



Lee, Min-Sung

Github : <https://github.com/Minsung-commit>
Notion : <https://url.kr/t6hijx>





Profile

이민성 Lee, Min-Sung

1993.12.03
010-3641-6141
lgt302@hanmail.net

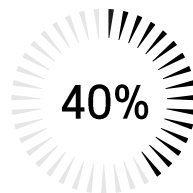
총신대학교 영어교육과 졸
멀티캠퍼스 DS전문가 과정 수료

Interest :
ML, Recommendation System,
Data Visualization, NLP

01 Projects

코로나 시대, 안전한 여행을 위한 SNS기반 감성숙소 추천 앱서비스

기여도



Data

- ✓ 인스타그램 감성숙소 계정 7개의 텍스트 데이터
- ✓ 네이버 블로그 데이터
- ✓ 코로나 거리두기 단계 & 네이버 Place API(분석X)

Environment

- ✓ Python 3.6+ / Tableau / Google Trends
- ✓ Django / AWS / HDFS / Spark / MongoDB

What I learned

1. 형태소 분석이 제대로 이루어지지 않으면 군집 분석에서 잘못 생성된 단어가 집계되는 등의 오류가 높아짐.
2. 서비스에서 요구하는 기능을 정의할 때엔 구현의 우선순위와 필요 작업목록을 계산해야 함.
3. 텍스트 데이터만으로도 공간디자인적 특성을 구분하여 분류하는 것이 가능함..

Purpose

- ✓ SNS데이터를 통해 감성숙소의 새로운 분류기준을 제시
- ✓ 코로나 관련 정보와 감성 숙소 추천서비스를 제공
- ✓ 코로나 시대 속 안전한 여행을 돕고자 함

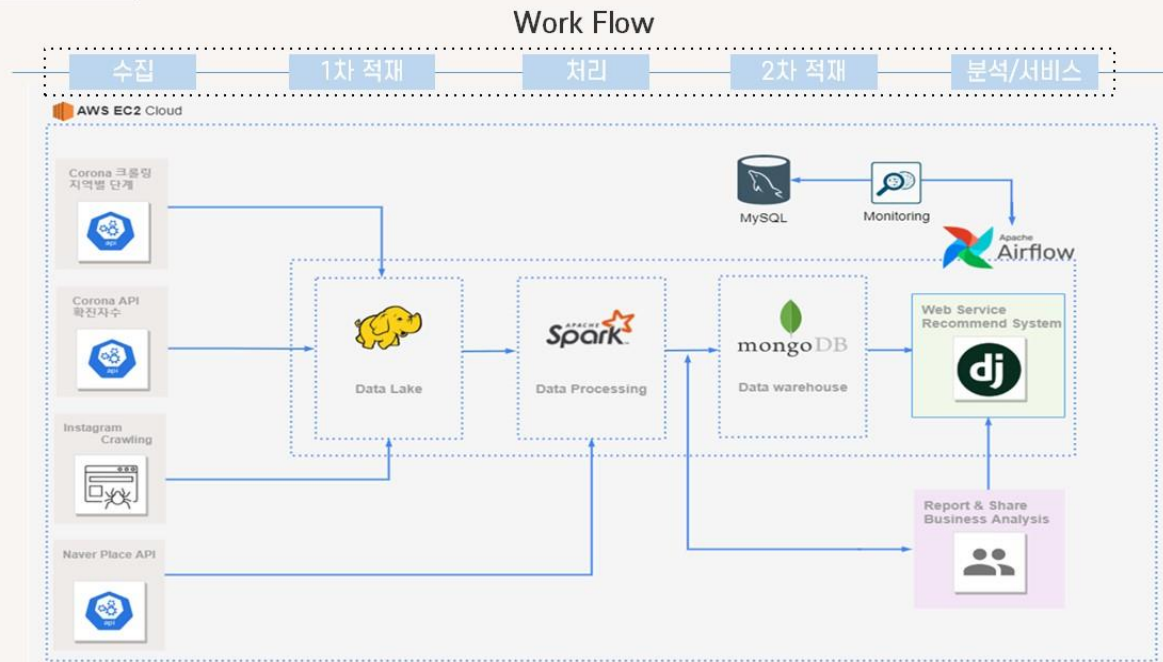
Methodology

- ✓ 경향성 및 패턴 분석 ➡ Spherical K-means 군집분석
- ✓ 군집별 토픽 모델링 ➡ TextRank 키워드 분석
- ✓ UserInfo의 부재 ➡ CB 필터링을 적용한 추천모델

01 Projects

Outcomes : 시스템 아키텍처

시스템 아키텍처



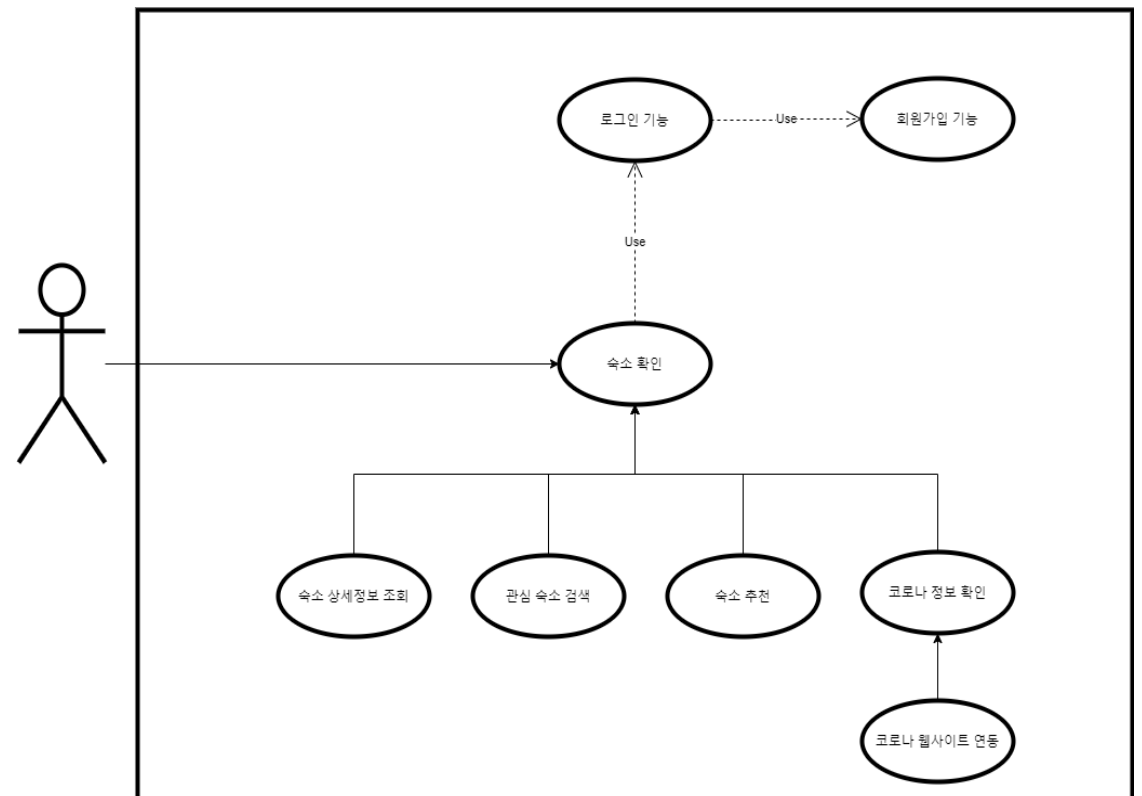
Description

1. 하둡 : 수집 데이터 저장
2. 스파크 : 수집 데이터 전처리
3. 클러스터링 모델 : 숙소 카테고리 분류
4. 몽고database : 분류된 숙소 정보 저장
5. 추천모델 : Django 내에서 몽고 database 연결

01 Projects

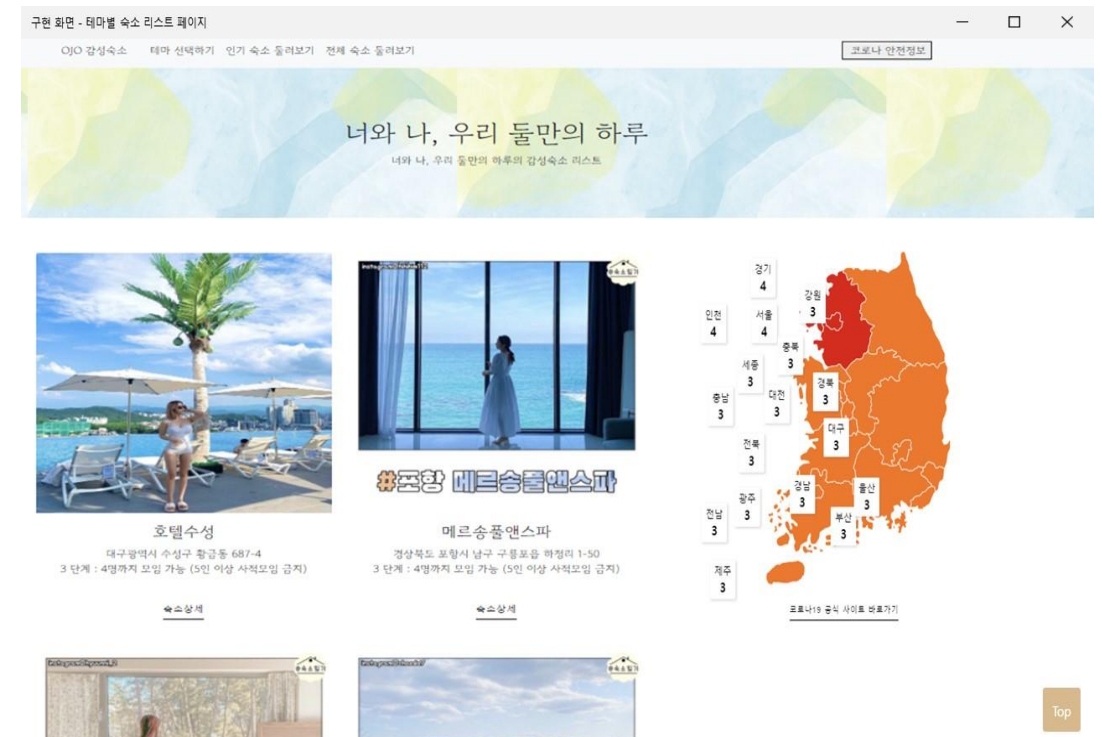
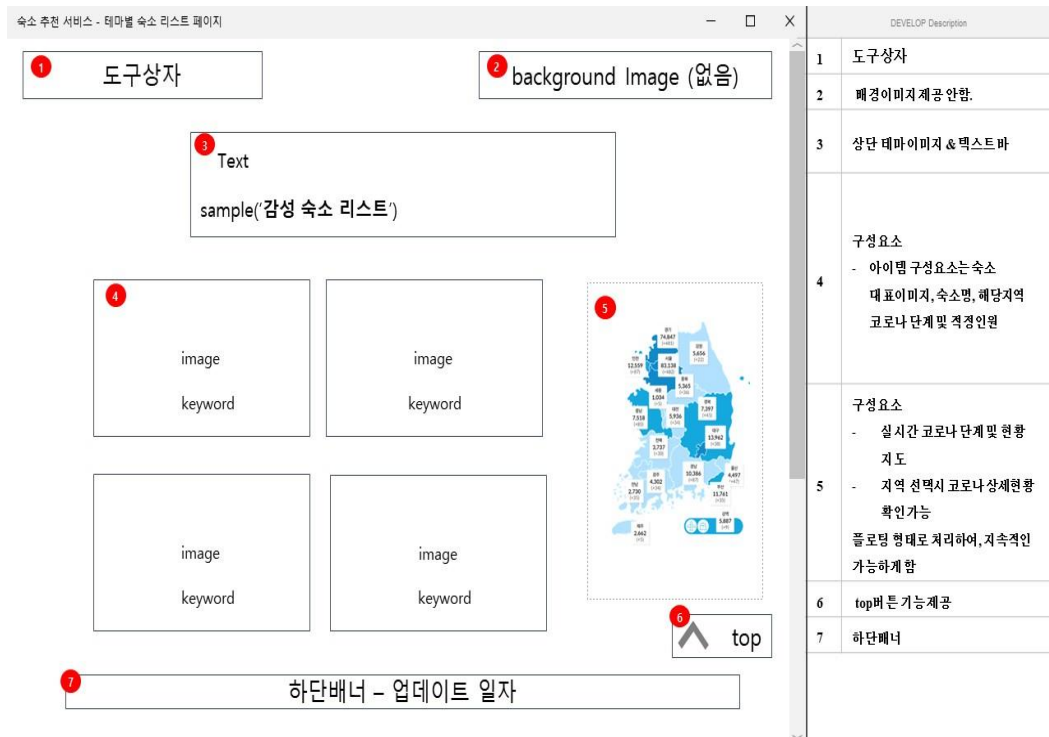
Outcomes : 요구사항 명세서 & 유즈케이스 다이어그램

RQ-ID	화면명	요구사항명	요구사항 내용	날짜	진행사항	비고
RQ-0001	관리자	통계	코로나 API 및 숙소리스트를 갱신 및 점검	9-7	반영	
RQ-0002	메인배너(고정)	좌측 상단 영역	[화면이동] 처음 화면으로 이동, 인기숙소 둘러보기, 코로나 안전정보	9-14	반영	
RQ-0003	하단배너(고정)	좌측	정보 업데이트 일자/ 주기 안내	9-14	미반영	
RQ-0101	인트로	화면 개설	메인배너, 하단배너 고정	9-14	반영	
RQ-0102	인트로	언어	한국어	9-14	반영	
RQ-0103	인트로	가운데 영역	사용자가 여행 유형(키워드)를 선택/ 선호 유형 없을 시 '인기 숙소' 키워드 선택	9-7	제외	
RQ-0104	인트로	로그인	사용자 구별을 위한 로그인 기능 제공	10-5	반영	
RQ-0105	인트로	계정 생성	신규 사용자를 위한 계정 생성 기능 제공	10-5	반영	
RQ-0106	인트로	재로그인	계정 오입력 시 재로그인을 위한 안내 기능 제공	10-5	반영	
RQ-0201	공통	배너	메인, 하단 배너	9-14	반영	
RQ-0202	공통	오른쪽 영역	우측에 코로나 지역별 단계 지도 및 최근 본 매물을 floating 형식으로 제공	10-5	일부반영	확인 아이템 제외
RQ-0203	공통	필터	지역, 인원, 숙소유형 등 요소를 선택 > 지역 선택만 가능	9-14	일부반영	지역선택
RQ-0206	공통	추천 매물 개수	상단에 추천 리스트의 개수를 표시	9-7	제외	
RQ-0207	공통	숙소 정보	숙소명, 숙소 이미지, 숙소 위치, 숙소 가격을 제공 > 숙소 가격은 제외	9-14	반영	
RQ-0208	공통	숙소별 선호 표시	숙소 이미지에 좋아요 표시(클릭)	9-7	제외	
RQ-0301	인기숙소	배너	메인, 하단 배너	9-14	반영	
RQ-0302	인기숙소	오른쪽 영역	우측에 코로나 지역별 단계 지도 및 최근 본 매물을 floating 형식으로 제공	9-14	화면이동	
RQ-0303	인기숙소	필터	카테고리, 지역, 인원, 숙소유형 등 요소를 선택	9-14	일부반영	
RQ-0304	인기숙소	추천 매물 개수	상단에 추천 리스트의 개수를 표시	9-14	제외	
RQ-0305	인기숙소	숙소 정보	숙소명, 숙소 이미지, 숙소 위치, 숙소 가격을 제공	9-14	반영	
RQ-0306	인기숙소	숙소별 선호 표시	숙소 이미지에 좋아요 표시(클릭)	9-14	제외	
RQ-0307	인기숙소	좋아요 개수 표시	인스타그램 데이터를 바탕으로, 숙소별 선호도(좋아요)정보를 제공하고, 이를 기준으로 우선순위 배치	10-5	반영	
RQ-0308	인기숙소	숙소 상세페이지 연결	아이템 버튼을 통해 해당 숙소의 상세정보 페이지로 연결	10-5	반영	
RQ-0401	상세 페이지	배너	메인, 하단 배너	9-14	반영	
RQ-0402	상세 페이지	형태	숙소에 대한 상세한 정보를 제공	9-7	반영	
RQ-0403	상세 페이지	상세 설명	숙소명, 숙소 이미지, 위치, 가격, 인원, 코로나관련 정보, 지역 여행수요, 링크(논의 필요) 등을 제공	9-7	일부반영	
RQ-0404	상세 페이지	안전여행 가이드	코로나 안전여행을 위한 가이드라인을 제공(http://mcov.mohw.go.kr/socdisBoardView.do?brdid=6&brdGubun=1)	9-7	반영	



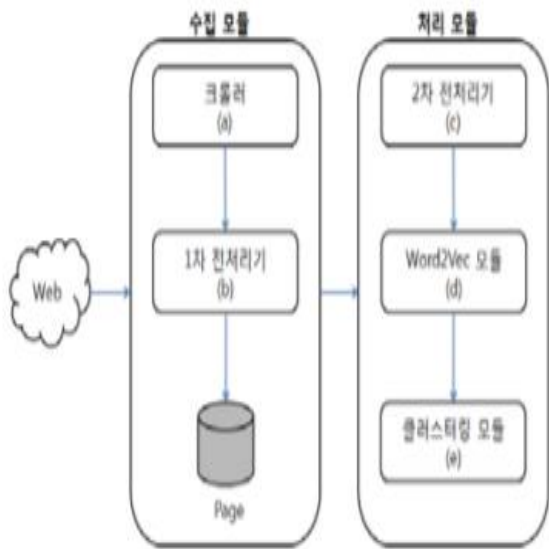
01 Projects

Outcomes : 화면 설계서 및 실제 화면

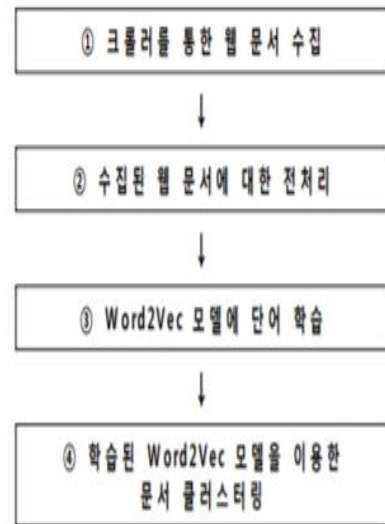


01 Projects

Outcomes : 분석 아키텍처



(그림 2) 시스템 구성도

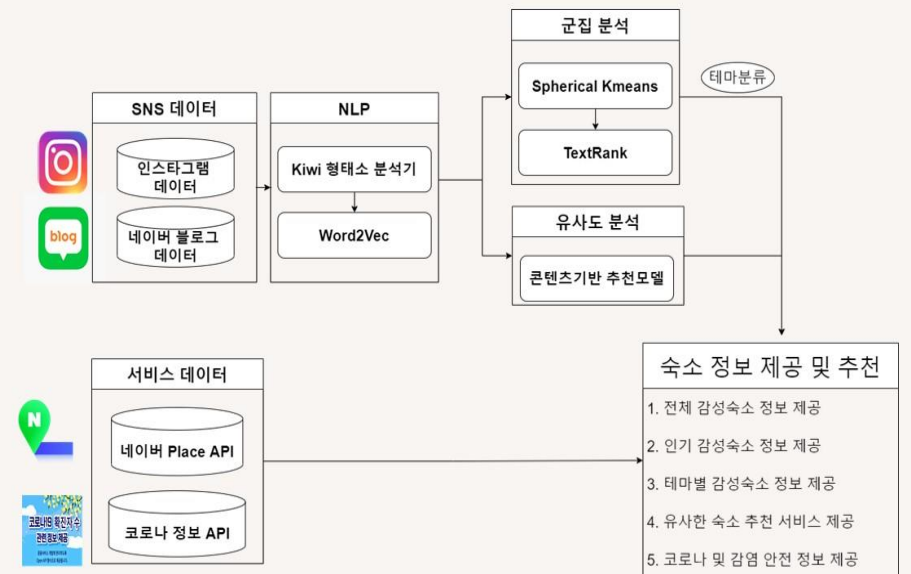


(그림 3) 전체 시스템 순서도

참고 아키텍처

출처 : 이현석 외 5인. "Word2Vec을 이용한 웹 문서 클러스터링 시스템 구현." 2016년 추계학술발표대회 논문집 제23권 제2호, 정보처리학회, 2016

분석 프로세스

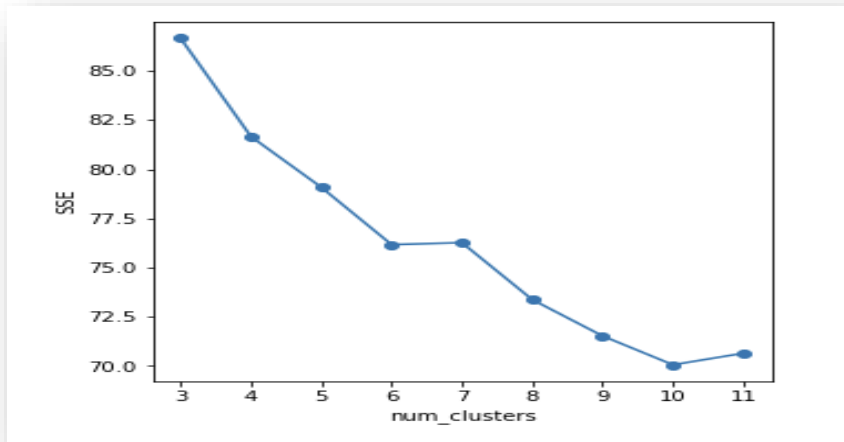


실제로 구현된 아키텍처

01 Projects

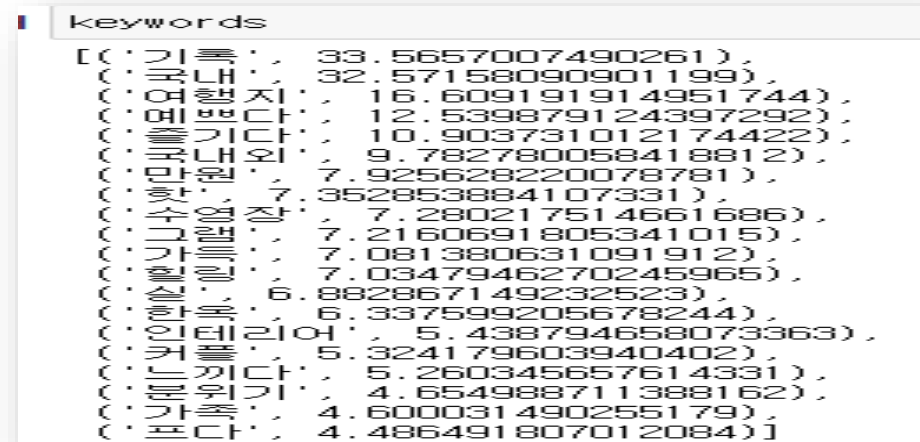
Outcomes : 군집분석 및 토픽모델링

Elbow를 통한 최적 군집수 도출



- ✓ 6과 10 지점에서 두번의 경사 변화가 나타남
- ✓ 10 이후로는 SSE의 변동성이 높음.
- ✓ 최적 군집수를 10으로 설정

TextRank적용 토픽 모델링



- ✓ 문서 요약에 대표적인 기법인 TextRank를 활용하여 군집별 키워드 비율을 분석하고, 이를 토대로 군집별 카테고리 해석을 진행함.
- ✓ 예 : 커플, 가족, 즐거다, 수영장 = '다같이 놀기 좋은 숙소'

02 Projects

머신러닝을 이용한 서울 아파트 실거래가 예측

기여도



Data

- ✓ 국토 교통부 실거래가 데이터
 - ✓ 거래가격, 일자, APT명, 면적, 층수 등
- ✓ 전국 병원 리스트(주소 정보 활용)
- ✓ 서울 지하철 행정동 정보(역 명, 행정동명 등)

Environment

- ✓ Python 3.6+ / Tableau
- ✓ Pandas / Seaborn / Meplotlib

Comment

초기엔 기존 변수만으로 분석결과를 얻으려 했었지만, 멘토님의 조언을 듣고 파생변수를 설정해볼 수 있었고 그 결과 더 유의미한 결과를 얻을 수 있었다. 이를 통해 파생 변수처럼 데이터를 다양한 각도로 바라볼 줄 아는 힘이 중요함을 체감할 수 있었던 프로젝트였다.

Purpose

- ✓ 서울시 아파트 거래 밀집 지역 및 실거래가를 예측
- ✓ 투기성 부동산 거래에 대한 규제 및 대책 형성에 일조

Methodology

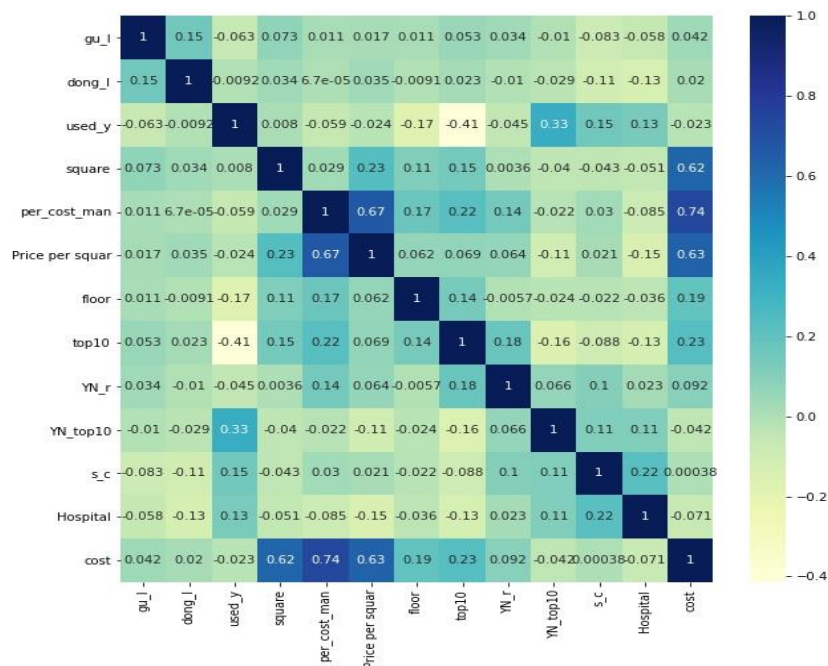
- ✓ 헤도닉 가격모형을 베이스로 하여 연구모형 구축
- ✓ 다양한 ML알고리즘을 시도하여, 최적의 알고리즘 도출
 - 👉 KNN, DT, RF, GBM, LGBM, Xgboost 비교
 - 👉 모델 성능은 MSE & RMSE를 통해 비교

02 Projects

머신러닝을 이용한 서울 아파트 실거래가 예측

Processing

피어슨 상관계수를 통한 상관관계 시각화



Price per square(구별 평당 공시지가)

Per_cost(해당 아파트 평당 가격)

Square(전용 면적)

Top10(아파트 브랜드)

Floor(층수)

5개 변수와 cost의 양의 상관관계를 확인

02 Projects

머신러닝을 이용한 서울 아파트 실거래가 예측

Processing



각 모델 별 MSE & RMSE 시각화

XGB > DT > RF > KNN > GBM > LGBM

순으로 모델 성능 확인

평균 오차(MSE)가

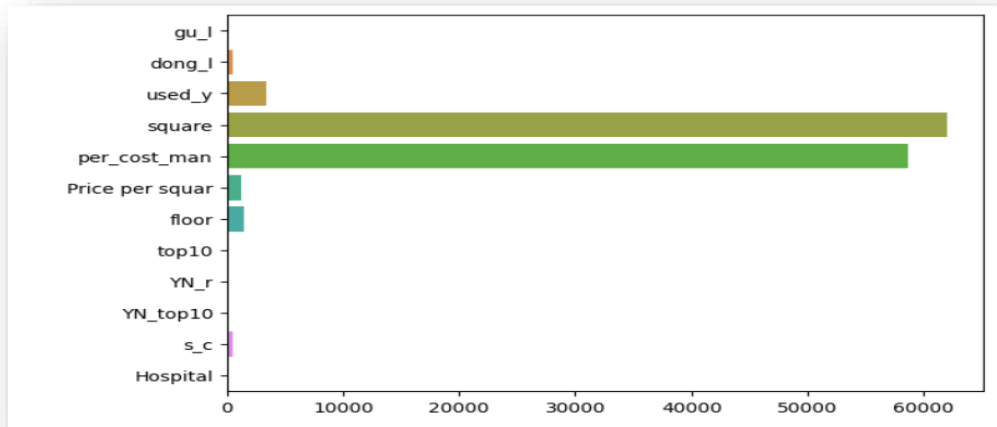
가장 낮게 나오는

LGBM 모델 선정

02 Projects

머신러닝을 이용한 서울 아파트 실거래가 예측

Processing



```
lgb.feature_importances_  
array([[ 88,  566, 3361, 62039, 58621, 1239, 1521,  51, 168,  
        78,  483,  0], dtype=int32)
```

Feature Selection : Feature Importance 이용

```
# 하이퍼 파라미터 튜닝  
%time  
from sklearn.model_selection import GridSearchCV  
  
params = {  
    'learning_rate' : [0.1, 0.01, 0.001, 0.0001],  
    'max_depth' : [1,2,3,4]  
}  
  
grid_cv = GridSearchCV(lgb, param_grid = params, cv=4, scoring='neg_mean_squared_error', verbose=1)  
grid_cv.fit(X_train, y_train)  
  
CPU times: user 3 µs, sys: 0 ns, total: 3 µs  
Wall time: 5.72 µs  
Fitting 4 folds for each of 16 candidates, totalling 64 fits  
  
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.  
[Parallel(n_jobs=1)]: Done 64 out of 64 | elapsed: 25.8min finished  
  
GridSearchCV(cv=4,  
             estimator=LGBMRegressor(learning_rate=0.001, linear_tree=True,  
                                     max_depth=4, n_estimators=9000,  
                                     objective='regression'),  
             param_grid={'learning_rate': [0.1, 0.01, 0.001, 0.0001],  
                         'max_depth': [1, 2, 3, 4]},  
             scoring='neg_mean_squared_error', verbose=1)
```

최적의 하이퍼 파라미터 : {'learning_rate': 0.1, 'max_depth': 4}
예측 오차 : -140939.36042289424

Model tuning : GridSearch CV 적용

02 Projects

머신러닝을 이용한 서울 아파트 실거래가 예측

Processing

```
lgb_r = LGBMRegressor(linear_tree = True,  
                       boosting_type = 'gbdt',  
                       objective = 'regression',  
                       n_estimators = 9000,  
                       learning_rate = 0.1,  
                       max_depth = 4,  
                       n_jobs = -1)
```

```
lgb_r.fit(X_train, y_train)
```

CPU times: user 2 μ s, sys: 0 ns, total: 2 μ s
Wall time: 5.25 μ s

MSE : 537552.7584483256
RMSE : 733.1798950109895

RMSE가 733(만원) 정도로 오차가 파라미터 튜닝 전보다 절반정도로 줄었다.

RMSE 결과값 변화 추이(단위 : 1만원)

<u>초기 Dataset</u>	1차 하위변수 제거	2차 하위변수 제거	<u>파라미터 튜닝 후</u>
<u>1352</u>	1352	1355	<u>733</u>

튜닝한 하이퍼 파라미터를 통해서 test 데이터 확인

```
pred[0]
```

62496.679686278745

```
y_test.iloc[0]
```

62500

예측값 = 62496.68
실제값 = 62500

03 Projects

2020 코로나 확산으로 인한 경제적 손실 분석 : 이태원 상권을 중심으로 기여도



Data

- ✓ 생활인구 특성 데이터 : 유동인구
- ✓ 상권 특성 데이터 : 추정 매출액
- ✓ 기타 특성 데이터 : 코로나 확진자, 폐업/공실률

Environment

- ✓ Python 3.6+ / Jupyter / Tableau
- ✓ Pandas / MySQL / Scikit-learn / Plotly

Background

- ✓ 코로나 이후 자영업계에 큰 손실이 발생되고 있음
- ✓ But 정부가 제안하는 자영업자 지원정책은 제한적임

Purpose

- ✓ 코로나 이후 개별 상권 자영업 손실 특성을 분석.
- ✓ 상권 특성을 반영한 개선된 지원기준을 제시하고자 함.

Methodology

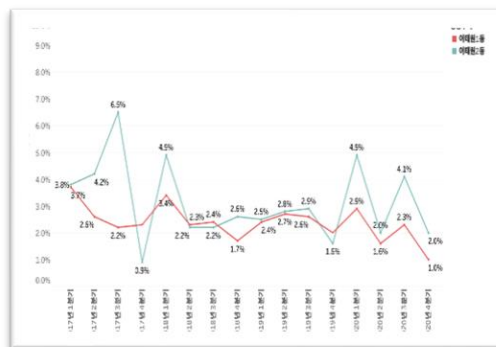
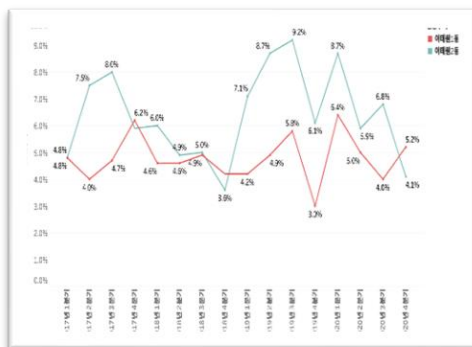
- ✓ 대표성, 위험성, 개성을 기준으로 타겟 지역 선정
 - ↳ 이태원 지역 선정(Gentrification)
- ✓ 매출 증감 분석 ↳ 상권별, 업종별로
- ✓ 업종, 상권 군집분석 ↳ K-means 적용

03 Projects

2020 코로나 확산으로 인한 경제적 손실 분석 : 이태원 상권을 중심으로

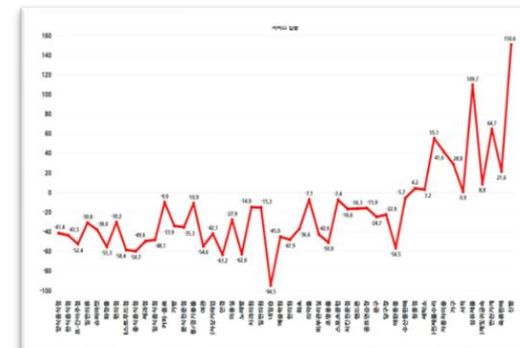
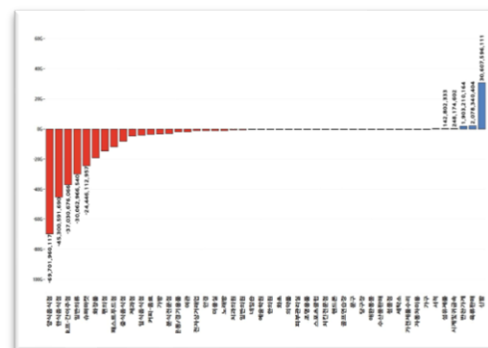
Issue 1 : 폐업률

- ✓ 코로나 이후 폐업률이 지속적으로 증가하지 않음
- ✓ 폐업률이 증가했을 것이란 최초 가설 성립 ✕



Solution 1 : 매출증감액 및 증감률 분석

- ✓ 예상과 달리, 일부 업종의 매출액 증가 확인
- ✓ 업종별 특성 구분의 필요성 도출

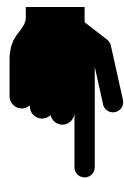


03 Projects

2020 코로나 확산으로 인한 경제적 손실 분석 : 이태원 상권을 중심으로

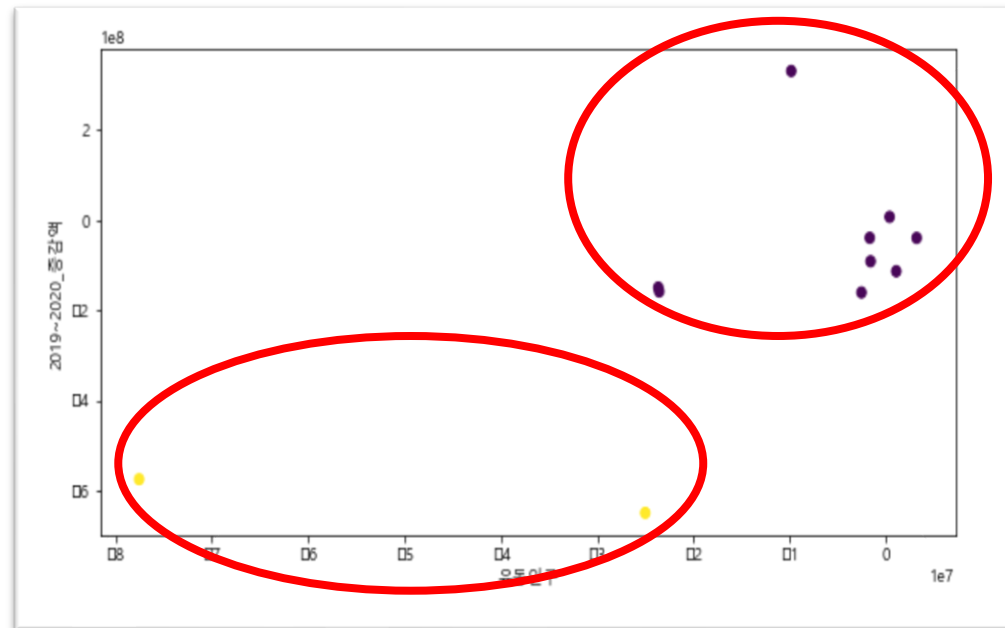
Issue 2 : 군집 수 설정 기준

- ✓ K-means 기법의 고질적인 한계이기도 한 군집수 설정
- ✓ 엘보우 & 실루엣 기법 > 상권별 업종 특성 반영 X



Solution 2 : 공신력 있는 기준 활용

- ✓ 서울신용보증재단의 코로나 상권 분류 기준
- ✓ 공신력있는 기관의 기준을 활용하여 상권 분류



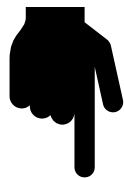
전년도 대비 매출 증감액과 유동인구 기준
이태원 내 11개 지역의 군집분석 결과

03 Projects

2020 코로나 확산으로 인한 경제적 손실 분석 : 이태원 상권을 중심으로

Issue 3 : 기존 정책과의 차별성


- ✓ 기존 정책의 한계점을 극복할 수 있는 새로운 방법
- ✓ 기존 정책의 한계점 : 상권 특성과 업종 특성을 반영 X




Solution 3 : 포인트제를 도입한 새로운 지급 기준 제시

- ✓ 업종과 상권을 모두 기준으로 할 수 있음
- ✓ 각 영역 별 점수를 합산하여 지원금을 산정
- ✓ 새로운 특성을 반영하기에도 용이하다는 장점

	업종	상권
1순위(2p)	고위험군	충격상권
2순위(1p)	중위험군	선방상권
3순위(0p)	저위험군	

 **활용 예시 - 나의 지원금은?**



이태원역 인근에서
슈퍼마켓을 하고 있는
자영업자 A

지원금 알아보기

+1 이태원역 인근
+2 슈퍼마켓

합계 : 3pts
지원금 : 250만원



THANK YOU

