

목차 A table of contents

1 프로젝트 개요

2 프로젝트 팀구성 및 역할

3 프로젝트 수행 절차 및 방법

4 프로젝트 수행 결과

5 자체 평가 의견





프로젝트 개요

- 프로젝트 주제 및 선정 배경
- 개발레퍼런스
- 개발환경
- 개발과정

프로젝트 개요 프로젝트 주제 및 선정 배경

서울 등 대도시 지역에서 젠트리피케이션 부작용이 사회적 이슈로 대두

- 지역 특성화 및 활성화에 따른 급격한 임대료 상승과 주거환경 악화 등의 문제로 기존 주민과 영세 상인이 내몰리게 됨

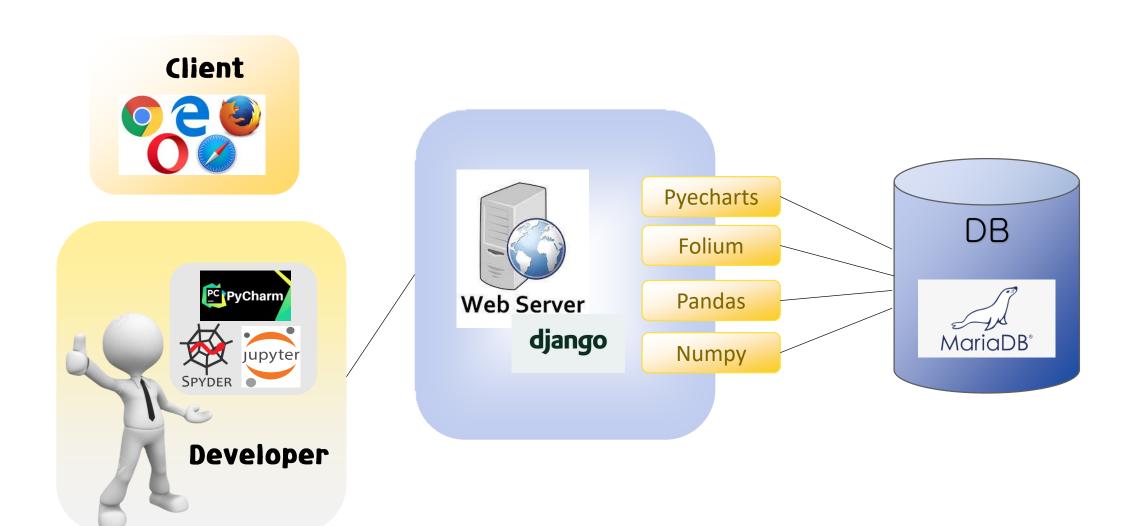
- 본 프로젝트의 목적은 객관적인 자료를 통하여 젠트리피케이션 발생 여부 및 단계를 정량화된 수치로 진단할 수 있는 '젠트리피케이션 지표'를 개발하는 것

1 프로젝트 개요 개발 레퍼런스

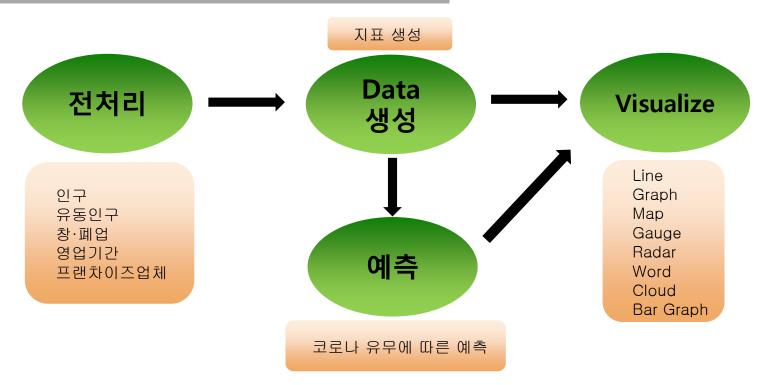


< 젠트리피케이션 지표 하위변수 선정 >

- 레퍼런스에 주어진 6개의 지표 중 인구, 창폐업횟수, 영업기간, 프랜차이즈 업체 수, 유동인구에 대한 데이터를 사용.
 → 가중치를 재조정하여 지표 구축



1 프로젝트 개요 개발 과정



- 코로나 영향을 적용한 분석 및 예측

2017	2018	2019	2020	2021 1/4	2021 2/4	2021 3/4	2021 3/4	2022 1/4
분석					예측			

- 코로나 영향을 적용하지 않은 분석 및 예측

2017	2018	2019	2020	2021 1/4	2021 2/4	2021 3/4	2021 3/4	2022 1/4
분석		예측						

프로젝트 팀 구성 및 역할





1조 팀 구성 및 역할

훈련생	역할	담당 업무
김민선	팀장	▶ 외부 데이터 수집, 정제 및 시각화▶ 데이터 예측
김유림	팀원	▶ 외부 데이터 수집, 정제 및 시각화▶ 웹 개발, DB구축
김재현	팀원	▶ 외부 데이터 수집, 정제 및 시각화▶ 발표자료 준비
김현섭	팀원	▶ 외부 데이터 수집, 정제 및 시각화▶ 웹 개발



프로젝트 수행 절차 및 방법

3 프로젝트 수행 절차 및 방법



프로젝트수행기간및과정

구분	기간	활동	비고
기획 및 데이터 수집	▶ 10/9(星) ~ 10/15(금,	▶ 주제 및 데이터 선정▶ 분석내역 및 기능목록 정리▶ 외부 데이터 수집	▶ 아이디어 선정 ▶ 서울시 열린데이터 광장 참조
설계 및 개발	▶ 10/16(星) ~ 10/22(금)	▶ 테이블, 화면, 프로그램, 아키텍처 정의 ▶ 데이터 전처리	
서비스 구축	▶ 10/23(월) ~ 10/29(금)	▶ 데이터 분석 및 예측▶ DB 구축 및 웹 개발	
정리	▶ 10/30(토) ~ 11/3(수,	▶ 오류 수정 및 개선 작업▶ 발표자료 준비	<i>> 총기간 26일</i>

프로젝트 수행 결과

- 데이터 전체리
- 분석 및 예측
- 시각화



프로젝트 수행 결과 결과 제시 ① 데이터 전처리

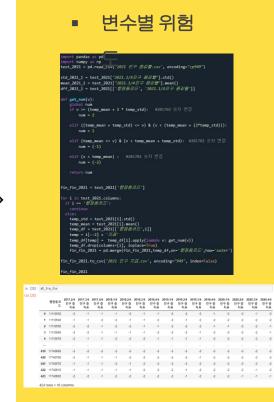


수집한 데이터를 전처리 및 계산

- 다양한 매체로부터 데이터를 수집 후 이를 필요한 방향으로 정제
- 측정에 필요한 하위 변수를 계산
- 젠트리피케이션 단계 구분을 위한 지표 구축









4 프로젝트 수행 결과 결과 제시 ② 분석 및 예측

- 5개의 동을 대표로 선정하여 예측을 진행함.

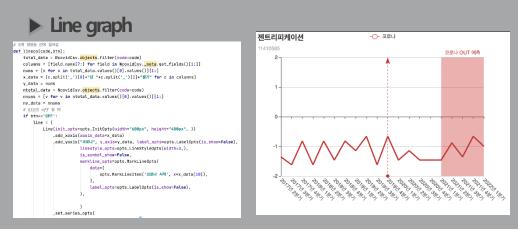
LSTM을 사용하여 5개의 변수를 예측하고 예측한 값을 토대로 지표를 선정함.

```
model = Sequential()
                                                                                                                      fc
                                                                                                           year
model.add(LSTM(self.LSTM_layer_depth, activation='<mark>relu</mark>', input_shape=(self.lag, 1)))
model.add(Dense(1))
                                                                                                    206 2017 / 2 19.000000
model.compile(optimizer='adam', loss='mse')
                                                                                                    581 2017 / 3 19.000000
deep_learner = DeepModelTS(
                                                                                                    956 2017 / 4 18.000000
data = df.
                                                                                                   1331 2018 / 1 19.000000
Y var = "fc".
lag =10.
                                                                                                   1706 2018 / 2 19.000000
LSTM_layer_depth = 100,
                                                                                                   2081 2018 / 3 19 000000
epochs = 100.
batch_size = 10,
                                                                                                   2456 2018 / 4 18.000000
train_test_split = 0.1
                                                                                                   2831 2019 / 1 18.000000
                                                                                                   3206 2019 / 2 18.000000
# Fitting the model
                                                                                                   3581 2019 / 3 19.000000
model = deep_learner.LSTModel()
                                                                                                   3956 2019 / 4 21.000000
0 2020 / 1 21.131948
1/1 [=======================] - Os 39ms/step - loss: 0.0065 - val_loss: 0.0019
Epoch 93/100
                                                                                                      1 2020 / 2 21.211412
                     ======] - Os 35ms/step - loss: 0.0019 - val_loss: 1.3411e-05
                                                                                                      2 2020 / 3 21.451483
1/1 [==================] - Os 39ms/step - loss: 1.3411e-05 - val_loss: 0.0011
Epoch 95/100
                                                                                                      3 2020 / 4 21.594770
4 2021 / 1 21.784111
1/1 [=========================] - Os 37ms/step - loss: 0.0037 - val_loss: 0.0062
Epoch 97/100
                                                                                                      5 2021 / 2 22.062078
                          =1 - Os 36ms/step - Loss: 0.0062 - val Loss: 0.0071
Epoch 98/100
                                                                                                      6 2021/3 22.563749
1/1 [=========================] - Os 49ms/step - loss: 0.0071 - val_loss: 0.0062
Epoch 99/100
                                                                                                      7 2021 / 4 23.135244
8 2022 / 1 23.745970
1/1 [========================] - ETA: Os - loss: 0.004 - Os 53ms/step - loss: 0.0040 - val_loss: 0.0016
```

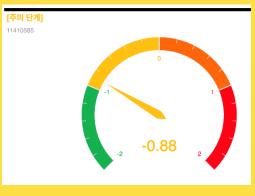
프로젝트 수행 결과

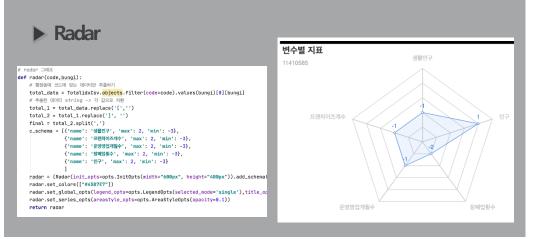
결과 제시 ③ 시각화









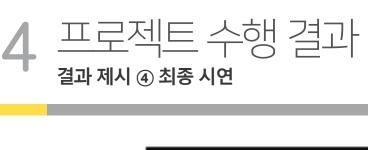


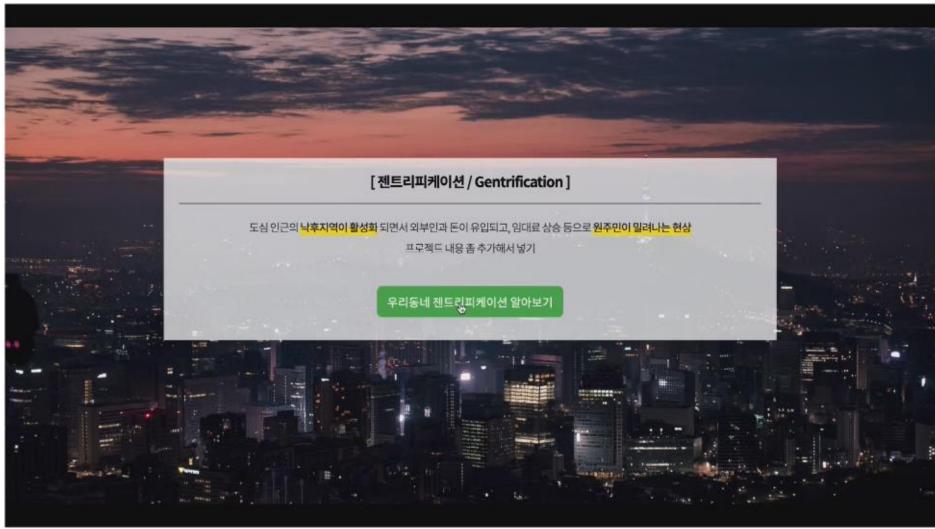
4 프로젝트 수행 결과 결과 제시 ③ 시각화

World Cloud | def worl(gu,bungl): | down_name = [[('coder)' for i in Area.objects.filter(gung),values()] | | down_name = [[('coder)' for i in Area.objects.filter(gung)_inades),values()] | | down_name = [[('coder)' for i in Area.objects.filter(com__inedom_names),values()] | | down_name = [[('coder)' for i in Area.objects.filter(com__inedom_names),values()] | | down_name = [[('coder)' for i in Area.objects.filter(com__inedom_names),values()] | | down_name = [[('coder)' for i in Area.objects.filter(com__inedom_names),values()] | | down_name = [[('coder)' for i in Area.objects.filter(com__inedom_names),values()] | | down_name = [[('coder)' for i in Area.objects.filter(com__inedom_names),values()] | | down_name = [[('coder)' for i in Area.objects.filter(com__inedom_names),values()] | | down_name = [[('coder)' for i in Area.objects.filter(com__inedom_names),values()] | | down_name = [[('coder)' for i in Area.objects.filter(com__inedom_names),values()] | | down_name = [[('coder)' for i in Area.objects.filter(com__inedom_names),values()] | | down_name = [[('coder)' for i in Area.objects.filter(com__inedom_names),values()] | | down_name = [[('coder)' for i in Area.objects.filter(com__inedom_names),values()] | | down_name = [[('coder)' for i in Area.objects.filter(com__inedom_names),values()] | | down_name = [[('coder)' for i in Area.objects.filter(com__inedom_names),values()] | | down_name = [[('coder)' for i in Area.objects.filter(com__inedom_names),values()] | | down_names = [[('coder)' for i in Area.objects.filter(com__inedom_names),values()] | | down_names = [[('coder)' for i in Area.objects.filter(com__inedom_names),values()] | | down_names = [[('coder)' for i in Area.objects.filter(com__inedom_names),values()] | | down_names = [[('coder)' for i in Area.objects.filter(cod__inedom_names),values()] | | down_names = [[('coder)' for i in Area.objects.filter(cod__inedom_names),values()] | | down_names = [[('coder)' for i in Area.objects.filter(cod__inedom_names),values() | | down_names = [[('coder)' for i in Area.











자체 평가 의견

5 자체 평가 의견

멤버	평가 의견 및 느낀점
김민*	데이터를 구하고 전처리하는 부분에서 많은 시간이 소모되어 개발부분에 투자한 시간이 적은 것 같아 아쉬웠습니다. 그래도 팀원들과 원활한 의사소통과 협업을 통해 프로젝트를 무사히 마칠 수 있었습니다. 또한 이번 프로젝트를 통해 시계열예측 모델에 대해 학습하고 다루어 볼 수 있는 좋은 경험이 되었습니다. LSTM을 통해 예측을 진행하였지만 향후에는 ARIMA, Facebook Prophet, Transformer 등의 다양한 시계열분석 모델들을 학습하여 더욱 더 적합한 모델을 구축해보면 좋을 것 같습니다.
김유*	지난 프로젝트와는 다른 프레임 워크를 사용해 새롭게 알아가는 부분이 많아서 재미있었습니다. 초반에 데이터 수집이나 예측 모델을 개발하는 과정에서 어려움을 느꼈는데 팀원들의 도움 덕분 에 프로젝트를 기한 내에 무사히 마무리 지을 수 있었습니다.

멤버

평가 의견 및 느낀점

김재*

필요한 몇몇 데이터들은 개인 정보 관련으로 공개된 데이터를 구할 수 없어 정확도를 높이지 못한 점이 아쉬 웠습니다. 그럼에도 팀원들 모두가 본인이 적극적으로 소통을 하며 맡은 부분에 책임감을 갖고 최선을 다하여 결과를 이끌어냈기에 아쉬움 속에서도 만족감을 느낄 수 있었습니다.

김현*

금융 관련 데이터 정보를 수집하는 데 한계가 있어서 프로젝트 주제를 정하는데 제한사항이 많았지만 적극적인 팀원들과 소통하며 프로젝트를 진행하는데 무리 없이 마무리할 수 있었습니다.
LSTM 모델로 예측하기에는 가진 데이터의 수가 적어서 정확한 평가를 하기에는 아쉬운 점들이 많았고 추후 ARMA, ARIMA 모델 등을 사용하는 개선이 필요할 것 같습니다.
이번 빅데이터 분석 프로젝트를 진행하면서 데이터를 정제하고 전처리 및 분석하는 방법을 얻을 수 있는 좋은 계기가 되었고 장고 웹 개발을 통해 MVT 구조를 이해하고 데이터 시각화를 구현할수 있게 되었습니다.

PowerPoint



