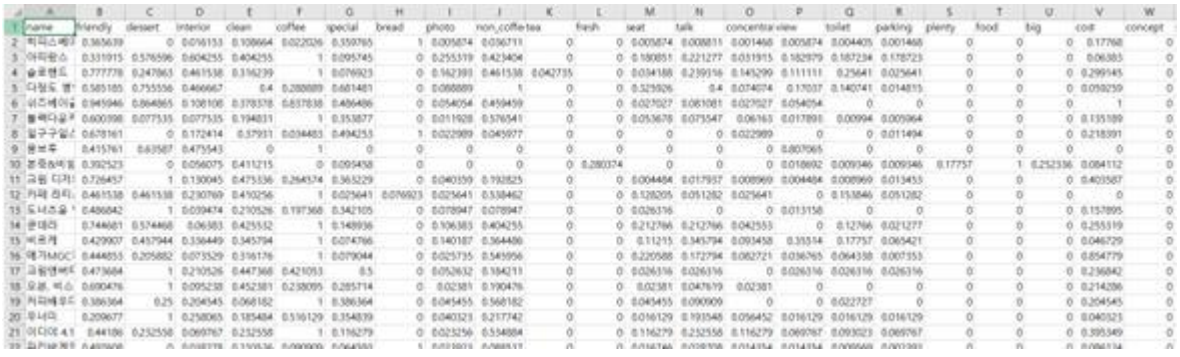


'서울시 지하철역 + 카페' 라는 쿼리로 크롤링을 진행하여 총 2만여개의 카페 데이터와 103개의 feature를 수집 완료했다. 그 후 데이터들을 하나의 파일로 합치고 전처리를 진행했다.

1. 중복제거 : 역간 거리가 짧은 경우 중복되는 카페들이 있다.
2. 이상치 제거 : 방탈출카페, 팝업스토어 등 목적 데이터와 다른 데이터들을 제거했다.
3. Relabeling : 성능을 높이기 위해 지표를 통합/폐기하여 relabeling을 진행했다.

열1	friendly	desser	interior	clean	coffee	special	bread	photo	non_co	tea	fresh	seat	talk
히피스베이글	249	0	11	74	15	245	681	4	25	0	0	4	6
아띠랑스	156	271	284	190	470	45	0	120	199	0	0	85	104
슬로핸드	91	29	54	37	117	9	0	19	54	5	0	4	28
다정도 병인 양	79	102	63	54	39	92	0	12	135	0	0	44	54
쉬즈베이글 먹대점	35	32	4	14	31	18	0	2	17	0	0	1	3

4. 스케일링 : 추후 모델링을 위해서 모든 데이터를 최대값으로 나눠서 0~1 사이 값을 갖도록 했다. (정규화)



추천을 위해 알고리즘을 선택해야 했다. 유저 피드백이 없는 Cold start 상태이기에 content based filtering을 바탕으로 하기로 했고 LightFM이라는 모델을 사용하여 cold start 문제를 그나마 해결하려고 했다.

[평가]

```
cafename = '서촌금상고로케'

result = recommend_cafe_list(data,cafe=cafename)


# data['name'] = data['열1']
# data.set_index('열1', inplace=True)
data.rename(columns={'열1':'name'},inplace=True)
result.rename(columns={'열1':'name'},inplace=True)
index = data.index[(data['name'] == cafename)]

user = data.iloc[index]

print(result['name'])
```

✓ 28.8s

```
14997      꽈페 홍대점
8497        부트브레드
14900    꿀넹쿠키 연남점
1037        일팔공일오
11737       빵미제빵소
Name: name, dtype: object
```

유저가 긍정적으로 평가한 카페 명을 입력값으로 받으면, 해당 카페와 비슷한 5개의 카페들을 추천해주도록 설계했다.

왼쪽은 예시로 '서촌금상고로케'와 비슷한 feature들을 가진 카페들을 추천해주었고 아래는 해당 카페들의 feature로 비슷한 값을 갖고 있음을 알 수 있다.

열1	friendl	desser	interio	clean	coffee	special	bread	photo	non_co	tea	fresh	seat	talk
일팔공일오	174	0	26	97	36	126	326	9	10	0	0	3	2
서촌금상고로케	2397	0	115	1499	191	2250	2904	224	277	0	0	115	214
부트브레드	130	0	13	61	40	61	202	4	20	0	0	2	3
빵미제빵소	131	0	3	98	30	89	231	0	17	0	0	0	1
꿀넹쿠키 연남점	162	0	46	98	38	176	209	30	28	0	0	9	32
꽈페 홍대점	266	0	111	152	63	363	478	96	63	0	0	34	48

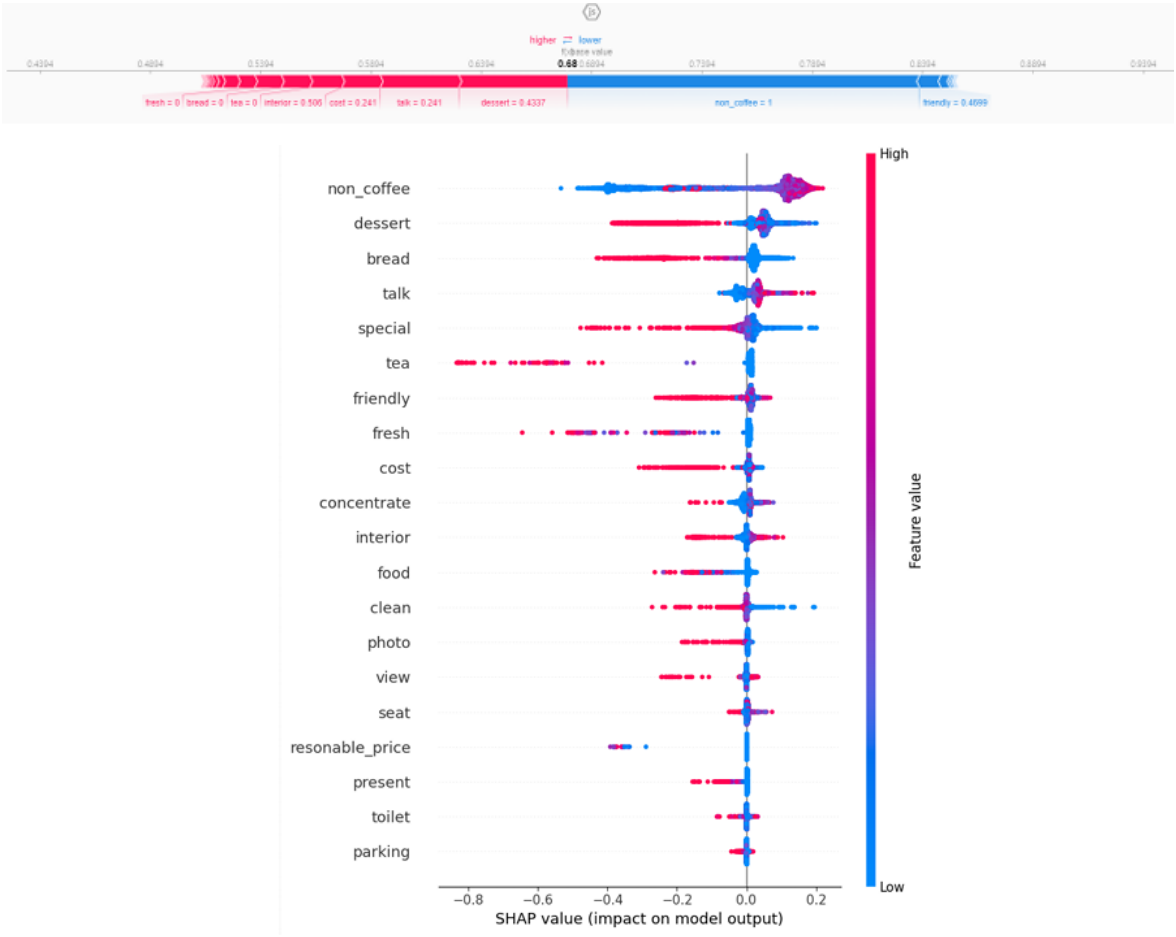
```
# lightgbm을 구현하여 shap value를 예측할 것
# ligthgbm 구현

# library
import lightgbm as lgb  # 없을 경우 cmd/anaconda prompt에서 install (LightGBM: Light Gradient-Boosting Machine)
from math import sqrt
from sklearn.metrics import mean_squared_error

# lightgbm model
lgb_dtrain = lgb.Dataset(data = train_x, label = train_y) # LightGBM 모델에 맞게 변환
lgb_param = {'max_depth': 20,       # original: 10
             'learning_rate': 0.01, # Step Size
             'n_estimators': 1000, # Number of trees
             'objective': 'regression'} # 목적 함수 (L2 Loss)
lgb_model = lgb.train(params = lgb_param, train_set = lgb_dtrain) # 학습 진행
lgb_model_predict = lgb_model.predict(test_x) # test data 예측
print("[REDACTED]: {}".format(sqrt(mean_squared_error(lgb_model_predict, test_y)))) # RMSE
```

✓ 1.1s

```
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves. (num_leaves=31).
[LightGBM] [Warning] Accuracy may be bad since you didn't explicitly set num_leaves OR 2^max_depth > num_leaves. (num_leaves=31).
[LightGBM] [Warning] Auto-choosing col-wise multi-threading, the overhead of testing was 0.001783 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 4493
[LightGBM] [Info] Number of data points in the train set: 10682, number of used features: 41
[LightGBM] [Info] Start training from score 0.689399
RMSE: 0.1278267980246556
```

LightFM을 통해 정확도 평가를 진행했을 때 RMSE값이 0.1278로 높은 정확도를 갖고 있음을 알 수 있다.

LightFM 모델을 사용해 예측한 결과에 대해서 해석/평가하기 위해 SHAP를 사용했다. SHAP는 우리가 relabeling한 특성들이 모델의 예측에 얼마나 기여하는지를 나타내고 특성들간의 상관관계를 나타낼 수 있었다.



해당 그래프들은 '커피가 맛있다'라는 지표를 기준으로 양/음의 상관관계를 가진 칼럼들을 보여준 것이다. 예를 들면 '커피가 맛있다'고 평가한 고객들은 '디저트가 맛있다'라는 평가도 같이 남기지만 '논커피가 맛있다'라는 평가는 남기지 않는 경향이 있다는 것을 알 수 있었다.

멘티 멘토 매칭 앱

[목표]

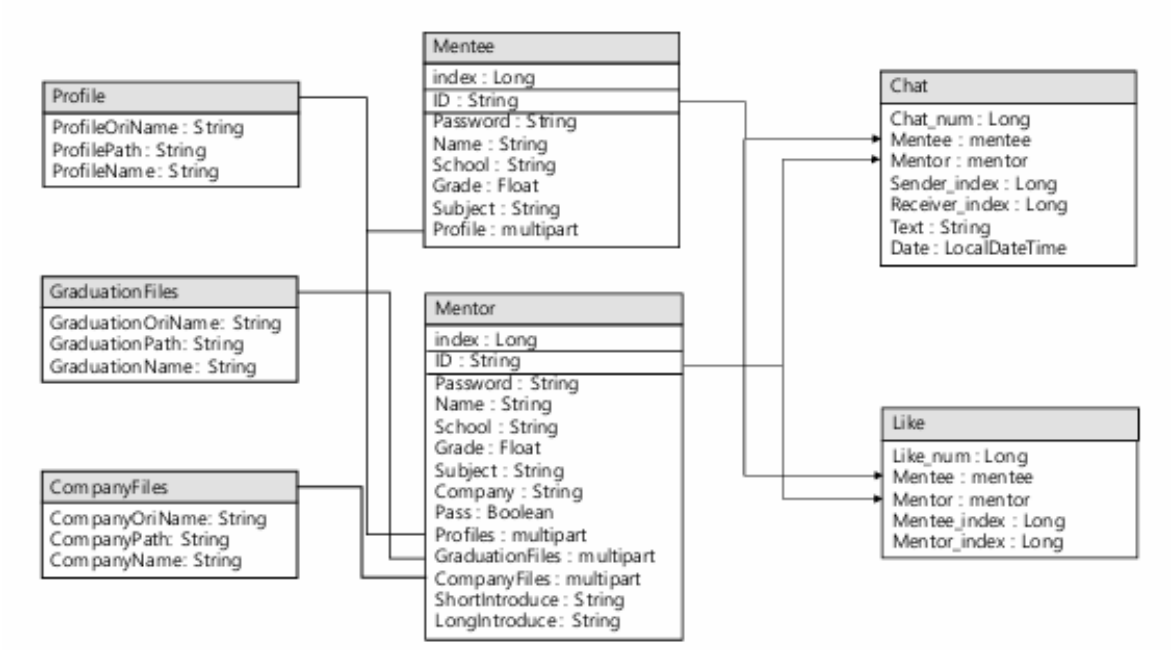
보다 현실적이고 유의미한 조언을 위해 서로 비슷한 멘토와 멘티끼리 매칭해주는 프로그램이 필요

[역할]

GCP 서버 구축, RESTful을 이용한 모바일 연동, MySQL을 이용한 데이터 저장, 관리자페이지를 통한 유저 데이터 저장 및 수정, 매칭 알고리즘 및 앱 기능 구현

[데이터]

앱에서 입력한 정보들을 GET/POST 방식으로 받아 데이터베이스에 저장했다. 최대한 멘토와 멘티가 비슷한 환경을 가져야 하기 때문에 '학교' '성적' '학과'를 KPI로 선정하고 수집했다.



필요와 기능에 따라 총 7가지 테이블을 만들었다.

1. 기본 데이터 테이블
    A. 멘티 테이블
    B. 멘토 테이블
2. 기능 테이블
    A. 채팅 테이블
    B. 좋아요 테이블

3. 이미지 테이블
    A. 프로파일 테이블
    B. 졸업증명서 테이블
    C. 재직증명서 테이블

1-A. 멘티 테이블

- Mentee

Index	Mentee의 고유 index 값
ID	Mentee의 ID
Password	Mentee의 password
Name	Mentee의 이름
School	Mentee의 학교
Grade	Mentee의 성적
Subject	Mentee의 학과
Profiles	Mentee의 프로필 정보

멘티 정보를 받는 테이블이다. 선정한 KPI (학교, 성적, 학과)와 기본 정보들을 저장한다.

1-B. 멘토 테이블

- Mentor

Index	Mentor의 고유 index 값
ID	Mentor의 ID
Password	Mentor의 password
Name	Mentor의 이름
School	Mentor의 학교
Grade	Mentor의 성적
Subject	Mentor의 학과
Company	Mentor의 회사
Pass	Mentor의 승인여부
Profiles	Mentor의 프로필 정보
GraduationFiles	Mentor의 졸업증명서 정보
CompanyFiles	Mentor의 재직증명서 정보
ShortIntroduce	Mentor의 한 줄 자기소개
LongIntroduce	Mentor의 자기소개

멘토 정보를 받는 테이블이다. KPI와 더불어 해당 경력이 신뢰성이 있는지 관리자로부터 확인이 필요하다고 생각해서 추후 관리자 페이지에서 따로 승인 절차를 받도록 했다. 참고로 해당 앱은 '멘티'가 주고객층으로 멘티가 멘토를 선택하는 과정을 거치기 때문에 멘토의 자기소개 등을 받아 멘티가 판단할 수 있도록 했다.

멘토의 경우 관리자의 검증을 거쳐 승인을 받아야 앱에 등록될 수 있으므로 관리자페이지에선 멘토의 정보들을 따로 데이터베이스에서 불러와 승인 절차를 거칠 수 있도록 했다.

2-A. 채팅 테이블

Chat_num	채팅의 고유 index 값
Mentee.ID	Mentee 정보
Mentor.ID	Mentor 정보
Sender_index	채팅을 보내려는 사람의 고유 index
Receiver_index	채팅을 받으려는 사람의 고유 index
Text	채팅으로 보내려는 메시지
Date	채팅이 전송된 시간

채팅 기능을 구현하기 위한 테이블이다. 상호간 채팅이 전달되어야 하므로, Sender와 Receiver를 받아 고유 인덱스를 통해 처리하도록 설계하였다.

채팅의 경우 다른 테이블들과 다르게 실시간으로 소통하고 있다는 느낌을 주고 싶었기 때문에 일정시간이 지나면 새로고침을 하도록 했다.

2-B. 좋아요 테이블

- Like

Like_num	좋아요의 고유 index 값
Mentee.ID	Mentee 정보
Mentor.ID	Mentor 정보
Mentee_index	좋아요한 Mentee 의 고유 index
Mentor_index	Mentee 에게 좋아요를 받은 Mentor 의 고유 index

좋아요 기능을 구현하기 위한 테이블이다. 멘티가 멘토 프로필을 보고 좋아요를 눌렀을 때 멘티와 멘토 둘 다 정보가 필요하므로, 두 주체의 정보를 받되 빠른 처리를 위해 고유 인덱스만 받도록 설계하였다.

3-A, B, C. 증명서 관련 테이블

- Profiles

ProfileOriName	저장된 프로필의 원본이름
ProfilePath	프로필이 저장된 경로
ProfileName	중복 방지를 위한 임의값 + 프로필 이름

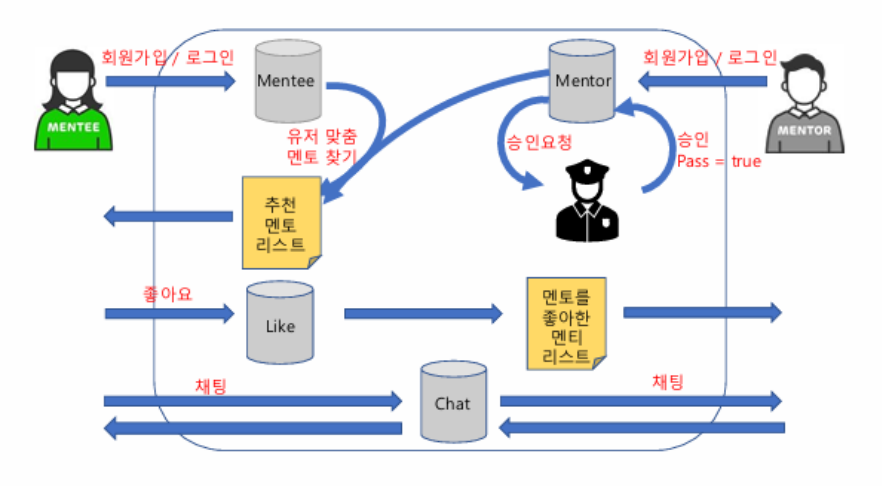
- GraduationFiles

GraduationOriName	저장된 졸업증명서의 원본이름
GraduationPath	졸업증명서가 저장된 경로
GraduationName	중복 방지를 위한 임의값 + 졸업증명서 원본이름

- CompanyFiles

CompanyOriName	저장된 재직증명서의 원본이름
CompanyPath	재직증명서가 저장된 경로
CompanyName	중복 방지를 위한 임의값 + 재직증명서 원본 이름

증명서 관련 테이블들이다. 파일, 이미지를 처리했고 파일을 이용하기 위해 필요한 값들을 저장했다.



전체적인 흐름은 위와 같다. 멘티와 멘토는 회원가입을 거쳐 각 데이터베이스에 정보를 저장하고, 멘토의 경우에는 관리자의 승인 요청 이후에 데이터베이스의 정보들이 유효하게 된다. 그 후 매칭 알고리즘을 통해 추천된 멘토들이 멘티에게 보여진다. 멘티는 추천받은 멘토들과 좋아요, 채팅 기능을 통해 상호작용할 수 있고 이는 전부 각각의 테이블에 저장된다.

[결과]

해당 서비스의 기대효과는 총 3가지이다.

1. 서비스 이용의 자유성

타 멘토링 서비스들의 강의형식이 아닌 개인과 개인의 매칭인 점에서 공간적, 시간적으로 자유롭고, 부담감이 적다.

2. 분야의 확장성

현재는 취업, 학업과 관련된 멘토링을 주 기능으로 쓰고 있지만, 다른 다양한 분야에서도 필요성이 두드러진다. 숙련된 사람들의 노하우를 알려주는 것도 멘토링의 일부이기 때문에 다양한 멘티들에게 다양한 멘토링을 해줄 수 있을 것이다.

3. 멘토링 효과 증가

대부분의 멘토링 시스템들은 단순히 고스펙이 우선이다. 하지만 이 서비스는 멘토와 멘티의 유사성을 위주로 보기 때문에 멘토들은 진입장벽이 낮고 멘티들은 효과적인 멘토링을 받을 수 있을 것이다.