



최종발표

오전반 3조 – 김선우, 임시은, 김한주, 박지우

Table of Contents

01



프로젝트 소개

02



상황 분석

03



아이디어 제시

04

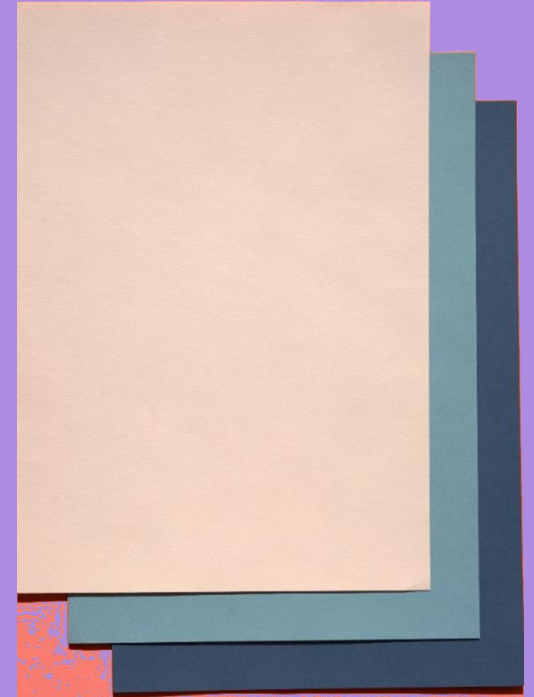


서비스 아키텍처

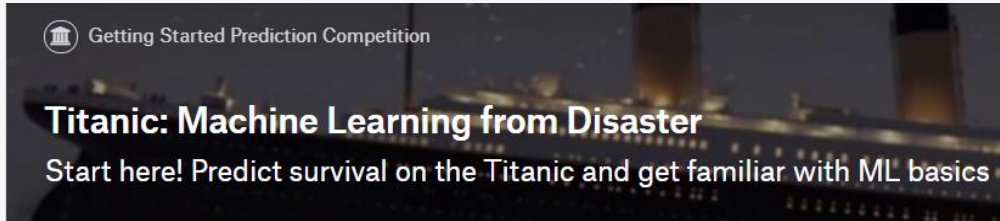
05



예측 모델링 구현




프로젝트 준비



Getting Started Prediction Competition

Titanic: Machine Learning from Disaster


Start here! Predict survival on the Titanic and get familiar with ML basics



Bike Sharing Demand

Forecast use of a city bikeshare system

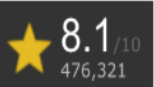
3,251 teams · 4 years ago



House Prices: Advanced Regression Techniques

Predict sales prices and practice feature engineering, RFs, and gradient boosting

4,336 teams · Ongoing



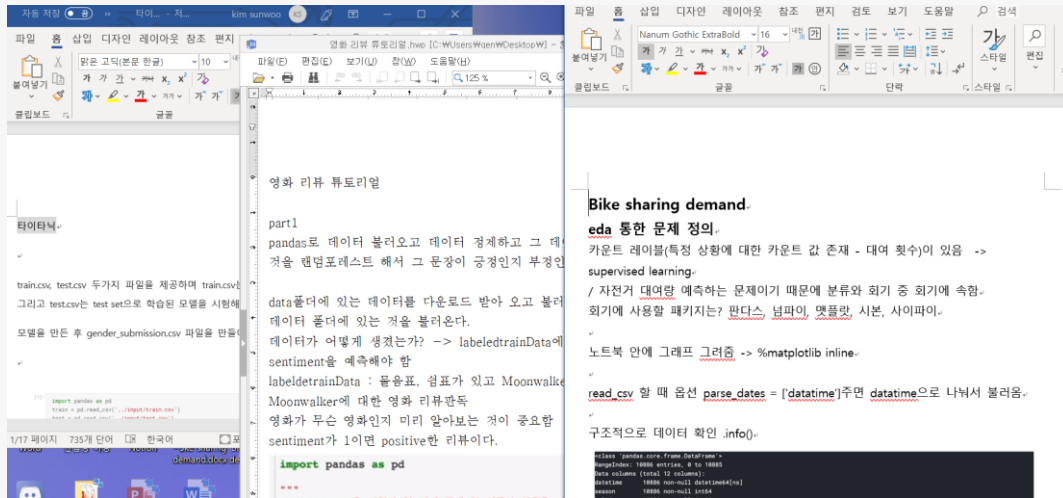
Sentiment analysis on IMDB movie reviews

Determine whether a movie review is positive or negative

55 teams · 2 years ago

구글 예측 모델 및 분석 대회 플랫폼인 kaggle에서 tutorial 과정을 진행하기로 결정하였습니다.

프로젝트 준비



영화 리뷰 긍정/부정 튜토리얼

1. 데이터 확인

- Data index 및 column 파악
- 특수 문자 및 HTML 태그 파악
- 눈으로 핵심 단어 확인

2. 데이터 정제 및 전처리

- BeautifulSoup을 통해 HTML 태그 제거
- 정규표현식으로 문자 공백 치환
- NLTK 데이터 사용으로 불용어 제거
- 어간추출, 음소표기법

3. 모델 설계 및 평가

- 사이킷런의 CountVectorizer
- 랜덤포레스트 분류기

타이타닉 튜토리얼

1. 데이터 확인

- Data index 및 column 파악
- NULL값 데이터 파악
- 그래프를 통하여 데이터 분석

2. 데이터 정제 및 전처리

- 연관성 없는 데이터는 삭제
- Data type을 숫자로 변환
- 그래프 분석을 통하여 데이터 정제
- Null값은 평균을 내준 후 평균

3. 모델 설계 및 평가

- 랜덤포레스트 분류기(랜덤(randomness)에 의해 트리들이 서로 조금씩 다른 특성을 갖는다는 점을 이용)
- 80퍼센트정도의 정확성

자전거 대여 예측 튜토리얼

1. 데이터 확인

- Data index 및 column 파악
- Data Null값 파악

2. 데이터 정제 및 전처리

- 날짜를 시,분,초 단위로 새로운 column
- 물려있는 데이터 아웃라이어 처리
- One-hot encoding, Feature select

3. 모델 설계 및 평가

- RMSE 평가 방식 사용
- RMSLE 손실 함수
- 교차 검증(K-Fold)
- 랜덤포레스트 분류기
- 선형 회귀 모델

집값 예측 튜토리얼

1. 데이터 확인

- 그래프로 outlier, distrib 확인
- Train과 test 데이터를 확인

2. 데이터 정제 및 전처리

- GrLivArea를 통해 outlier 확인 및 제거
- Train data와 test data를 concatenate
- 데이터에 따라 drop, fillna를 적절히 사용

3. 모델 설계 및 평가

- 중요한 features는 더 추가
- Lasso, Elastic Net, Kernel Ridge 등 다양한 회귀 모델을 사용한 후 앙상블
- K-fold 분류기

각자 Kaggle tutorial 진행 및 분석 후 워드로 정리하여 서로에게 발표하고 공유하였습니다.

프로젝트 소개



문제

빅데이터를 활용한 “미세먼지의 사회적 영향 분석 및 비즈니스 아이디어 제시”

- 유동인구데이터(SK텔레콤), 카드매출데이터(신한카드), SNS데이터(와이즈넷), 환경기상데이터(케이웨더), 유통데이터(GS리테일), 공공데이터 등 다양한 데이터를 활용하여 미세먼지로 인한 소비/경제/행동변화에 따른 **사회적 영향 분석 및 예측 모델링**을 통한 비즈니스 아이디어 제시

다양한 데이터를 직접 다뤄 볼 수 있다는 점, 아이디어를 제시하고 구현한다는 점에서 innovation을 선택하였습니다.

프로젝트 소개




2019 빅콘테스트
2019 BIG CONTEST





Innovation 분야


| 유동인구데이터 | 카드매출데이터 | SNS데이터 | 환경기상데이터 | 유통데이터 |
|---------|---------|-----------|-------------|------------|
| 기준년도 | 기준일자 | 문서 KEY값 | 데이터측정 날짜 시간 | 영업일자 |
| 법정동_코드 | 구코드 | 문서 등록일 | 측정기 고유번호 | 구코드 |
| 법정동_명칭 | 법정동코드 | 블로그 카페 뉴스 | 실외 측정기 구분 | 법정동코드 |
| 시도_코드 | 업종코드 | | 미세먼지 PM-10 | 매출지수 |
| 시도_명칭 | 성별코드 | | 이산화탄소 농도 | 식사_비중 |
| 시군구_코드 | 나이코드 | 문서 제목 | 휘발성유기화합물 농도 | 간식_비중 |
| 시군구_명칭 | 이용건수 | 문서 본문 | 소음 데이터(db) | 마실거리_비중 |
| 길이 | 이용금액 | | 온도(°C) 데이터 | 홈&리빙_비중 |
| 면적 | | | 습도(%) 데이터 | 헬스&뷰티_비중 |
| X_좌표 | | | 미세먼지 PM-2.5 | 취미&여가활동_비중 |
| Y_좌표 | | | | 사회활동_비중 |
| | | | | 임신/육아_비중 |
| | | | | 기호품_비중 |


데이터 전처리


 SNS_1.xlsx


 SNS_2.xlsx


 SNS_3.xlsx


 SNS_4.xlsx


 SNS_5.xlsx

























 SNS_6.xlsx

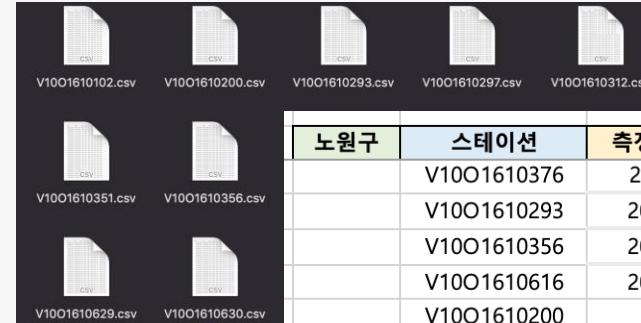
 SNS_7.xlsx

 SNS_8.xlsx

 성연령유동 > 지역에 따른 성 및 연령별 유동인구


 시간대유동 > 지역에 따른 시간대별 유동인구

| | |
|---|--|
|  노원_종로_FLOW_AGE_201804.CSV |  노원_종로_FLOW_TIME_201804.CSV |
|  노원_종로_FLOW_AGE_201805.CSV |  노원_종로_FLOW_TIME_201805.CSV |
|  노원_종로_FLOW_AGE_201806.CSV |  노원_종로_FLOW_TIME_201806.CSV |
|  노원_종로_FLOW_AGE_201807.CSV |  노원_종로_FLOW_TIME_201807.CSV |
|  노원_종로_FLOW_AGE_201808.CSV |  노원_종로_FLOW_TIME_201808.CSV |
|  노원_종로_FLOW_AGE_201809.CSV |  노원_종로_FLOW_TIME_201809.CSV |
|  노원_종로_FLOW_AGE_201810.CSV |  노원_종로_FLOW_TIME_201810.CSV |
|  노원_종로_FLOW_AGE_201811.CSV |  노원_종로_FLOW_TIME_201811.CSV |
|  노원_종로_FLOW_AGE_201812.CSV |  노원_종로_FLOW_TIME_201812.CSV |
|  노원_종로_FLOW_AGE_201901.CSV |  노원_종로_FLOW_TIME_201901.CSV |
|  노원_종로_FLOW_AGE_201902.CSV |  노원_종로_FLOW_TIME_201902.CSV |
|  노원_종로_FLOW_AGE_201903.CSV |  노원_종로_FLOW_TIME_201903.CSV |



| 노원구 | 스테이션 | 측정기 등록일 | 위치 |
|-----|-------------|------------|------|
| | V1001610376 | 2017.12.27 | 상계동 |
| | V1001610293 | 2017.12.27 | 상계동 |
| | V1001610356 | 2017.12.27 | 상계동 |
| | V1001610616 | 2017.12.23 | 상계2동 |
| | V1001610200 | # | 상계동 |
| | V1001610643 | 2017.12.23 | 월계동 |
| | V1001610642 | 2017.12.22 | 공릉2동 |

 CARD_SPENDING.txt

 GS리테일_동별 매출지수용 기준값 확인_AMT_NEW.xlsx

참고)구_행정동코드

참고)분석용상품대분류코드

동별매출지수

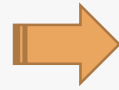
카테고리별 매출비중

종합테이블

실제 주어진 데이터들은 데이터 별로 너무나도 달랐습니다.

데이터 전처리

| HDONG_NM | HDONG_NM |
|----------|----------|
| 청운호자동 | 11110515 |
| 사직동 | 11110530 |
| 삼청동 | 11110540 |
| 부암동 | 11110550 |
| 평창동 | 11110560 |




| STD_YMD | STD_Y | STD_M | STD_D |
|----------|-------|-------|-------|
| 20180401 | 2018 | 4 | 1 |
| 20180401 | 2018 | 4 | 1 |
| 20180401 | 2018 | 4 | 1 |
| 20180401 | 2018 | 4 | 1 |
| 20180401 | 2018 | 4 | 1 |
| 20180401 | 2018 | 4 | 1 |
| 20180401 | 2018 | 4 | 1 |



| | OPER_DT | BOR_CD | ADMD_CD | AMT_IND | LCLS_10_P | LCLS_20_P | LCLS_30_P | # |
|---|------------|-----------|------------|-----------|-----------|-----------|-----------|---|
| 0 | 20180401.0 | 1111.0 | 11110515.0 | 73.7 | 27.9 | 30.0 | 33.5 | |
| 1 | 20180401.0 | 1111.0 | 11110530.0 | 125.8 | 23.7 | 25.8 | 35.0 | |
| 2 | 20180401.0 | 1111.0 | 11110540.0 | 67.4 | 25.4 | 24.7 | 32.4 | |
| 3 | 20180401.0 | 1111.0 | 11110550.0 | 101.1 | 41.6 | 21.8 | 28.3 | |
| 4 | 20180401.0 | 1111.0 | 11110560.0 | 101.3 | 31.1 | 26.7 | 26.2 | |
| | LCLS_40_P | LCLS_50_P | LCLS_60_P | LCLS_70_P | LCLS_80_P | | | |
| 0 | 1.6 | 5.4 | 0.0 | 1.3 | 0.4 | | | |
| 1 | 4.6 | 9.3 | 0.5 | 0.2 | 0.9 | | | |
| 2 | 6.5 | 9.0 | NaN | 1.3 | 0.7 | | | |
| 3 | 2.7 | 3.7 | 0.6 | 1.2 | 0.2 | | | |
| 4 | 1.5 | 13.5 | NaN | 0.7 | 0.2 | | | |

| | STD_DD | GU_CD | MCT_CAT_CD | SEX_CD | AGE_CD | USE_CNT | USE_AMT |
|---|----------|----------|------------|--------|--------|---------|---------|
| 0 | 20180401 | 11110515 | 21 | 1 | 55 | 4 | 22 |
| 1 | 20180401 | 11110515 | 21 | 2 | 20 | 35 | 184 |
| 2 | 20180401 | 11110515 | 21 | 2 | 25 | 70 | 425 |
| 3 | 20180401 | 11110515 | 21 | 2 | 30 | 18 | 82 |
| 4 | 20180401 | 11110515 | 21 | 2 | 35 | 4 | 44 |

각 데이터마다 공통으로 존재하는 부분을 하나의 기준으로 통합하고
데이터 처리를 위해 숫자가 아닌 값들을 전부 int, float의 숫자형으로 변경해 주었습니다

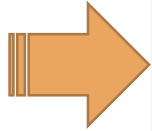


```
In [11]: clf = IsolationForest(random_state=1)
clf.fit(df[['AMT_IND', 'LCLS_10_P', 'LCLS_20_P', 'LCLS_30_P']])
y_pred_outliers = clf.predict(df[['AMT_IND', 'LCLS_10_P']])
out = pd.DataFrame(y_pred_outliers)
out = out.rename(columns={0: "out"})
race_an1 = pd.concat([df, out], 1)
print(race_an1.shape)
```

결측치를 적절히 처리하고, 이상치를 Isolation Forest 알고리즘을 사용해 제거 했습니다.

데이터 전처리

pre_air.csv
Pre_cardData.csv
PRE_Flow.csv
pre_GS2.csv
pre_sns.csv



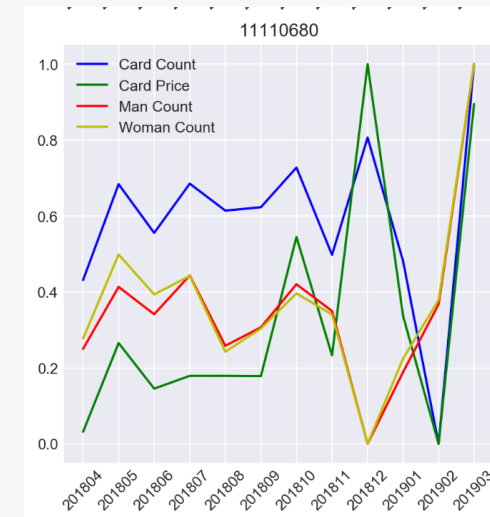
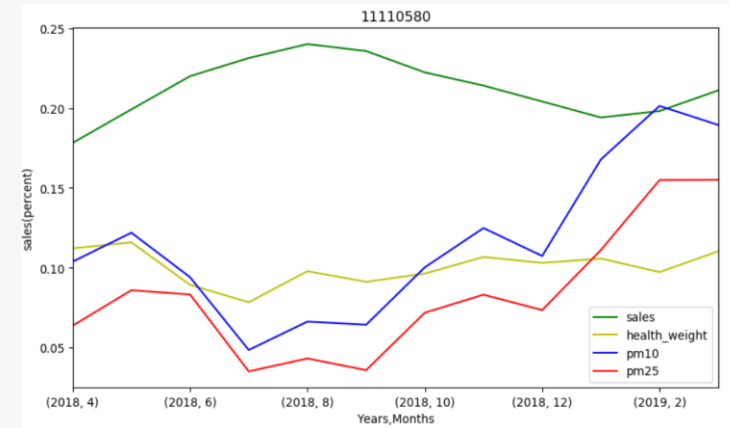
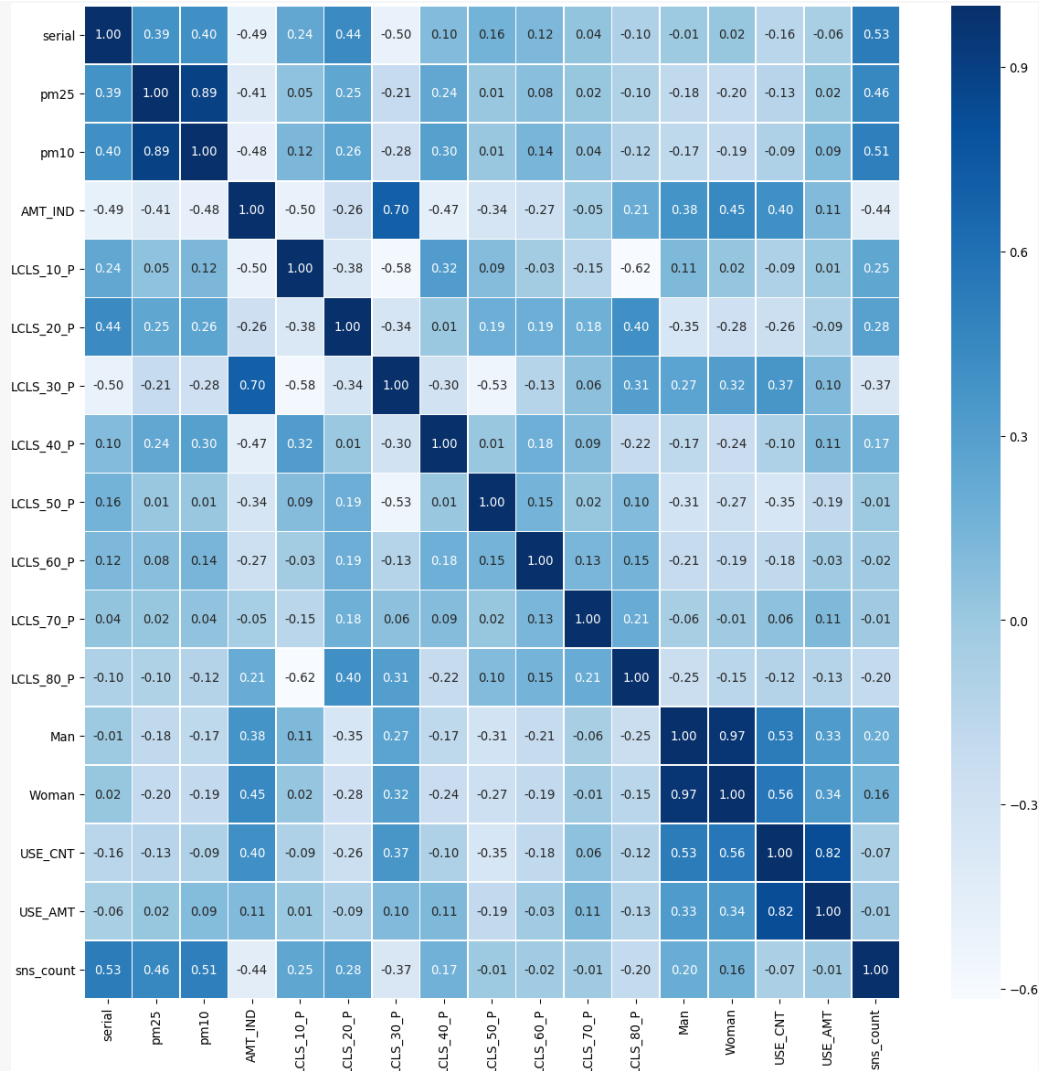
pre_all.csv

| | serial | Year | Month | Day | pm25 | pm10 | pm10_quil | pm25_quil | AMT_IND | LCLS_10_P | ... |
|---|----------|------|-------|-----|-----------|-----------|-----------|-----------|---------|-----------|-----|
| 0 | 11110515 | 2018 | 4 | 1 | 31.375000 | 66.041667 | 2.083333 | 2.333333 | 73.7 | 27.9 | ... |
| 1 | 11110515 | 2018 | 4 | 2 | 24.833333 | 64.500000 | 2.000000 | 2.000000 | 86.4 | 27.0 | ... |
| 2 | 11110515 | 2018 | 4 | 3 | 23.458333 | 70.083333 | 2.083333 | 2.125000 | 72.2 | 31.8 | ... |
| 3 | 11110515 | 2018 | 4 | 4 | 10.818182 | 18.545455 | 1.090909 | 1.272727 | 71.7 | 34.0 | ... |
| 4 | 11110515 | 2018 | 4 | 5 | 6.000000 | 9.588235 | 1.000000 | 1.000000 | 60.7 | 38.1 | ... |

5 rows × 21 columns

각 데이터당 하나의 csv파일로 저장한후 이를 지역,날짜 기준으로 전부 통합한 csv파일을 만들었습니다.

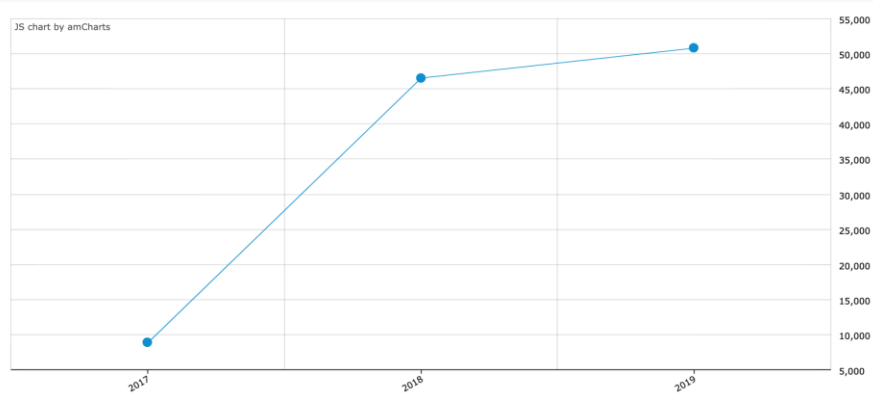
데이터 전처리



**#어떤 비즈니스 아이디어를
제시할 것인가?**


상황 분석

"미세먼지"에 대한 전체 언론사 뉴스 검색량
2017-08-20~2019-08-20 기준




빅카인즈, 2019.08.20

다양한 분야에 특화된 카드 상품들




[IBK기업] 이사배카드
H&B스토어, 화장품, 미용 할인으로 뷰티에 관심 있는 고객 대상 특화서비스 제공

비교함담기



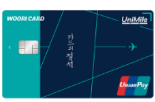
[하나카드] 카카오톡 하나카드
카카오톡 이용할 땐 카카오톡 하나카드

비교함담기



[우리카드] 카드의정석 명명당이
반려동물의 일상을 담다!

비교함담기



[우리카드] 카드의정석 UniMile
UniMile 하나로 6대 항공사 모두 OK

비교함담기

BC 카드

기존 카드상품에는 미세먼지에 특화된 상품이 없었습니다.

상황 분석

■ 썸타는 우리카드

온라인쇼핑 **할인**부터
해외이용 **수수료 면제**까지

- 국내 온라인쇼핑/외식/영화/ 등 10~20% 할인
- 해외이용 수수료 면제, 해외 가맹점
이용금액 1~2% 캐시백



○ (현황·문제점) 카드상품 출시 전 자체 수익성 분석 및 내부통제 기준 마련이 의무화*되어 있으나,

* 신용카드업자는 카드상품 설계·변경시 상품 설계기준을 포함한 상품의 수익성 분석을 실시하고, 이와 관련한 내부통제기준을 마련하여야 함(감독규정 §24의12)

- 예상수익 과대산정 및 예상손실 과소산정 등 엄밀하지 않은 수익성 분석과 미흡한 내부통제로 사후에 손실이 큰 카드상품이 지속 발생

금융감독원, 2019. 04. 09

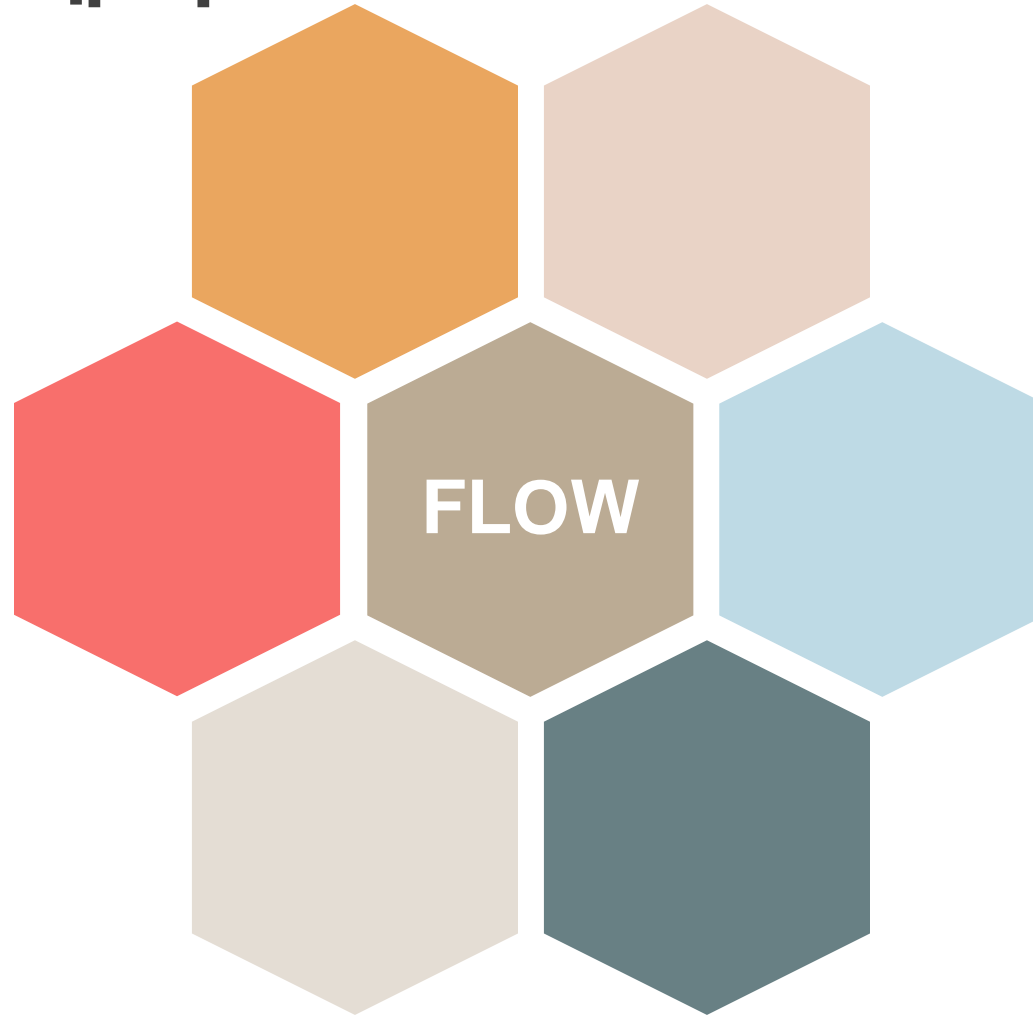
한국신용평가 KIS는 2019년 신용카드산업의 KIS Credit Outlook이 '부정적'이라는 평가를 내렸습니다.

한국신용평가 KIS, 2019. 01

#AI 기반 최적의 혜택을 주는 카드
FLOW

아이디어 제시

FLOW 는



주어진 데이터를 통해 미세먼지에 따른 소비자의 구매 패턴을 머신러닝 기반으로 예측하여
기업과 사용자가 최적의 이익을 얻을 수 있는 혜택을 가진 카드 입니다.

#서비스 아키텍처

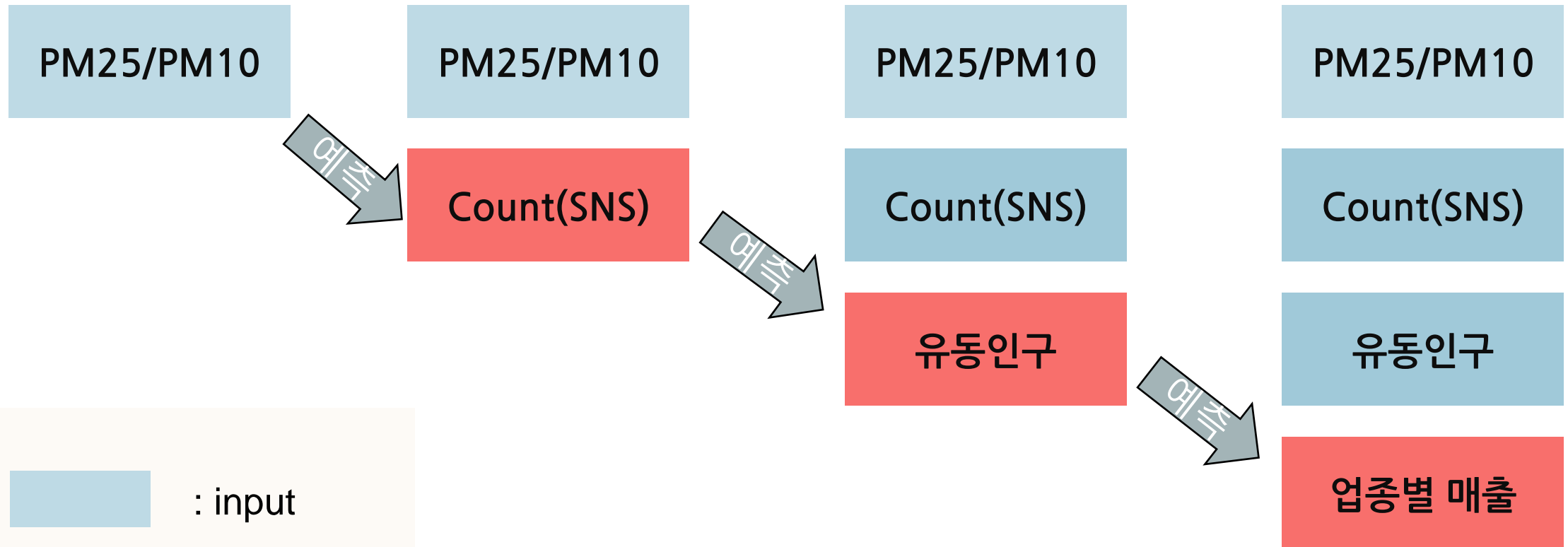
서비스 아키텍처 및 기술

Innovation 분야

| 유동인구데이터 | 카드매출데이터 | SNS데이터 | 환경기상데이터 | 유통데이터 |
|---------|---------|-----------|-------------|------------|
| 기준년도 | 기준일자 | 문서 KEY값 | 데이터측정 날짜 시간 | 영업일자 |
| 법정동_코드 | 구코드 | 문서 등록일 | 측정기 고유번호 | 구코드 |
| 법정동_명칭 | 법정동코드 | 블로그 카페 뉴스 | 실외 측정기 구분 | 법정동코드 |
| 시도_코드 | 업종코드 | | 미세먼지 PM-10 | 매출지수 |
| 시도_명칭 | 성별코드 | | 이산화탄소 농도 | 식사_비중 |
| 시군구_코드 | 나이코드 | 문서 제목 | 휘발성유기화합물 농도 | 간식_비중 |
| 시군구_명칭 | 이용건수 | 문서 본문 | 소음 데이터(db) | 마실거리_비중 |
| 길이 | 이용금액 | | 온도(°C) 데이터 | 홈&리빙_비중 |
| 면적 | | | 습도(%) 데이터 | 헬스&뷰티_비중 |
| X_좌표 | | | 미세먼지 PM-2.5 | 취미&여가활동_비중 |
| Y_좌표 | | | | 사회활동_비중 |
| | | | | 임신/육아_비중 |
| | | | | 기호품_비중 |

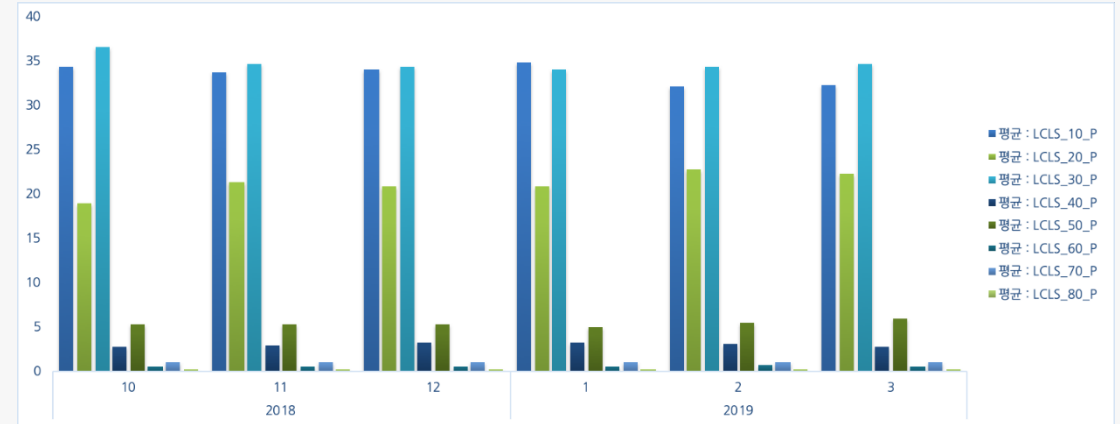
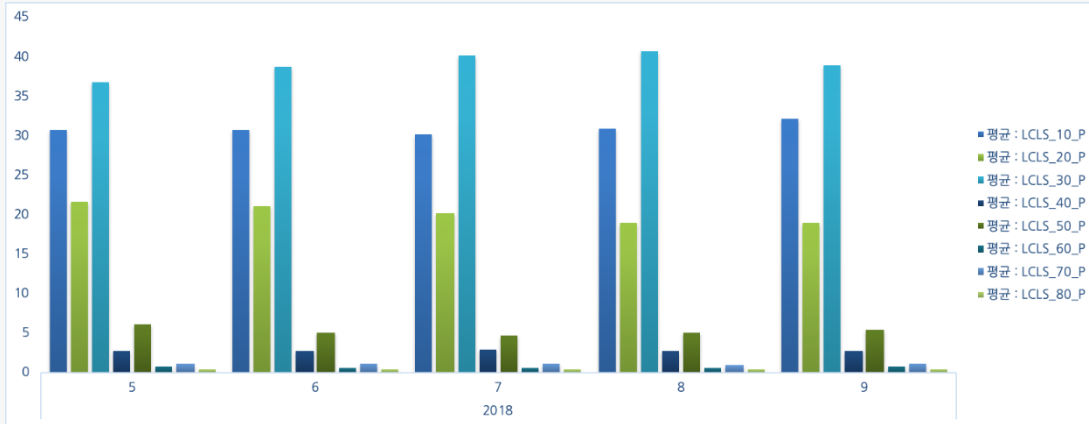
미세먼지로 인한 소비/경제/행동변화에 따른 사회적 영향 분석 및 예측 모델링

서비스 아키텍처 및 기술



변수 선택 과정

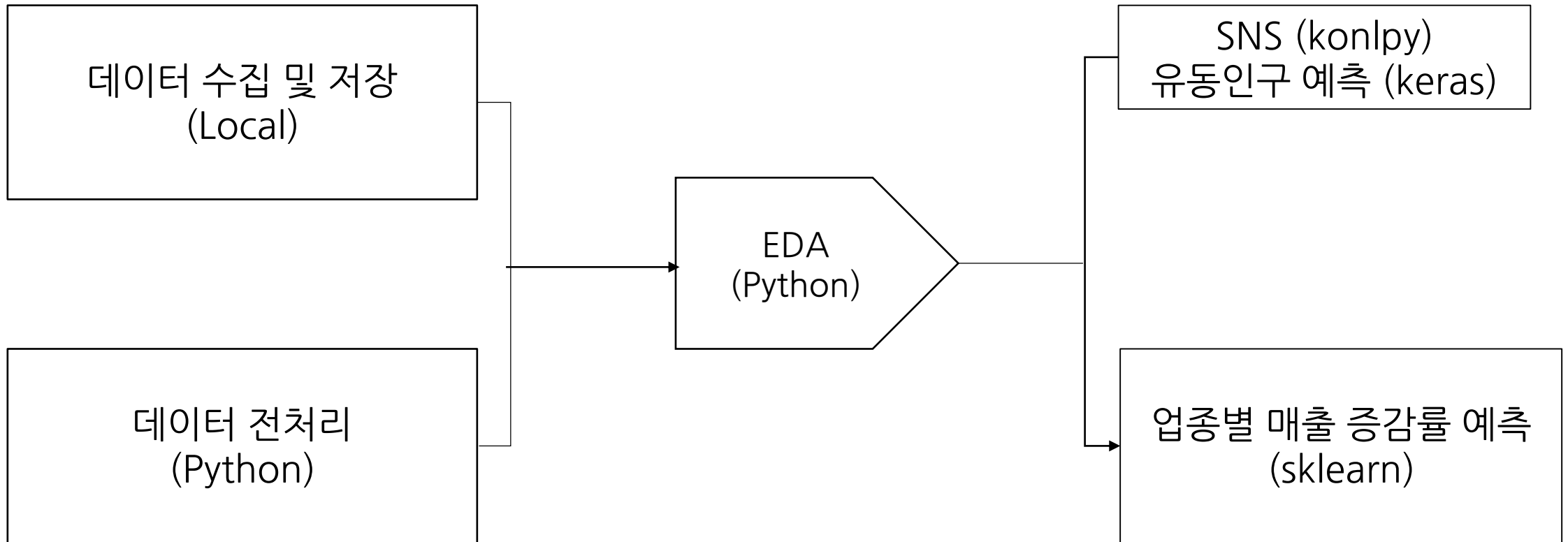
- EDA



EDA를 하면서 미세먼지가 비교적 덜한 2018년 5~9월의 경우와 2018.10~2019.3월의 경우를 나눠서 보았을 때, 품목별로 비교할 수 있었으나 이 과정에서 미치는 피쳐의 영향력에 대해서는 크게 알 수 없었음

이 때문에 다양한 Feature Engineering의 방법 중에서 처음부터 특정한 피쳐를 무시할 수 없으므로, 소거법인 **Backward Elimination**(Recursive Feature Elimination)을 택하여 변수를 선택하고자 함

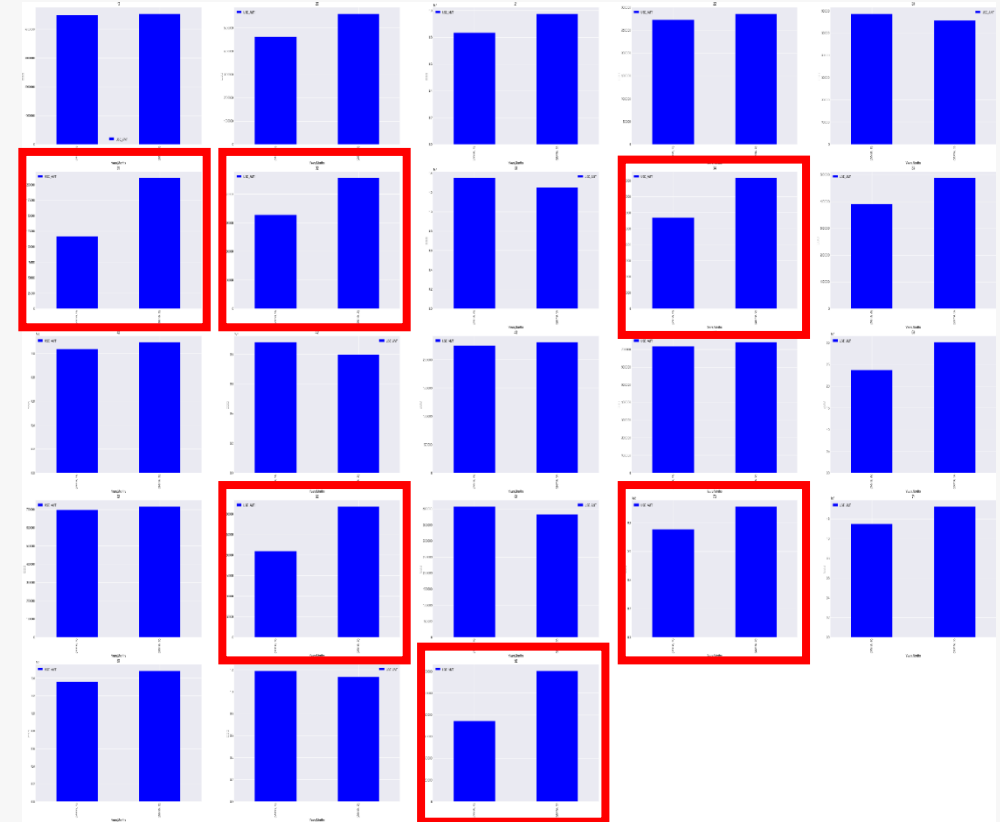
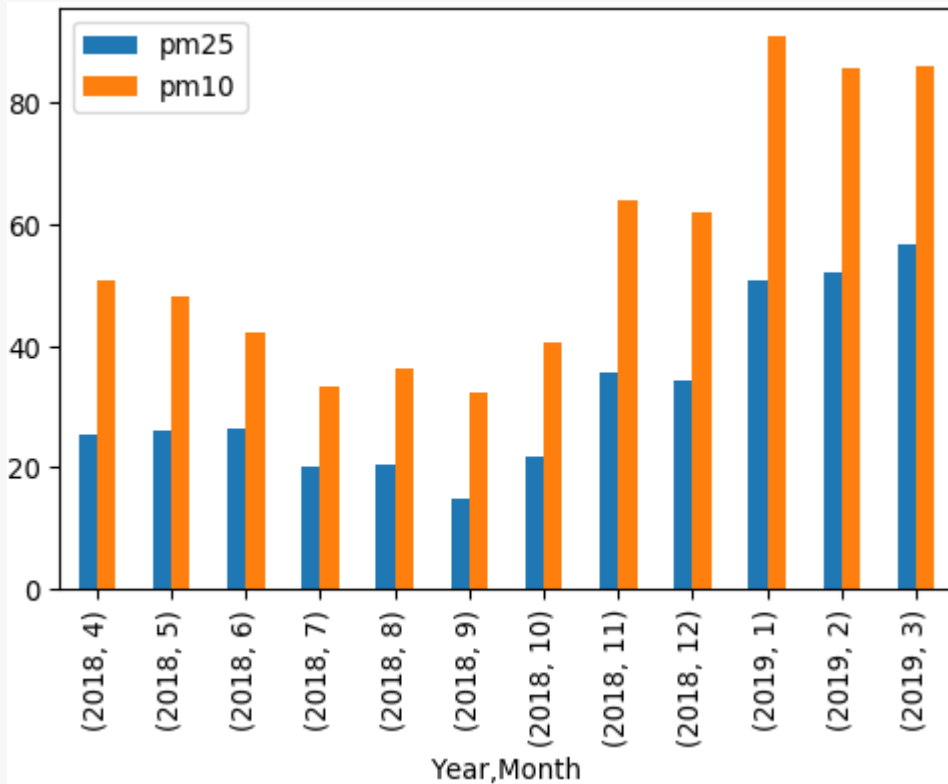
서비스 아키텍처 및 기술



#SNS데이터 분석

카드데이터 업종별 차이

- 미세먼지가 높은 달, 낮은 달 구분



수리서비스(세탁소), 요식업, 보건위생, 의료기관, 자동차 판매, 서적문구, 가전 등의 업종에서 영향을 받고있었습니다.

키워드 찾기

- word2vec를 통해 미세먼지 관련 키워드 분석

단어 간 유사도를 고려하기 위해서는 단어의 의미를 벡터화

이를 가능하게 하는 방법 중 하나가 워드투벡터(Word2Vec)

```
In [132]: from gensim.models import Word2Vec  
model = Word2Vec(result, size=30, window=5, min_count=5, workers=4, sg=0)
```

```
In [133]: f=model.wv.most_similar("미세먼지")  
print(f)
```

```
[('초미세먼지', 0.8594000339508067), ('극성', 0.8676562404632568), ('황사', 0.8152838945388794), ('농도', 0.8058701157569886), ('안전지대',  
0.7867635488610132), ('숨쉬기', 0.7827082872390747), ('최악', 0.7752515077590942), ('기승', 0.7591137290000916), ('꽃가루', 0.7589143514633  
179), ('갈수록', 0.7454203367233276)]
```

미세먼지라는 키워드가 나왔을 때 카드매출과 관련이 있는 특정 키워드를 찾을 수 없었습니다.

SNS데이터 단어 빈도수

- 미세먼지가 높은 달과 낮은 달에 나오는 단어 빈도수

Counter({'미세먼지': 11124, '수': 10566, '것': 10473, '보기': 6027, '피부': 5877, '때': 5809, '더': 5805, '곳': 5502, '사용': 5468, '기능': 5412, '미': 4811, '추가': 4692, '아름': 4398, '등': 4309, '날': 4213, '기타': 4205, '오늘': 4129, '번역': 4076, '본문': 4074, '복사': 4070, '저': 3991, '집': 3940, '생각': 3779, '때문': 3756, '아이': 3708, '제품': 3675, '내': 3654, '결말': 3643, '우리': 3606, '사진': 3570, '나': 3343, '시간': 3293, '마스크': 3286, '그': 3164, '제': 3045, '위': 2999, '후': 2944, '사람': 2943, '오즘': 2866, '청소': 2720, '하나': 2677, '거': 2669, '중': 2568, '전': 2553, '관리': 2526, '안': 2524, '도': 2512, '오': 2505, '바로': 2491, '문': 2414, '말': 2341, '물': 2313, '눈': 2295, '해': 2286, '카페': 2281, '차량': 2252, '정도': 2237, '추천': 2217, '진짜': 2198, '및': 2173, '좀': 2133, '맛': 2129, '차': 2120, '위해': 2062, '개': 2013, '비': 2007, '날씨': 1986, '느낌': 1947, '가격': 1866, '공기': 1854, '엄마': 1847, '제거': 1821, '점': 1819, '가지': 1773, '효과': 1756, '시작': 1753, '살': 1697, '일': 1692, '중고차': 1691, '여행': 1670, '이제': 1666, '먼지': 1663, '친구': 1642, '도': 1633, '공기청정기': 1618, '방법': 1617, '얼굴': 1603, '번': 1599, '한번': 1598, '마음': 1579, '구매': 1561, '보고': 1559, '케어': 1559, '경우': 1548, '다시': 1547, '이번': 1546, '꼭': 1541, '성분': 1540, '건강': 1529, '앞': 1518, '필터': 1514, '팩': 1509, '역': 1496, '출': 1493, '그날': 1489, '용': 1461, '코': 1433, '모두': 1431, '부분': 1430, '볼': 1424, '걱정': 1422, '지금': 1420, '확인': 1419, '여기': 1414, '다음': 1406, '조금': 1404, '방': 1396, '다른': 1383, '아주': 1382, '준비': 1381, '전체': 1374, '서울': 1368, '처음': 1353, '사실': 1350, '가장': 1333, '진행': 1331, '아침': 1329, '기분': 1320, '이용': 1317, '크림': 1294, '저희': 1284, '역시': 1273, '맛집': 1273, '알': 1268, '길': 1252, '손': 1248, '실내': 1244, '음식': 1242, '땀': 1235, '목': 1232, '환경': 1230, '만': 1220, '차단': 1219, '모습': 1206, '소개': 1204, '저장': 1202, '로': 1199, '몸': 1190, '이상': 1188, '위치': 1186, '시': 1179, '법': 1170, '약': 1162, '통해': 1155, '장소': 1151, '한국': 1142, '공간': 1136, '를': 1130, '커피': 1125, '얼': 1125, '저녁': 1123, '감': 1122, '작업': 1116, '게': 1114, '속': 1113, '지': 1099, '문제': 1096, '고민': 1094, '지도': 1091, '상담': 1085, '설치': 1082, '입': 1081, '방': 1072, '상태': 1065, '후기': 1052, '정보': 1047, '계속': 1037, '겨울': 1035, '대한': 1032, '건': 1029, '거리': 1024, '못': 1022, '달': 1019, '항': 1019, '의': 1018, '가족': 1018, '선물': 1016, '아파트': 1001, '개선': 998, '더욱': 996, '하늘': 992, '시공': 996, '듯': 994, '줄': 994, '수분': 992, '운동': 991, '주문': 990, '황사': 979, '직접': 976, '왜': 964, '전화': 957, '지역': 950, '주변': 943, '동만': 940, '생활': 939, '선택': 935, '체험': 931, '관심': 930, '보': 927, '꽃': 924, '아기': 919, '도움': 915, '판매': 909, '사업': 908, '일상': 902, '물렌징': 902, '오후': 897, '위': 896, '모든': 895, '자동차': 889, '이유': 883, '라인': 883, '폼': 881, '바람': 875, '대해': 872, '포스팅': 869, '배': 866, '자극': 864, '선진': 862, '발생': 859, '끝': 853, '화장품': 853, '머리': 850, '실': 848, '레이스': 841, '물질': 839, '만': 835, '주말': 831, '기술': 830, '은': 829, '명': 828, '부산': 821, '방문': 819, '이마기': 818, '고객': 815, '메뉴': 813, '사랑': 812, '마지막': 812, '기': 809, '주': 804, '옷': 803, '완전': 797, '공원': 793, '최고': 789, '도지': 787, '종류': 786, '만들기': 785, '외출': 784, '먼저': 783, '대': 781, '온': 781, '점문': 778, '디자인': 777, '벌': 776, '도착': 776, '말': 775, '자연': 774, '자리': 773, '가기': 772, '선': 771, '미세': 771, '비용': 771, '자주': 766, '술': 766, '두피': 765, '국내': 759, '열': 755, '팝업': 755, '여러분': 754, '고기': 754, '브랜드': 753, '삼푸': 752, '구입': 751, '매일': 750, '추출': 749, '병원': 747, '방': 747, '아래': 743, '분위기': 740, '문의': 740, '개인': 737, '중국': 735, '이벤트': 732, 'نامه': 728, '바디': 727, '인천': 724, '스킨': 720, '서비스': 718, '인테리어': 716, '구경': 715, '혼자': 714, '코스': 713, '진정': 712, '참': 707, '식물': 703, '원인': 702, '난': 700, '연체': 699, '책': 693, '업체': 692, '예약': 684, '간': 683, '죽': 682, '할인': 679, '티': 678, '두': 674, '기도': 668, '무료': 666, '막': 665, '친환경': 664, '타입': 661, '관련': 660, '무': 659, '상품': 658, '여성': 657, '지원': 653, '아빠': 651, '항상': 651, '마사지': 648, '유지': 647, '각질': 644, '트러블': 641, '회사': 639, '산업': 639, '배출': 637, '거의': 637, '대구': 637, '글': 636, '자': 636, '해결': 634, '블로그': 633, '산': 629, '증상': 625, '교회': 625, '닉': 625, '마무리': 623, '착용': 621, '제일': 614, '매우': 613, '타고': 608, '나무': 608, '계획': 606, '피지': 600, '애': 600, '기본': 600, '세트': 596, '애어': 596, '화이트': 595, '이름': 595, '실명': 594, '제대로': 592, '돈': 592, '제품': 591, '세안': 590, '내부': 590, '무엇': 589, '블랙': 587, '미리': 587, '뿐': 586, '점심': 586, '여름': 586, '어디': 585, '피': 585, '오후': 584, '바': 584, '첫': 583, '인': 582, '내일': 581, '통': 580, '매장': 579, '보호': 577, '남자': 577, '일반': 575, '쇼핑': 575, '뷰티': 575, '참고': 572, '리': 569, '시민': 569, '신경': 567, '주의': 567, '화장': 565, '보통': 565, '밤': 565, '비교': 565, '이': 563, '활동': 563, '상세': 562, '단지': 561, '청소기': 561, '호텔': 560, '덕': 560, '얼마나': 559, '시설': 558, '해도': 558, '시술': 558, '개울': 558, '세상': 556, '정리': 555, '일단': 553, '세척': 552, '현재': 552, '비타민': 552, '기간': 552, '필수': 551, '하니': 549, '별로': 548, '모공': 546, '방지': 544, '천연': 543, '다이어트': 543, '결기도': 542, '과': 540, '감기': 539, '창문': 538, '행사': 538, '보습': 537, '오빠': 537, '초미세먼지': 536, '빌라': 536, '방충': 532, '소리': 531, '빨래': 529, '살짝': 529, '질문': 525, '다리': 525, '토너': 525, '거품': 524, '에너지': 524, '새해': 524, '입구': 523, '여자': 523, '용량': 522, '촬영': 522, '도시': 521, '사이': 521, '늘': 520, '구성': 519, '제주도': 517, '구역': 517, '기존': 516, '고': 516, '에어컨': 516, '스': 515, '정화': 514, '적': 513, '원가': 513, '인기': 513, '호': 513, '부담': 513, '개발': 512, '면': 511, '강남': 511, '학생': 511, '점': 510, '지역': 510, '예발': 506, '먹기': 505, '농도': 505, '장': 505, '시스템': 505, '거실': 504, '오염': 503, '원래': 502, '예정': 501, '미국': 500, '여름': 498, '바닥': 498, '결과': 497, '버스': 496, '라면': 495, '세': 495, '산책': 495, '학교': 495, '시장': 494, '키즈': 494, '날땀': 493, '스트레스': 493, '사항': 491, '구': 490, '유': 490, '빵': 485, '일본': 485, '염증': 485, '폼': 483, '운

미세먼지가 높은 달에 유의미하다고
생각되는 빈도수:

피부(5877), 청소(2720), 차량(2252),
중고차(1691), 공기청정기(1618),
건강(1529), 약(1116), 치료(1099),
아기(919), 자동차(889), 키즈(494),
스트레스(493)

차량(60) , 병원(70) , 가전(35) ,
보건위생업종(71)

예측 모델링

1. 가진 데이터로 최종적으로 하고자하는 업종별 매출량 예측이 되는지 여부를 알기 위하여, 머신러닝 진행 (전제 확인)

```
In [63]: #prediction with linear regression model
lin_reg_predictions = lin_reg.predict(data_test_prepared)
mse, r2, rmse = evaluation(data_test_labels, lin_reg_predictions)

print('Test Set Check', 'MSE: ', mse, '\nR2 score: ', r2, '\nRMSE: ', rmse)

Test Set Check MSE: 0.05474590927577427
R2 score: 0.9977948285652508
RMSE: 0.23397843762999673
```

Linear regression

```
In [64]: #prediction with Decision Tree model
dt_predictions = tree_reg.predict(data_test_prepared)
mse, r2, rmse = evaluation(data_test_labels, dt_predictions)

print('Test Set Check', 'MSE: ', mse, '\nR2 score: ', r2, '\nRMSE: ', rmse)

Test Set Check MSE: 1.5209163111388293
R2 score: 0.9387373184894464
RMSE: 1.233254357843032
```

Decision Tree

```
In [65]: #prediction with Random Forest regressor model
rf_predictions = random_forest_reg.predict(data_test_prepared)
mse, r2, rmse = evaluation(data_test_labels, rf_predictions)

print('Test Set Check', 'MSE: ', mse, '\nR2 score: ', r2, '\nRMSE: ', rmse)

#So when I used mean USE_AMT's best result was with random forest regressor (75% R2 score and 82 MSE)

Test Set Check MSE: 1.1580126045318537
R2 score: 0.9533551209510529
RMSE: 1.0761099407271795
```

Random Forest

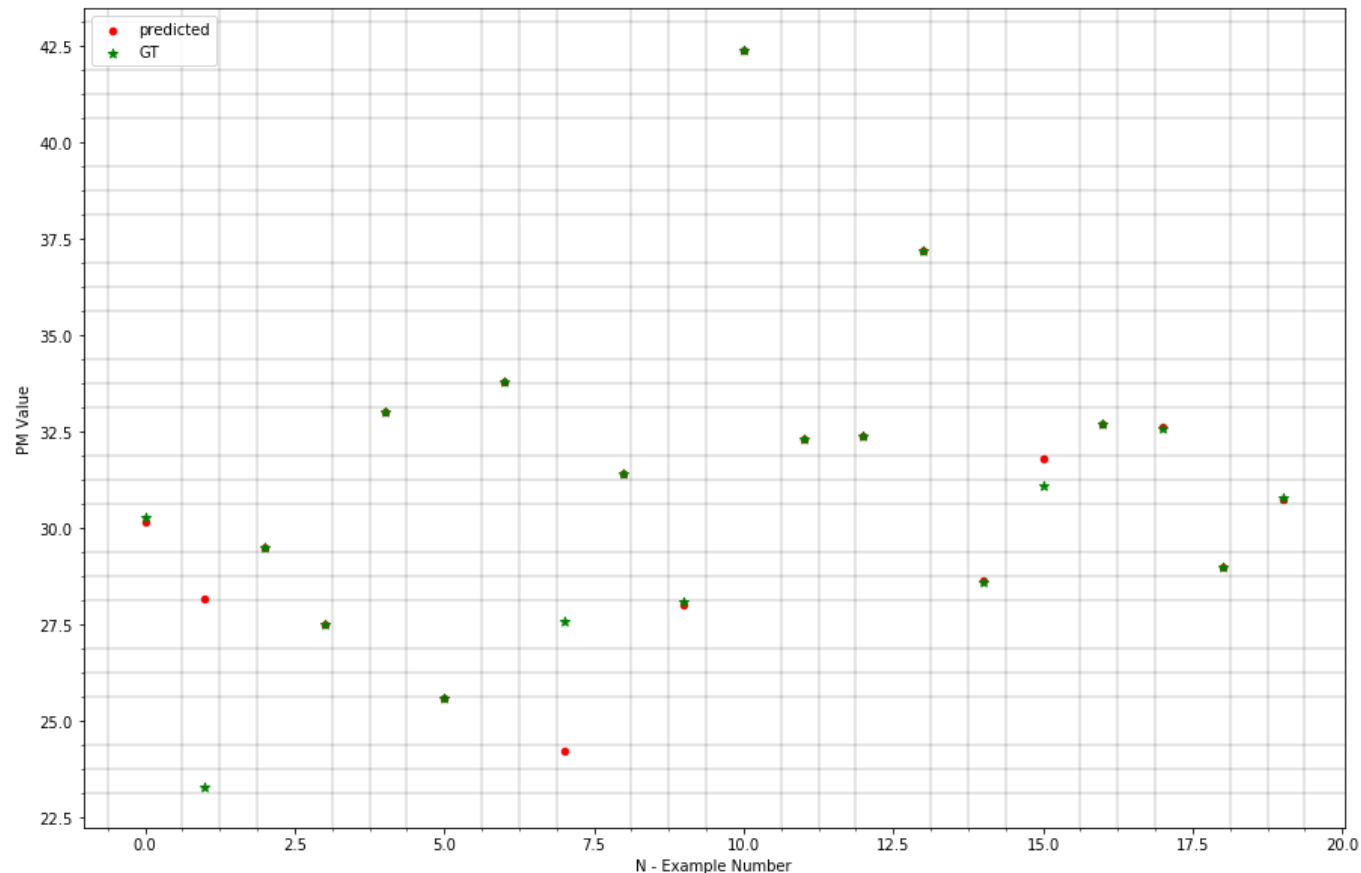
예측 모델링

1. GS category 10 (rf를 사용한 test 결과 중 random하게 10개를 가져옴)

```
In [71]: print('Predictions: ', '\tGround Truth Labels')
for i in range(len(rf_predictions[:10])):
    print(int(rf_predictions[i]), '\t\t', int(data_test_labels.iloc[i]), '\n')
```

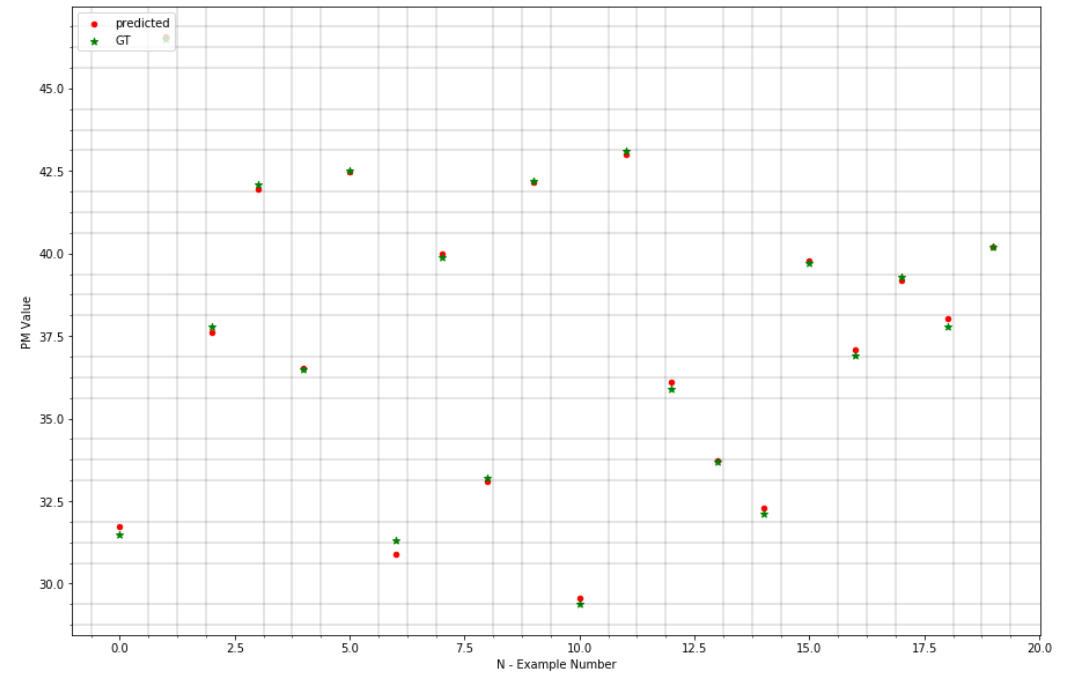
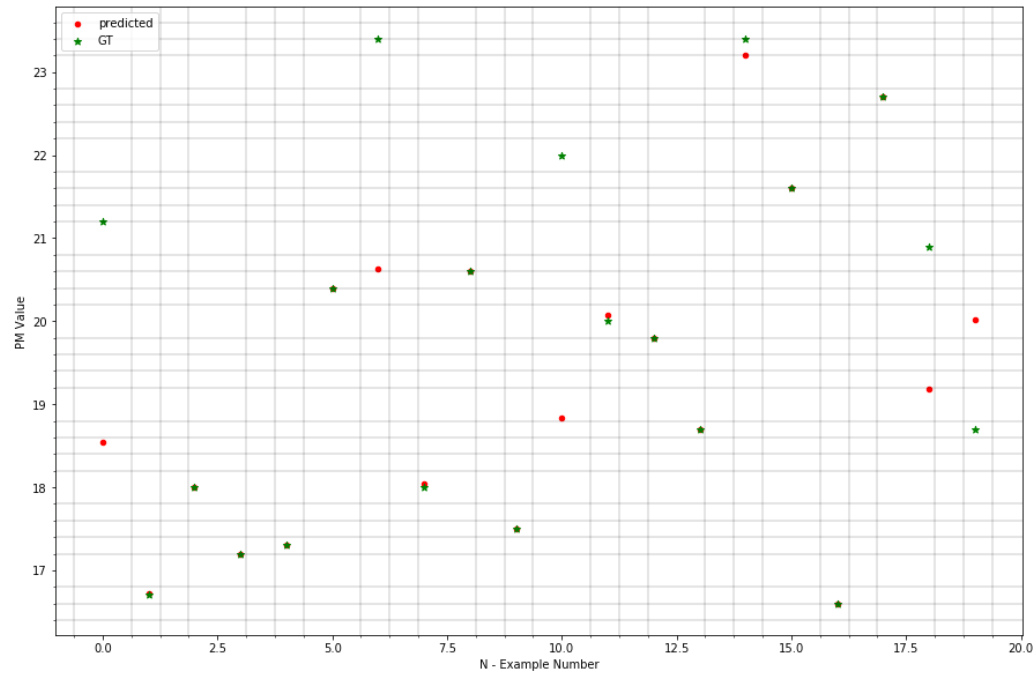
| Predictions: | Ground Truth Labels |
|--------------|---------------------|
| 30 | 30 |
| 28 | 23 |
| 29 | 29 |
| 27 | 27 |
| 33 | 33 |
| 25 | 25 |
| 33 | 33 |
| 24 | 27 |
| 31 | 31 |
| 28 | 28 |

10개 중 8개로 overlap된 것이 보임



예측 모델링

1. GS category 20 & 30



**가진 데이터로 최종적으로 하고자 하는 업종별 매출액 예측이 된다는 전제 확인

예측 모델링

2. 모델 결정

- 선형회귀를 이용해서 데이터를 처리

선형회귀를 사용하는 이유는 시계열 데이터를 통해 업종별 패턴을 파악하기 위함

- 선형회귀를 사용하기 위해서는 아래 가정이 충족해주는 것이 좋다.

- 오차항은 평균이 0이고 분산이 일정한 정규 분포를 갖는다.
- 독립변수와 종속변수는 선형 관계이다.
- 오차항은 자기 상관성이 없다.
- 데이터에 아웃라이어가 없다.
- 독립변수와 오차항은 서로 독립이다.
- 독립변수 간에서는 서로 선형적으로 독립이다.

예측 모델링

- 가정을 만족하기 위한 변수의 관계를 살펴 보기 전에 예측할 변수를 가정
- 사용하면 될 Input_data의 독립변수
 1. SNS 미세먼지 키워드 비중
 2. 미세먼지 PM25 수치, PM10 수치
 3. 유동인구 남자, 여자수
- 사용하며 될 target_data의 종속변수
 1. GS의 업종별 매출 비중
 2. 카드 매출의 업종별 매출 건수
 3. 카드 매출의 업종별 매출액

예측 모델링

- 선형회귀 가정을 만족하는지 변수 간의 검증

1. 독립변수와 종속변수는 선형적이어야 한다.

- 1) SNS개수와 업종별 카드, gs의 1년치 월별 데이터
- 2) 유동인구와 업종별 카드, gs의 1년치 월별 데이터
- 3) 미세먼지와 업종별 카드, gs의 1년치 월별 데이터

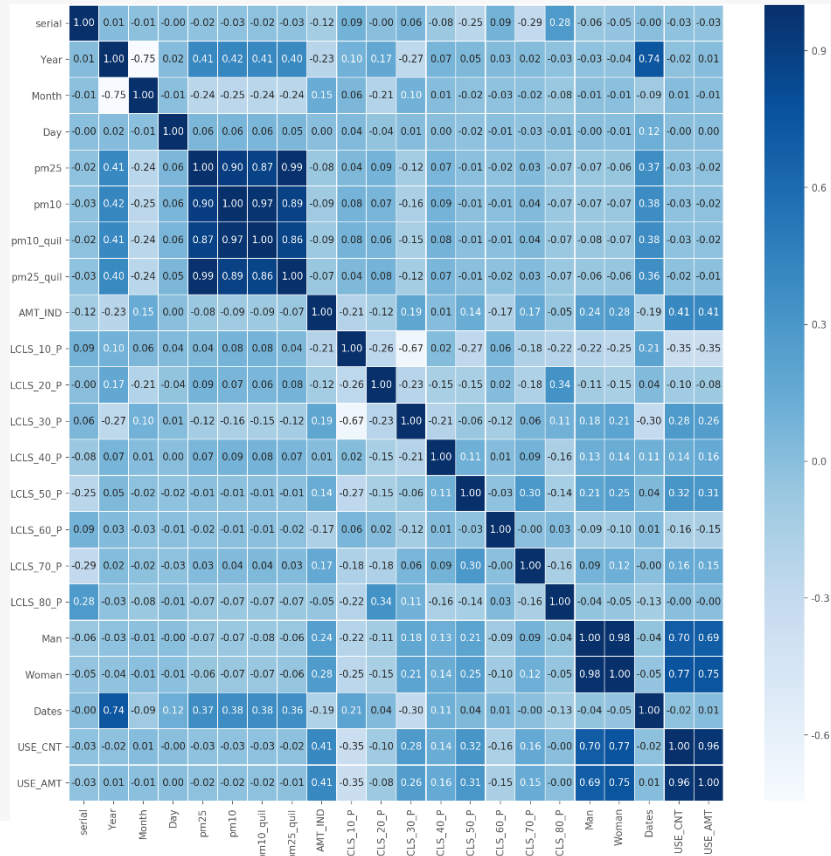
2. 독립변수와 종속변수 데이터의 아웃라이어가 없다.

- 1) 각각 데이터의 아웃라이어 제거한 1년치 월별 데이터

예측 모델링

- 선형회귀 가정을 만족하는지 변수 간의 검증

3. 독립변수간의 다중공선성이 존재하지 않아야 한다.



1. 유동인구의 남자, 여자수는 상관관계가 있으므로 둘을 합쳐서 유동인구 데이터 설정

2. pm25와 pm10의 상관관계 역시 존재하므로 pm25 와 pm10을 따로 분리하여 모델을 사용

예측 모델링

- 선형회귀 가정을 충족시킨 후의 최종 변수 설정

- 최종 Input_data의 독립변수
 - SNS 미세먼지 키워드 비중
 - 미세먼지 PM25 수치 or PM10 수치
 - 유동인구수

- 최종 target_data의 종속변수
 - GS의 업종별 매출 비중
 - 카드 매출의 업종별 매출 건수
 - 카드 매출의 업종별 매출액

| | | | | Count | pm25 | SEX_CD | USE_CNT | USE_AMT | People |
|------|-------|-----|----------|----------|-----------|--------|----------|----------|--------|
| YEAR | MONTH | DAY | serial | | | | | | |
| 2019 | 2 | 27 | 11110615 | 7.450419 | 44.250000 | 1 | 2.639057 | 6.285998 | 669530 |
| 2018 | 6 | 12 | 11110615 | 4.853668 | 10.708333 | 1 | 3.555348 | 6.785588 | 713223 |
| | | 26 | 11110650 | 5.424953 | 41.416667 | 0 | 2.197225 | 5.783825 | 103852 |
| | 10 | 6 | 11110615 | 5.016834 | 6.777778 | 0 | 2.890372 | 6.926577 | 468581 |
| 2019 | 2 | 9 | 11110615 | 6.068109 | 32.333333 | 1 | 3.295837 | 7.574558 | 484501 |
| 2018 | 12 | 16 | 11110530 | 5.721680 | 53.208333 | 1 | 3.988984 | 9.932367 | 112410 |
| | | | 11350695 | 5.721680 | 36.000000 | 1 | 3.295837 | 7.064759 | 159834 |
| 2019 | 1 | 25 | 11110630 | 7.092227 | 40.608696 | 0 | 1.386294 | 5.192957 | 191311 |
| | | 3 | 11110530 | 8.012710 | 39.608696 | 0 | 2.197225 | 7.935587 | 216632 |
| 2018 | 7 | 31 | 11110615 | 5.536823 | 15.041667 | 0 | 3.433987 | 8.472823 | 654997 |
| | 9 | 9 | 11110615 | 5.648986 | 13.875000 | 1 | 4.110874 | 8.034631 | 386476 |
| | | 10 | 11110650 | 5.988097 | 11.541667 | 0 | 1.386294 | 4.477337 | 104167 |
| | 5 | 27 | 11350695 | 5.330079 | 33.000000 | 1 | 1.386294 | 4.465908 | 174740 |
| | 9 | 18 | 11110530 | 5.965891 | 27.666667 | 0 | 2.890372 | 8.452121 | 251199 |

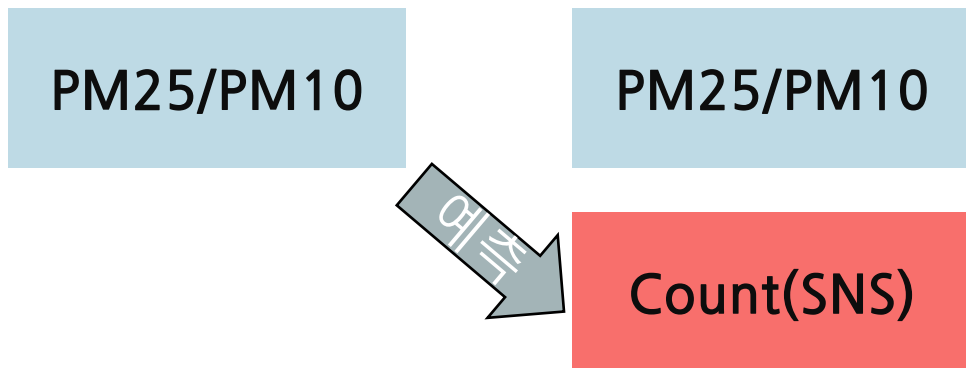
- 종속변수를 한번에 다변수선형회귀를 이용하여 구할 수도 있으나 업무 분담의 편리성을 위해서 단변수선형회귀를 이용하여 각 데이터를 하나씩 사용함

예측 모델링

- 카드매출 업종별 매출 패턴 분석
- 선형회귀 모델 사용
 - 1) Linear Regression
 - 2) Keras Regressor(Neural Network)
 - 3) Random Forest regressor
 - 이 중 RandomForest가 가장 mse와 R2-score가 양호하게 나옴

예측 모델링

3. PM10/ PM25를 Regressor를 통해 Count(SNS) 예측값 도출



```
#prediction with linear regression model
lin_reg_predictions = lin_reg.predict(data_test_prepared)
mse, r2, rmse = evaluation(data_test_labels, lin_reg_predictions)

print('Test Set Check', 'MSE: ', mse, '\nR2 score: ', r2, '\nRMSE: ', rmse)
```

```
Test Set Check MSE:  0.8664046928973682
R2 score:  0.3187024979903088
RMSE:  0.9308086231322571
```

```
#prediction with Decision Tree model
dt_predictions = tree_reg.predict(data_test_prepared)
mse, r2, rmse = evaluation(data_test_labels, dt_predictions)

print('Test Set Check', 'MSE: ', mse, '\nR2 score: ', r2, '\nRMSE: ', rmse)
```

```
Test Set Check MSE:  0.0005173655064430409
R2 score:  0.9995931695314497
RMSE:  0.02274567005922316
```

```
#prediction with Random Forest regressor model
rf_predictions = random_forest_reg.predict(data_test_prepared)
mse, r2, rmse = evaluation(data_test_labels, rf_predictions)

print('Test Set Check', 'MSE: ', mse, '\nR2 score: ', r2, '\nRMSE: ', rmse)
```

```
Test Set Check MSE:  0.00029698371749782945
R2 score:  0.9997664667948737
RMSE:  0.017233215529837413
```

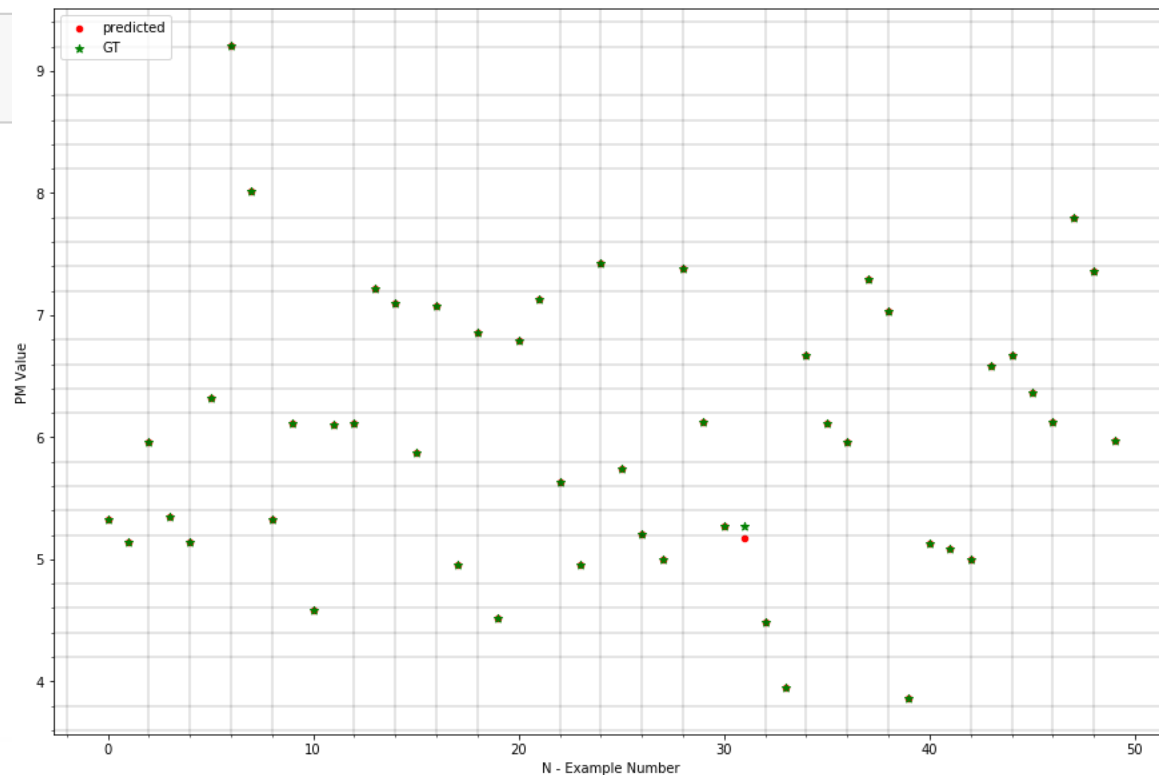
예측 모델링

3. PM10/ PM25를 Regressor를 통해 Count(SNS) 예측값 도출

예측 값중 10개를 가져와 라벨 값과 비교, 시각화

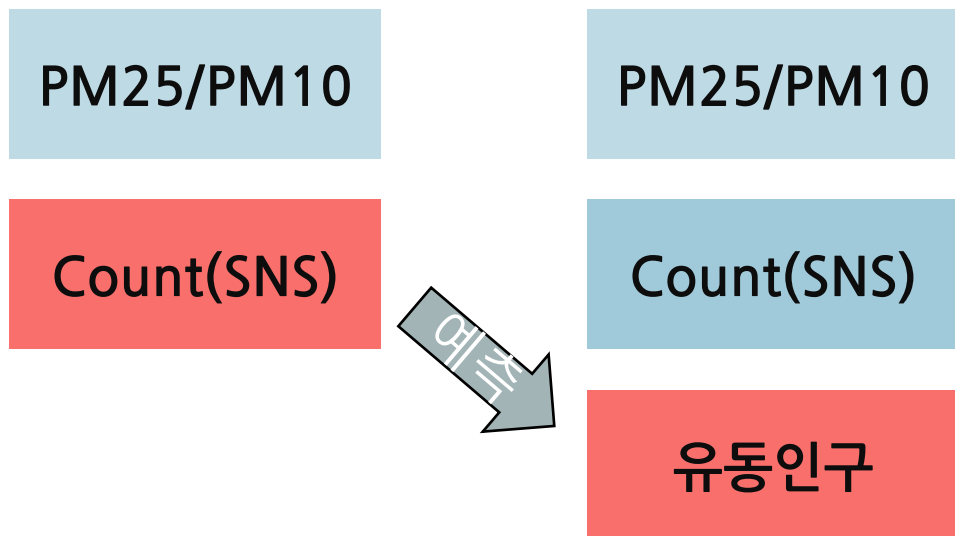
```
print('Predictions: ', '\tGround Truth Labels')
for i in range(len(rf_predictions[:10])):
    print(int(rf_predictions[i]), '\t\t', int(data_test_labels.iloc[i]), '\n')
```

| Predictions: | Ground Truth Labels |
|--------------|---------------------|
| 5 | 5 |
| 5 | 5 |
| 5 | 5 |
| 5 | 5 |
| 5 | 5 |
| 6 | 6 |
| 9 | 9 |
| 8 | 8 |
| 5 | 5 |
| 6 | 6 |



예측 모델링

4. PM10/ PM25, Count(SNS)를 Neural Net(for Regression)을 통해 유동인구 예측값 도출



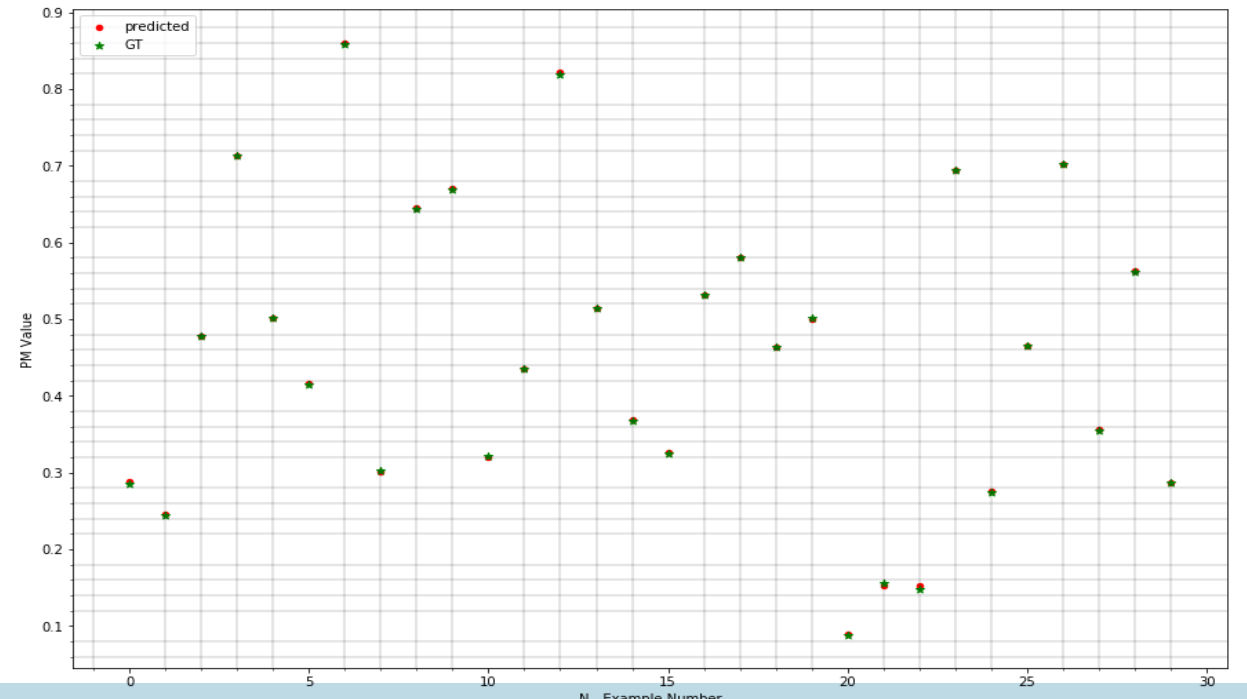
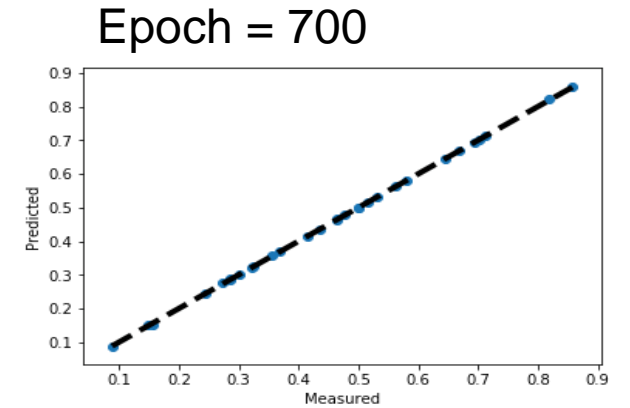
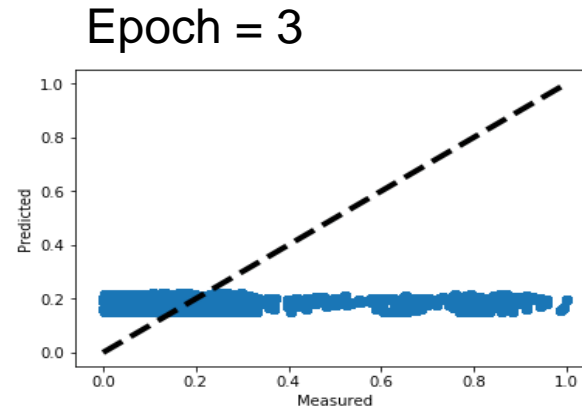
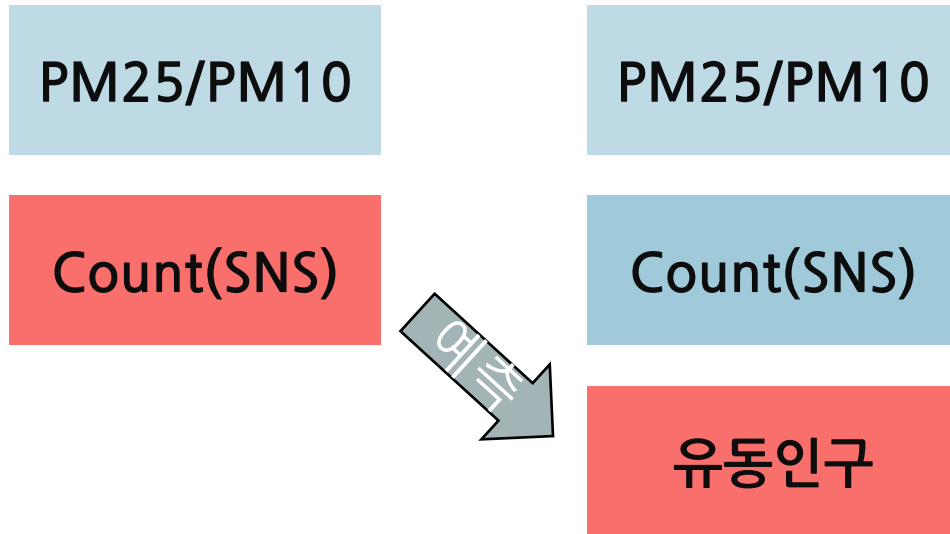
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.datasets import make_regression
from sklearn.preprocessing import MinMaxScaler
from numpy import array
# generate regression dataset
scalarX, scalarY= MinMaxScaler(), MinMaxScaler()
scalarX.fit(X)
scalarY.fit(y)
X = scalarX.transform(X)
y = scalarY.transform(y)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)

# define and fit the final model
model = Sequential()
model.add(Dense(32, #input_dim=2,
                  activation='relu'))
model.add(Dense(4, activation='relu'))
model.add(Dense(1, activation='linear'))
model.compile(optimizer='adam', loss='mean_squared_error', metrics=['accuracy'])
model.fit(X_train, y_train, epochs=100, verbose=0)
model.summary()

y_pred = model.predict(X_test)
```

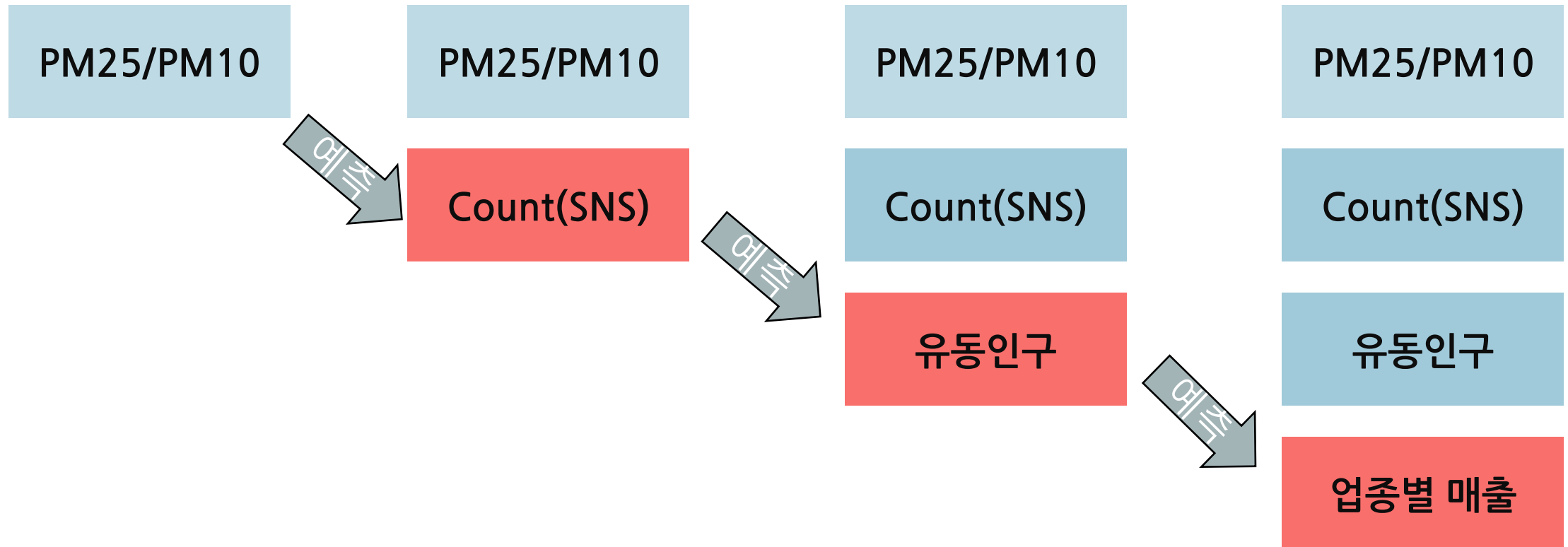
예측 모델링

4. PM10/ PM25, Count(SNS)를 Neural Net(for Regression)을 통해 유동인구 예측값 도출



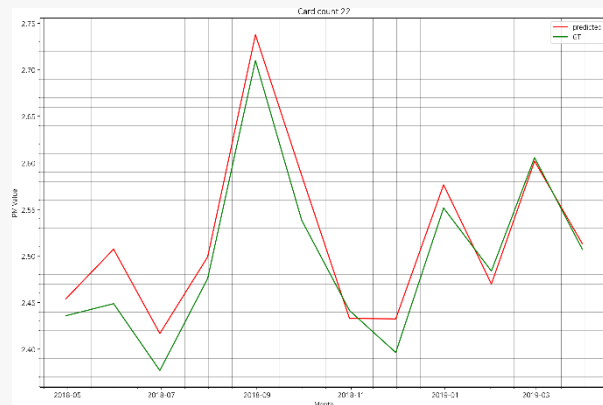
예측 모델링

5. 예측한 유동인구 데이터와 기존 피쳐들로 업종별 매출액 예측

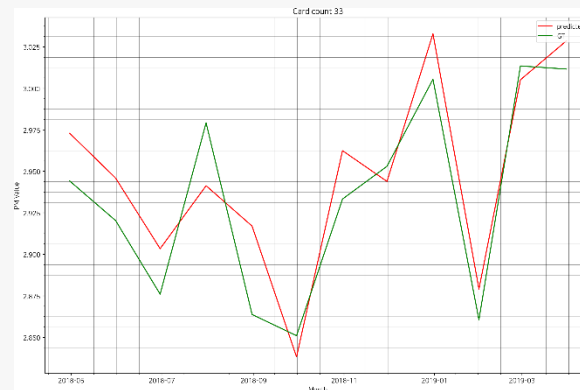


예측 모델링

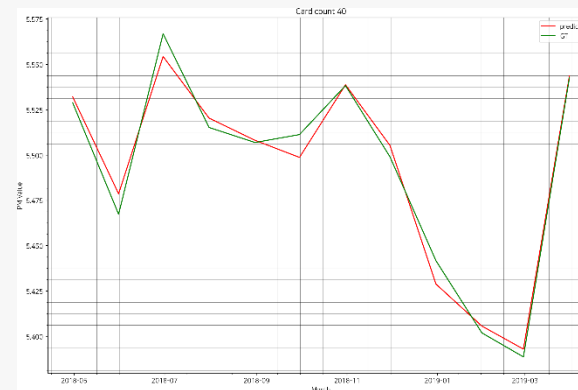
카드매출 업종별 매출건수 패턴 분석



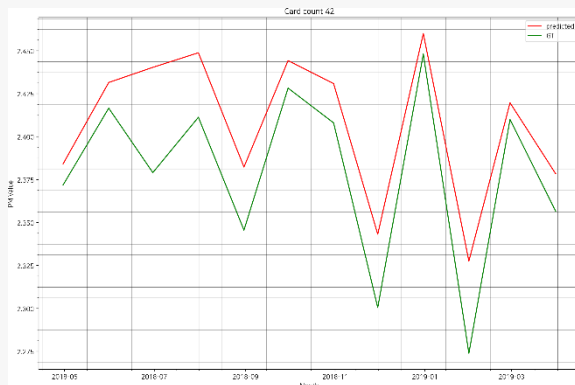
문화, 취미 매출 건수



연료, 판매 매출 건수



유통업 매출 건수



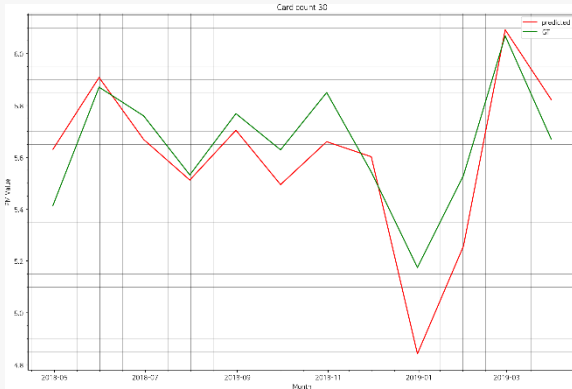
의복 매출 건수



의료기관 매출 건수

예측 모델링

카드매출 업종별 매출액 패턴 분석



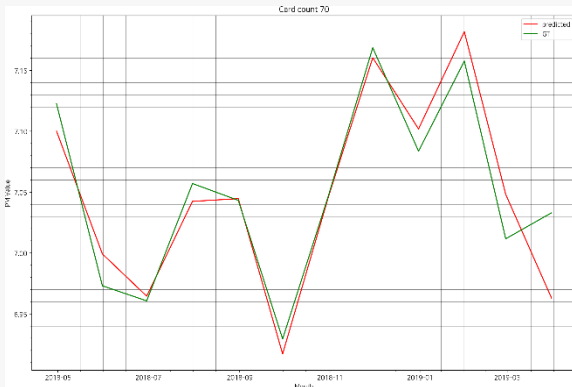
가구 매출액



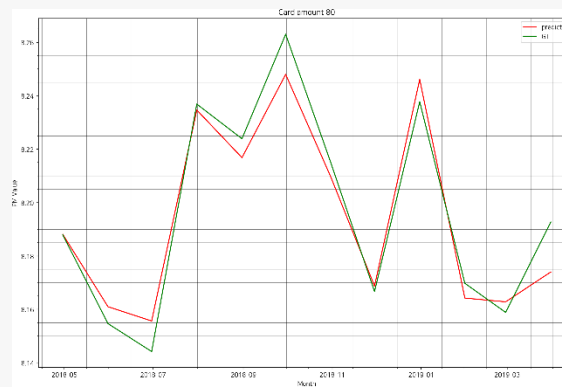
유통업 매출액



서적,문구 매출액



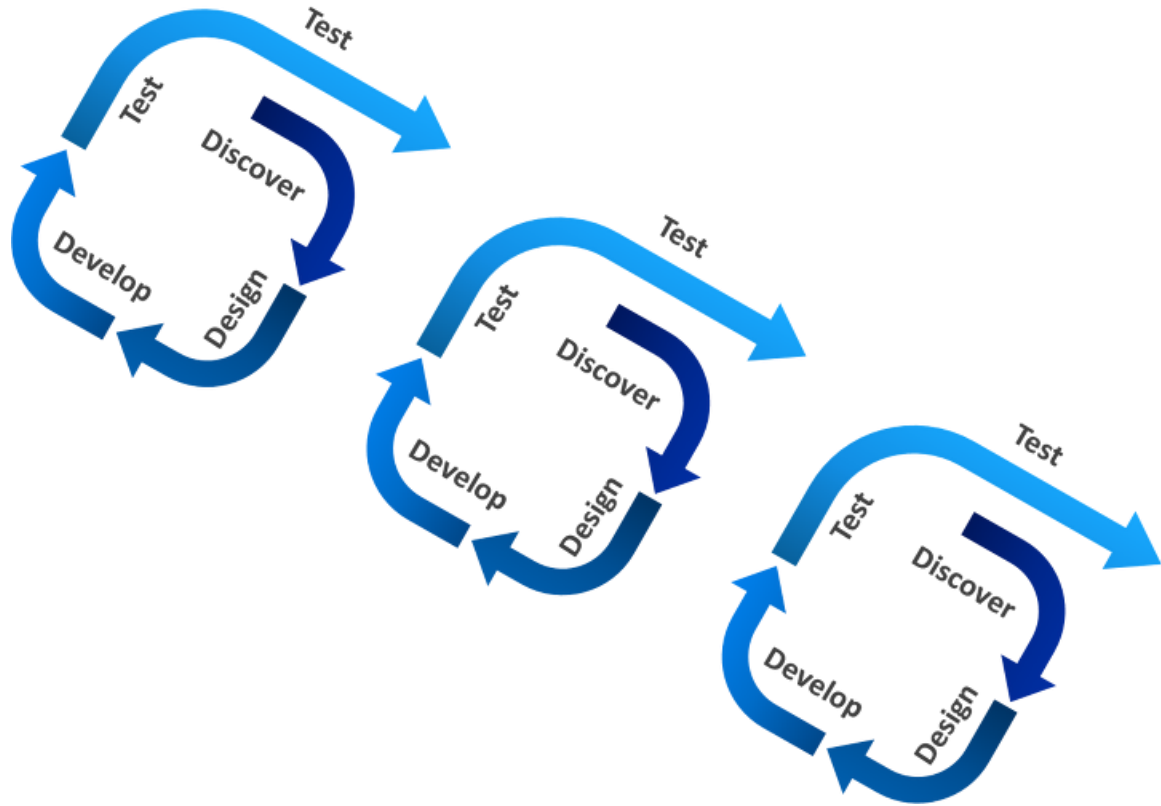
의료기관 매출액



요식업소 매출액

- 이것을 통해 예측모델이 어느정도 신뢰할 수 있음을 알 수 있다.

프로젝트 수행 방식



[DevOps]

1. 개발 환경

Local

PyCharm, Jupyter, Anaconda

2. 커뮤니케이션

Kakao

구글드라이브 모든 데이터셋, 코드,
그래프 통합 관리

실습 이후 오프라인 회의의 반복
(회의 시작할 때 기존 결과 피드백, 회의 끝난
후 구글드라이브에 회의록 작성)

3. 방법론

Agile

향후 과제

1. Neural Net(for Regression)의 최적의 파라미터 값 조정

2. 구한 패턴으로 미세먼지 맞춤형 카드 상품 제안

3. 정확도 낮은 업종별 예상 매출량의 정확도 높이기

(카드 매출 데이터는 업종 별 데이터 크기의 편차가 커, 서울특별시 빅데이터 캠퍼스에서 신한 카드 카드 매출 데이터를 추가로 다운받아볼 예정)

참고 문헌

- 김인중, 나기현, 양소희, 장재민, 김윤종, 신원영, & 김덕중. (2017). 딥러닝과 통계 모델을 이용한 T-커머스 매출 예측. 정보과학회논문지, 44(8), 803-812.

- 하나금융경영연구소. (2019). 미세먼지가 바꾼 소비 행태 변화. n.p.: KEB 하나은행.

- 안길승, 서민지, 허선, 박유진. (n.d.). 신경망을 이용한 SNS상에서의 정보확산 예측모형: Digg 사례를 중심으로. 정보화연구, 13권(4), pp. 609-616.

- [Subinium Tutorial] House Prices (Advanced) . (n.d.). <https://www.kaggle.com/subinium/subinium-tutorial-house-prices-advanced>.

데이터 제공

- 이동인구데이터(SK텔레콤)
- 카드매출데이터(신한카드)
- SNS데이터(와이즈넷)
- 환경기상데이터(케이웨더)
- 유통데이터(GS리테일)

#감사합니다