카페추천시스템

[목표]

카페를 방문하는 목적이 다양해졌으므로 목적에 맞는 카페를 추천해주는 시스템이 필요

[역할]

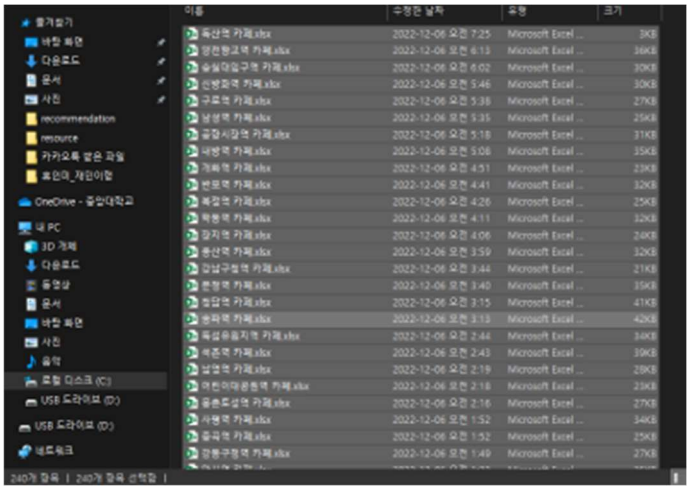
데이터 수집 자동화, 전처리, 평가, 시각화

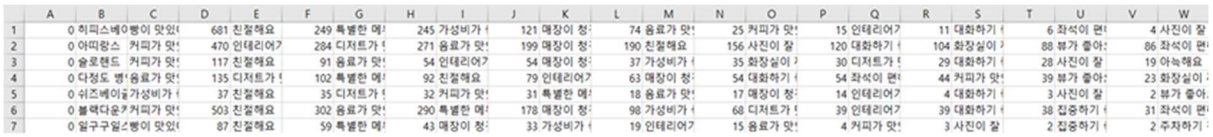
[데이터]

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



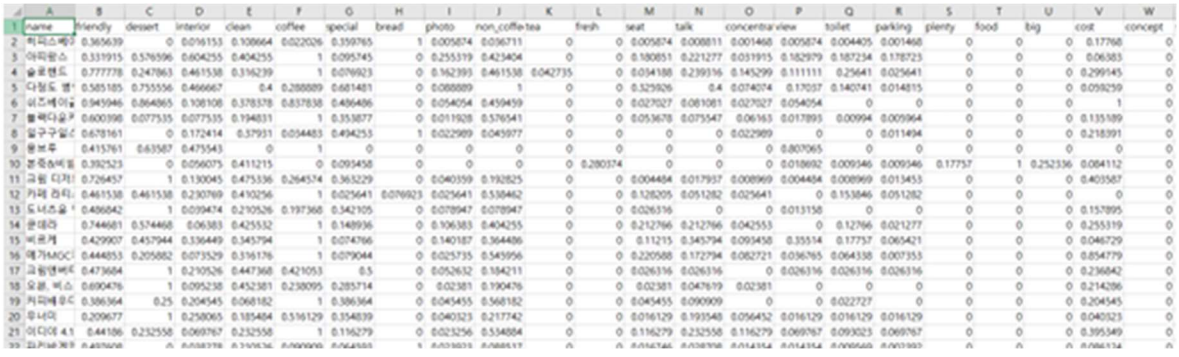


'서울시 지하철역 + 카페' 라는 쿼리로 크롤링을 진행하여 총 2만여개의 카페 데이터와 103개의 feature를 수집 완료했다. 그 후 데이터들을 하나의 파일로 합치고 전처리를 진행했다.

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 덕대점	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

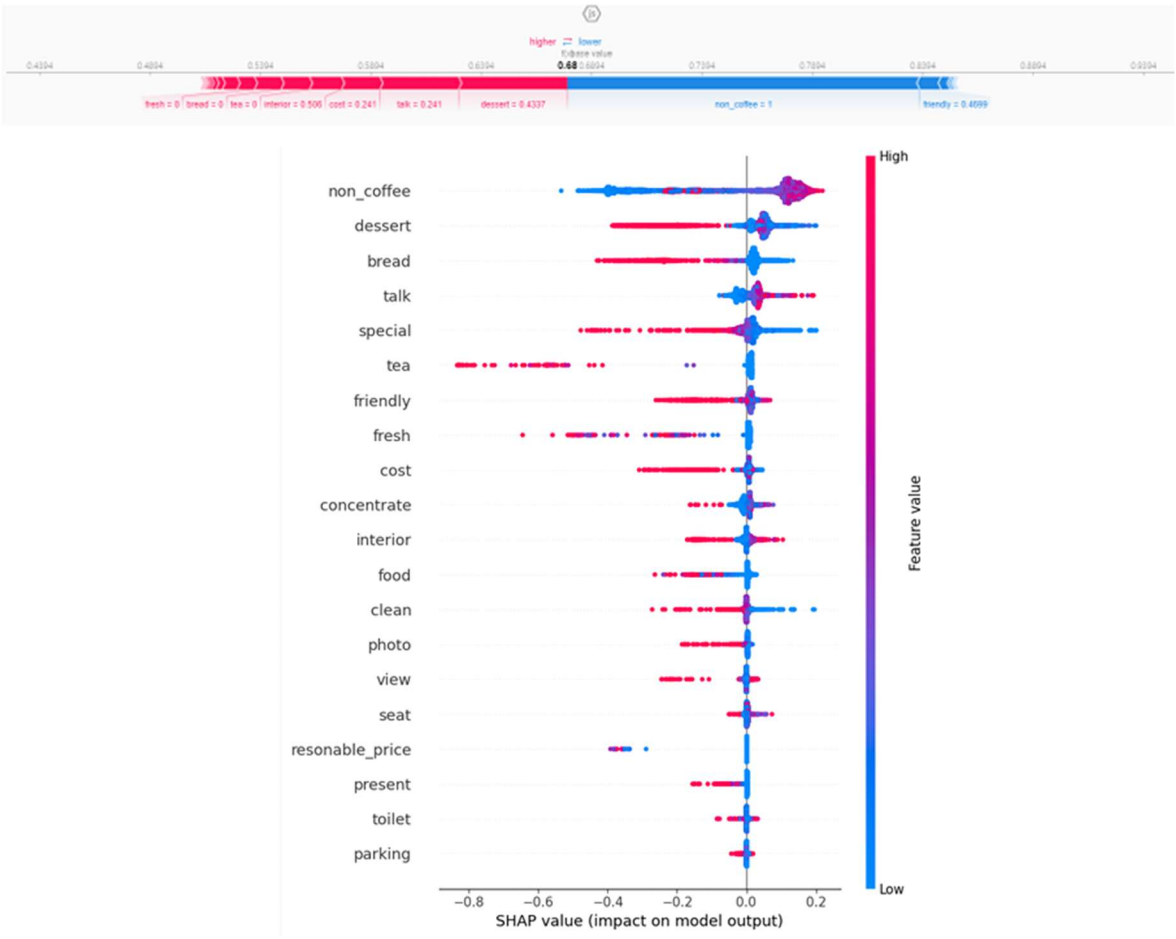
```
# lightgbm을 구현하여 shap value를 예측할 것
# ligthgbm 구현

# 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("RMSE: {}".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.